aws

Manual do usuário

EC2 Image Builder

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

EC2 Image Builder: Manual do usuário

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens de marcas da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestigie a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

O que é o Image Builder? 1
Recursos do Image Builder 2
Sistemas operacionais compatíveis 4
Formatos de imagem compatíveis 4
Cotas padrão 4
AWS Regiões e endpoints 4
Conceitos 5
Preços 8
Relacionado Serviços da AWS 8
Como o Image Builder funciona 9
Elementos da AMI 10
Gerenciamento de componentes 11
Recursos criados 11
Distribuição 12
Compartilhar recursos 13
Compliance 13
Versionamento semântico 13
Usando referências de versão 14
Padrões de referência de versão disponíveis 15
Recursos que retornam referências de versão 15
Configurar 17
Perfil vinculado ao serviço do Image Builder 17
Requisitos de configuração 18
Repositório de contêiner para pipelines de imagens de contêiner 18
Host dedicado para imagens do macOS 19
Pré-requisitos do IAM 19
Pré-requisitos do Systems Manager Agent 20
Tutoriais do Image Builder 22
Compilar sua primeira imagem 22
Criar um componente personalizado com parâmetros de entrada 22
Use os parâmetros do Systems Manager com o Image Builder 23
Assistente de pipeline: criar AMI 23
Etapa 1: especificar detalhes do pipeline 23
Etapa 2: Escolher fórmula 24

Etapa 3: Definir a configuração da infraestrutura - opcional ... 27
Etapa 4: definir as configurações de distribuição - opcional ... 28
Etapa 5: Revisão ... 28
Etapa 6: limpar ... 28
Assistente de pipeline: criar imagem de contêiner ... 31
Etapa 1: especificar detalhes do pipeline ... 31
Etapa 2: Escolher fórmula ... 32
Etapa 3: Definir a configuração da infraestrutura - opcional ... 36
Etapa 4: definir as configurações de distribuição - opcional ... 36
Etapa 5: Revisão ... 36
Etapa 6: limpar ... 37
Componente personalizado com parâmetros ... 39
Use parâmetros no documento do componente do YAML ... 40
Definir parâmetros do componente em uma fórmula do Image Builder com o console ... 44
Use um parâmetro de imagem base em sua receita ... 45
Etapa 1: Localizar ou criar um parâmetro do Parameter Store ... 46
Etapa 2: configurar as permissões do IAM (opcional) ... 46
Etapa 3: criar uma receita de imagem que use o parâmetro ... 48
Componentes ... 49
Listar e visualizar componentes ... 51
Listar componentes do Image Builder ... 52
Listar as versões de construção de componentes do AWS CLI ... 54
Obtenha detalhes do componente no AWS CLI ... 55
Obtenha detalhes da política de componentes no AWS CLI ... 55
Use AWS Marketplace componentes ... 55
Descubra os AWS Marketplace componentes ... 56
Inscrever-se em AWS Marketplace componentes ... 58
Use AWS Marketplace componentes em uma receita ... 58
Usar componentes gerenciados ... 59
Instalação de aplicativo Windows pelo pacote de distribuidor ... 59
Hardening do CIS ... 64
Componentes de fortalecimento do STIG ... 65
Desenvolver componentes personalizados ... 111
Crie um documento do componente do YAML ... 112
Criar componentes personalizados com o Image Builder ... 116
AWSTOE aplicativo gerenciador de componentes ... 127

AWSTOE downloads ... 128
Regiões do compatíveis ... 129
Referência de comando ... 131
Configuração manual ... 137
Usar o framework de documentos de componente ... 151
Módulos de ação ... 204
Configurar entrada ... 320
Recursos de imagens ... 325
Listar versões de compilação e imagens ... 325
Listar imagens ... 326
Listar imagens que estão aguardando ação ... 332
Listar versões da compilação de imagem ... 334
Visualizar os detalhes do recurso de imagem ... 337
Ver os detalhes da imagem no console do Image Builder ... 338
Obtenha detalhes da política de imagem no AWS CLI ... 346
Criar imagens personalizadas com o Image Builder ... 346
Cancelar a criação da imagem a partir do AWS CLI ... 348
Importar e exportar imagens de VM ... 349
Importar uma VM para o Image Builder ... 349
Distribua discos VM de sua imagem, crie a partir do AWS CLI ... 354
Importar imagens de disco ISO ... 355
Sistemas operacionais compatíveis para importação de imagem de disco ISO ... 355
Pré-requisitos ... 355
Importar uma imagem de disco ISO para o Image Builder ... 357
Execute uma instância a partir da AMI de saída ... 361
Gerenciar descobertas de segurança ... 361
Configurar verificações de segurança ... 362
Gerenciar descobertas de segurança ... 364
Limpar recursos do Image Builder ... 366
Gerenciar o ciclo de vida de imagem ... 367
Pré-requisitos ... 368
Criar um perfil do IAM para o gerenciamento do ciclo de vida do Image Builder ... 369
Criar um perfil do IAM para o gerenciamento do ciclo de vida entre contas do Image Builder ... 370
Listar políticas de ciclo de vida ... 372
Visualizar políticas de ciclo de vida ... 375

Visualizar os detalhes da política de ciclo de vida no console do Image Builder 375
Criar políticas de ciclo de vida 378
Criar políticas de ciclo de vida (imagem de AMI) 378
Criar políticas de ciclo de vida (imagem de contêiner) 381
Como as regras do ciclo de vida funcionam 384
Regras de exclusão do ciclo de vida da AMI 385
Visualizar os detalhes das regras de gerenciamento do ciclo de vida 386
Configurar imagens personalizadas 387
Fórmulas 387
Liste e visualize fórmulas de imagem 388
Liste e visualize fórmulas de contêiner 390
Criar uma nova versão de uma fórmula de imagem 392
Criar uma nova versão de fórmula de contêiner 406
Limpar recursos 416
Configurações de infraestrutura 416
Listar e visualizar as configurações de infraestrutura 418
Criar uma configuração de infraestrutura 419
Atualizar uma configuração de infraestrutura 424
AWS PrivateLink Endpoints de VPC 426
Configurações de distribuição 432
Listar e visualizar configurações de distribuição 434
Criar e atualizar a distribuição da AMI 435
Criar e atualizar a distribuição de imagens de contêiner 450
Configurar a distribuição da AMI entre contas 454
Especificar um modelo de execução da AMI 462
Use a distribuição aprimorada de AMI 465
Compartilhar recursos 468
Proprietários de recurso 469
Pré-requisitos para compartilhar recursos do Image Builder 469
Consumidores de recursos 470
Criar o compartilhamento de um recurso 470
Opção 1: criar um compartilhamento de recursos de RAM 471
Opção 2: aplicar uma política de recursos e promover um compartilhamento de recursos existente 471
Cancelar o compartilhamento de um recurso 474
Marcar recursos 475

Marque um recurso do AWS CLI 476
Desmarque um recurso do AWS CLI 476
Liste todas as tags de um recurso específico do AWS CLI 477
Excluir recursos 478
Console: excluir recursos 478
Excluir recursos (AWS CLI) 480
Fluxos de trabalho de imagem 482
Framework do fluxo de trabalho: estágios 483
Acesso ao serviço 483
Use fluxos de trabalho gerenciados 484
Listar fluxos de trabalho de imagem 484
Criar um fluxo de trabalho de imagem 488
Criar um documento de fluxo de trabalho do YAML 491
Estrutura de um documento de fluxo de trabalho do YAML 491
Ações de etapa 504
Variáveis dinâmicas 547
Instruções condicionais 553
Gerenciar pipelines 559
Configurar a execução do pipeline 560
Desativar automaticamente um pipeline com defeito 560
Configurar o registro do pipeline 561
Execute o pipeline manualmente 562
Usar expressões cron 563
Indicar e visualizar pipelines 569
Listar pipelines de imagens do AWS CLI 569
Obtenha detalhes do pipeline de imagens no AWS CLI 569
Criar e atualizar pipelines (AMI) 570
Crie um pipeline de AMI a partir do AWS CLI 571
Atualizar pipeline com o console 573
Atualize o pipeline a partir do AWS CLI 577
Criar e atualizar pipelines (contêiner) 578
Crie um pipeline a partir do AWS CLI 579
Atualizar pipeline com o console 581
Atualize o pipeline a partir do AWS CLI 585
Configurar fluxos de trabalho do pipeline 587
Definir grupos de teste para fluxos de trabalho de teste 587

Definir parâmetros de fluxo de trabalho em um pipeline do Image Builder com o console 588
Especificar o perfil de serviço do IAM que o Image Builder usará para executar ações de fluxo de trabalho 589
EventBridge Regras de uso 589
EventBridge termos 590
Veja EventBridge as regras para seu pipeline do Image Builder 591
Use EventBridge regras para agendar a construção de um pipeline 592
Integrar produtos e serviços 594
Amazon EventBridge 597
Mensagens de eventos que o Image Builder envia 598
Amazon Inspector 601
AWS Marketplace 602
AWS Marketplace assinaturas no Image Builder 603
Descubra produtos de AWS Marketplace imagem no console do Image Builder 604
Use um produto de AWS Marketplace imagem nas receitas do Image Builder 606
Amazon Simple Notification Service 607
Tópicos do SNS criptografado 608
Formato da mensagem SNS 609
Programas de conformidade 615
Monitore eventos e logs 617
CloudTrail troncos 617
Informações do Image Builder em CloudTrail 617
CloudWatch Registros 619
Segurança no Image Builder 621
Proteção de dados 622
Gerenciamento de criptografia e chaves 623
Armazenamento de dados 629
Privacidade do tráfego entre redes 629
Gerenciamento de Identidade e Acesso 629
Público 630
Autenticação com identidades 630
Como o Image Builder utiliza políticas e perfis do IAM 630
Gerencie perímetros de dados 641
Políticas baseadas em identidade 642
Permissões personalizadas de fluxo de trabalho 645
Políticas baseadas em recurso 647

Políticas gerenciadas 648
Perfis vinculados ao serviço 664
Solução de problemas 666
Validação de conformidade 668
Resiliência 669
Segurança da infraestrutura 670
Gerenciamento de patches 671
Práticas recomendadas 673
Limpeza necessária após a construção 674
Substitua o script de limpeza do Linux 685
Solução de problemas do Image Builder 689
Solucionar problemas em compilação de pipelines 689
Cenários de solução de problemas 691
Histórico do documento 697
..... dccxiv

O que é o Image Builder?

EC2 O Image Builder é totalmente gerenciado AWS service (Serviço da AWS) que ajuda você a automatizar a criação, o gerenciamento e a implantação de imagens personalizadas, seguras e de up-to-date servidor. Você pode usar o Console de gerenciamento da AWS, AWS Command Line Interface, ou APIs para criar imagens personalizadas em seu Conta da AWS.

Você é o proprietário das imagens personalizadas que o Image Builder cria em sua conta. Você pode configurar pipelines para automatizar as atualizações e a correção do sistema para as imagens de sua propriedade. Você também pode executar um comando independente para criar uma imagem com os recursos de configuração que você definiu.

O assistente de pipeline do Image Builder pode guiá-lo pelas etapas para criar uma imagem personalizada, da seguinte forma:

Etapa 1: especificar detalhes do pipeline

- Dê um nome ao seu pipeline e adicione tags.
- Defina as configurações de verificação de metadados e vulnerabilidades.
- Defina um cronograma para o seu pipeline.

Etapa 2: personalizar sua imagem

Você pode selecionar uma fórmula existente ou criar uma.

- Escolha uma imagem de base para suas personalizações.
- Adicione ou remova um software da sua imagem de base.
- Personalize configurações e scripts com componentes de compilação.
- Selecione os componentes de teste a serem executados.

Etapa 3: definir seu fluxo de trabalho

Um fluxo de trabalho de imagem define a sequência de etapas que o Image Builder executa durante os estágios de compilação e teste do processo de criação de imagens. Isso faz parte do framework do fluxo de trabalho do Image Builder.

Etapa 4: configurar a infraestrutura de compilação

- Selecione um perfil do IAM para associar ao perfil de instância referente a instâncias que o Image Builder executa durante o processo de criação de imagens.
- Selecione um ou mais tipos de instância que podem ser aplicados na inicialização.
- Selecione um tópico do Amazon Simple Notification Service (SNS) para receber notificações do Image Builder.
- Especifique a VPC, a sub-rede e grupos de segurança que se aplicam ao processo de criação de imagens.
- Selecione as configurações de solução de problemas, como onde o Image Builder grava os logs e se deseja encerrar a instância de compilação em caso de falha (padrão) ou mantê-la em execução para solucionar problemas adicionais.

Etapa 5: definir a distribuição da imagem

- Selecione Regiões da AWS onde o Image Builder distribuirá sua Amazon Machine Image (AMI) ou imagem de contêiner.
- Se o pipeline do Image Builder criar uma AMI, o Image Builder também permitirá a seguinte configuração:
  - Selecione uma chave do KMS para criptografia.
  - Configure o compartilhamento de AMI Contas da AWS entre organizações.
  - Associe uma licença autogerenciada do License Manager à imagem distribuída.
  - Configure um modelo de execução para sua imagem.

Recursos do Image Builder

EC2 O Image Builder fornece os seguintes recursos:

Aumente a produtividade e reduza as operações de construção em conformidade e imagens up-to-date

O Image Builder reduz a quantidade de trabalho envolvido na criação e no gerenciamento de imagens em escala ao automatizar seus pipelines de compilação. Você pode automatizar suas compilações fornecendo sua preferência de cronograma de execução de compilações. A automação

reduz o custo operacional de manter seu software com os patches mais recentes do sistema operacional.

Aumentar o tempo de atividade do serviço

O Image Builder fornece acesso aos componentes de teste que você pode usar para testar suas imagens antes da implantação. Você também pode criar componentes de teste personalizados com AWS Task Orchestrator and Executor (AWSTOE) e usá-los. O Image Builder distribuirá sua imagem somente se todos os testes configurados tiverem êxito.

Aumentar o nível de segurança para implantações

O Image Builder permite criar imagens que removem a exposição desnecessária às vulnerabilidades de segurança dos componentes. Você pode aplicar configurações AWS de segurança para criar out-of-the-box imagens seguras que atendam aos critérios de segurança internos e do setor. O Image Builder também fornece coleções de configurações para empresas em setores regulamentados. Você pode usar essas configurações para ajudá-lo a compilar imagens compatíveis com os padrões do STIG de forma rápida e fácil. Para obter uma lista completa dos componentes do STIG disponíveis por meio do Image Builder, consulte Componentes de hardening do STIG gerenciados pela Amazon para o Image Builder.

Aplicação centralizada e rastreamento de linhagem

Usando integrações integradas com AWS Organizations, o Image Builder permite que você aplique políticas que restringem as contas a executar instâncias somente a partir das aprovadas. AMIs

Compartilhamento simplificado de recursos entre Contas da AWS

EC2 O Image Builder se integra com AWS Resource Access Manager (AWS RAM) para permitir que você compartilhe determinados recursos com qualquer um Conta da AWS ou por meio AWS Organizations dele. EC2 Os recursos do Image Builder que podem ser compartilhados são:

- Componentes
- Imagens
- Fórmulas de imagem
- Fórmulas de contêiner

Para obter mais informações, consulte Compartilhe recursos do Image Builder com AWS RAM.

Sistemas operacionais compatíveis

O Image Builder é compatível com as seguintes versões de sistemas operacionais:

Sistema operacional/distribuição	Versões compatíveis
Amazon Linux	2 e 2023
CentOS	7 e 8
CentOS Stream	8
macOS	12,x (Monterey), 13,x (Ventura), 14,x (Sonoma), 15,x (Sequoia), 26,x (Tahoe)
Red Hat Enterprise Linux (RHEL)	7, 8, 9 e 10
SUSE Linux Enterprise Server (SUSE)	12, 15 e 16
Ubuntu	18.04 LTS, 20.04 LTS, 22.04 LTS e 24.04 LTS
Windows Server	2012 R2, 2016, 2019, 2022 e 2025

Formatos de imagem compatíveis

Para suas imagens personalizadas que criam uma imagem de máquina da Amazon (AMI), você pode escolher uma AMI existente como ponto de partida. Para imagens de contêiner do Docker, você pode escolher entre imagens públicas hospedadas em DockerHub, imagens de contêiner existentes no Amazon ECR ou imagens de contêiner gerenciadas pela Amazon como ponto de partida.

Cotas padrão

Para ver as cotas padrão do Image Builder, consulte Endpoints and Quotas do Image Builder.

AWS Regiões e endpoints

Para ver os endpoints de serviço do Image Builder, consulte Endpoints e cotas do Image Builder.

Conceitos

Os termos e conceitos a seguir são fundamentais para sua compreensão e uso do EC2 Image Builder.

AMI

Uma Amazon Machine Image (AMI) é a unidade básica de implantação na Amazon EC2 e é um dos tipos de imagens que você pode criar com o Image Builder. Uma AMI é uma imagem de máquina virtual pré-configurada que contém o sistema operacional (OS) e o software pré-instalado para EC2 implantar instâncias. Para obter mais informações, consulte Imagens de máquina da Amazon (AMIs).

Pipeline de imagens

Um pipeline de imagens fornece uma estrutura de automação para criar imagens seguras AMIs e de contêiner AWS. O pipeline de imagens do Image Builder está associado a uma fórmula de imagem ou fórmula de contêiner que define a compilação, a validação e as fases de teste para o ciclo de vida da compilação de um imagem.

Um pipeline de imagens pode ser associado a uma configuração de infraestrutura que define onde a imagem é criada. É possível definir atributos, como tipo de instância, sub-redes, grupos de segurança, registros em log e outras configurações relacionadas à infraestrutura. Você também pode associar o pipeline de imagens a uma configuração de distribuição para definir como gostaria de implantar a imagem.

Imagem gerenciada

Uma imagem gerenciada é um recurso no Image Builder que consiste em uma AMI ou imagem de contêiner, além de metadados, como versão e plataforma. A imagem gerenciada é usada pelos pipelines do Image Builder para determinar qual imagem de base usar para a compilação. Neste guia, as imagens gerenciadas às vezes são chamadas de “imagens”; no entanto, uma imagem não é o mesmo que uma AMI.

Fórmula da imagem

Uma fórmula de imagem do Image Builder é um documento que define a imagem de base e os componentes que são aplicados à imagem de base para produzir a configuração desejada para a imagem AMI de saída. É possível usar uma fórmula de imagem para duplicar compilações. As receitas de imagens do Image Builder podem ser compartilhadas, ramificadas e editadas usando o assistente do console AWS CLI, o ou a API. É possível usar fórmulas de imagem com seu software de controle de versão para manter fórmulas de imagem compartilháveis e com versionamento.

Fórmula do contêiner

Uma fórmula de contêiner do Image Builder é um documento que define a imagem de base e os componentes que são aplicados à imagem de base para produzir a configuração desejada para a imagem de contêiner de saída. É possível usar uma fórmula de contêiner para duplicar compilações. É possível compartilhar, ramificar e editar fórmulas de imagem do Image Builder usando o assistente do console, o AWS CLI ou a API. É possível usar fórmulas de contêiner com seu software de controle de versão para manter fórmulas de contêiner compartilháveis e com versionamento.

Imagem de base

A imagem de base é a imagem e o sistema operacional selecionados usados em seu documento de fórmula de imagem ou de contêiner, junto com os componentes. A imagem de base e as definições do componente combinadas produzem a configuração desejada para a imagem de saída.

Componentes

Um componente define a sequência de etapas necessárias para personalizar uma instância antes da criação da imagem (um componente de compilação) ou para testar uma instância que foi inicializada a partir da imagem criada (um componente de teste).

Um componente é criado a partir de um documento YAML ou JSON declarativo em texto simples que descreve a configuração do runtime para compilar e validar ou testar uma instância produzida pelo seu pipeline. Os componentes são executados na instância usando um aplicativo de gerenciamento de componentes. O aplicativo de gerenciamento de componentes analisa os documentos e executa as etapas desejadas.

Depois de criados, um ou mais componentes são agrupados usando uma fórmula de imagem ou uma fórmula de contêiner para definir o plano de compilação e teste de uma imagem de contêiner ou máquina virtual. Você pode usar componentes públicos que são de propriedade e gerenciados por AWS, ou você pode criar seus próprios. Para obter mais informações sobre componentes, consulte Como o Image Builder usa o AWS Task Orchestrator and Executor aplicativo para gerenciar componentes.

Documento do componente

Um documento YAML ou JSON declarativo em texto simples que descreve a configuração de uma personalização que você pode aplicar à sua imagem. O documento é usado para criar um componente de compilação ou teste.

Estágios de runtime

EC2 O Image Builder tem dois estágios de execução: compilação e teste. Cada estágio de runtime tem uma ou mais fases com a configuração definida pelo documento do componente.

Fases de configuração

A lista a seguir mostra as fases que são executadas durante os estágios de compilação e teste:

Estágio de compilação:

Fase da compilação

> Um pipeline de imagem começa com a fase de compilação do estágio de compilação quando ele é executado. A imagem de base é baixada e a configuração especificada para a fase de compilação do componente é aplicada para compilar e inicializar uma instância.

Fase de validação

> Depois que o Image Builder inicializar a instância e aplicar todas as personalizações da fase de compilação, a fase de validação começa. Durante essa fase, o Image Builder garante que todas as personalizações funcionem conforme o esperado, com base na configuração que o componente especifica para a fase de validação. Se a validação da instância for bem-sucedida, o Image Builder interrompe a instância, cria uma imagem e continua até o estágio de teste.

Estágio de teste:

Fase de teste

> Durante essa fase, o Image Builder inicializa uma instância a partir da imagem que ele criou após a conclusão bem-sucedida da fase da validação. O Image Builder executa componentes de teste durante essa fase para verificar se a instância está íntegra e funciona conforme o esperado.

Fase de teste do host do contêiner

> Depois que o Image Builder executa a fase de teste para todos os componentes selecionados na fórmula do contêiner, o Image Builder executa esta fase para fluxos de trabalho do contêiner. A fase de teste do host do contêiner pode executar testes adicionais que validam o gerenciamento de contêineres e as configurações de runtime personalizadas.

Fluxo de trabalho

Os fluxos de trabalho definem a sequência de etapas que o Image Builder realiza ao criar uma nova imagem. Todas as imagens têm fluxos de trabalho de criação, teste e distribuição.

Tipo de fluxo de trabalho

`BUILD`

Abrange a configuração do estágio de compilação para cada imagem criada.

`TEST`

Abrange a configuração do estágio de teste para cada imagem criada.

`DISTRIBUTION`

Abrange a configuração do estágio de distribuição para cada imagem criada.

Preços

Não há custo para usar o EC2 Image Builder para criar imagens personalizadas de AMI ou contêiner. No entanto, o preço padrão se aplica a outros serviços que são usados no processo. A lista a seguir inclui o uso de algumas Serviços da AWS que podem gerar custos quando você cria, cria, armazena e distribui sua AMI personalizada ou imagens de contêiner, dependendo da sua configuração.

- Lançamento de uma EC2 instância
- Armazenar logs no Amazon S3
- Validar imagens com o Amazon Inspector
- Armazenando snapshots do Amazon EBS para seu AMIs
- Armazenar imagens de contêiner no Amazon ECR
- Enviar imagens de contêiner para o Amazon ECR e extrair imagens de contêiner do Amazon ECR
- Se o Systems Manager Advanced Tier estiver ativado e as EC2 instâncias da Amazon forem executadas com ativação local, você poderá ser cobrado pelos recursos por meio do Systems Manager

Relacionado Serviços da AWS

EC2 O Image Builder usa outros Serviços da AWS para criar imagens, dependendo da configuração da receita do Image Builder. Para obter mais informações sobre a integração de produtos e serviços para suas imagens personalizadas, consulte Integrar produtos e serviços no Image Builder.

Como funciona o EC2 Image Builder

Quando você usa o console do EC2 Image Builder para criar um pipeline de imagem personalizado, o sistema orienta você pelas etapas a seguir.

1. Especifique os detalhes do pipeline: insira informações sobre seu pipeline, como nome, descrição, tags e um cronograma para executar compilações automatizadas. Se preferir, você pode escolher construções manuais.
2. Escolha a fórmula: escolha entre criar uma AMI ou criar uma imagem de contêiner. Para os dois tipos de imagens de saída, você insere um nome e uma versão para sua fórmula, seleciona uma imagem base e escolhe os componentes a serem adicionados para criação e teste. Você também pode escolher o controle de versão automático para garantir que sempre use a versão mais recente disponível do sistema operacional (SO) para sua imagem base. Além disso, as fórmulas de contêiner definem Dockerfiles e o repositório Amazon ECR de destino para sua imagem de contêiner Docker de saída.

   > **Note**
   >
   > Os componentes são os blocos de construção que são consumidos por uma fórmula de imagem ou uma fórmula de recipiente. Por exemplo, pacotes para instalação, etapas de fortalecimento da segurança e testes. A imagem base e os componentes selecionados formam uma fórmula de imagem.

3. Defina a configuração da infraestrutura — O Image Builder inicia EC2 instâncias em sua conta para personalizar imagens e executar testes de validação. As configurações da infraestrutura especificam os detalhes da infraestrutura para as instâncias que serão executadas na sua Conta da AWS durante o processo de criação.
4. Defina as configurações de distribuição: escolha as AWS regiões para as quais distribuir sua imagem após a conclusão da compilação e a aprovação em todos os testes. O pipeline distribui automaticamente sua imagem para a região em que executa a compilação, e você pode adicionar distribuição de imagens para outras regiões.

As imagens que você cria a partir da sua imagem base personalizada estão no seu Conta da AWS. Você pode configurar seu pipeline de imagens para produzir versões atualizadas e corrigidas de sua imagem inserindo um cronograma de construção. Quando a compilação for concluída, você poderá receber uma notificação por meio do Amazon Simple Notification Service (SNS). Além de produzir uma imagem final, o assistente do console do Image Builder gera uma receita que pode ser usada

com sistemas de controle de versão e pipelines de integration/continuous implantação contínua (CI/CD) existentes para automação repetível. Você pode compartilhar e criar novas versões da sua fórmula.

Conteúdo da seção

- Elementos da AMI
- Gerenciamento de componentes
- Recursos criados
- Distribuição
- Compartilhar recursos
- Compliance

Elementos da AMI

Uma Amazon Machine Image (AMI) é uma imagem de máquina virtual (VM) pré-configurada que contém o sistema operacional e o software para EC2 implantar instâncias.

Uma AMI inclui os seguintes elementos:

- Um modelo para o volume raiz da VM. Quando você inicia uma Amazon EC2 VM, o volume do dispositivo raiz contém a imagem para inicializar a instância. Quando o armazenamento de instâncias é usado, o dispositivo raiz é um volume de armazenamento de instâncias criado em um modelo no Amazon S3. Para obter mais informações, consulte Amazon EC2 Root Device Volume.
- Quando o Amazon EBS é usado, o dispositivo raiz é um volume do EBS criado em um snapshot do EBS.
- Permissões de lançamento que determinam o Contas da AWS que pode ser iniciado VMs com a AMI.
- Dados de mapeamento de dispositivos de blocos que especifica os volumes a serem anexados à instância quando ela for executada.
- Um identificador de recurso exclusivo para cada região, para cada conta.
- Cargas de metadados, como tags, e propriedades, como região, sistema operacional, arquitetura, tipo de dispositivo raiz, provedor, permissões de inicialização, armazenamento para o dispositivo raiz e status de assinatura.
- Uma assinatura AMI para imagens do Windows para proteção contra adulteração não autorizada. Para obter mais informações, consulte Documentos de identidade da instância.

Gerenciamento de componentes

EC2 O Image Builder usa um aplicativo de gerenciamento de componentes AWS Task Orchestrator and Executor (AWSTOE) que ajuda você a orquestrar fluxos de trabalho complexos, modificar configurações do sistema e testar seus sistemas com componentes de script baseados em YAML. Por ser AWSTOE um aplicativo independente, ele não requer nenhuma configuração adicional. Pode ser executado em qualquer infraestrutura de nuvem e no local. Para começar a usar AWSTOE como um aplicativo independente, consulteConfiguração manual para desenvolver componentes personalizados com AWSTOE.

O Image Builder usa AWSTOE para realizar todas as atividades na instância. Isso inclui criar e validar sua imagem antes de tirar um snapshot e testar o snapshot para garantir que ele funcione conforme o esperado antes de criar a imagem final. Para obter mais informações sobre como o Image Builder usa AWSTOE para gerenciar seus componentes, consulteUsar componentes para personalizar sua imagem do Image Builder. Para obter mais informações sobre a criação de componentes com AWSTOE, consulte Como o Image Builder usa o AWS Task Orchestrator and Executor aplicativo para gerenciar componentes.

Teste de imagem

Você pode usar componentes de AWSTOE teste para validar sua imagem e garantir que ela funcione conforme o esperado, antes de criar a imagem final.

Geralmente, cada componente de teste consiste em um documento YAML que contém um script de teste, um binário de teste e metadados de teste. O script de teste contém os comandos de orquestração para iniciar o binário de teste, que pode ser escrito em qualquer linguagem compatível com o sistema operacional. Os códigos de status de saída indicam o resultado do teste. Os metadados de teste descrevem o teste e seu comportamento; por exemplo, o nome, a descrição, os caminhos para testar o binário e a duração esperada.

Recursos criados

Quando você cria um pipeline, nenhum recurso externo ao Image Builder é criado, a menos que o seguinte seja verdadeiro:

- Quando uma imagem é criada por meio do cronograma do pipeline
- Quando você escolhe Executar pipeline no menu Ações no console do Image Builder
- Quando você executa um desses comandos a partir da API ou AWS CLI: StartImagePipelineExecution ou CreateImage

Os seguintes recursos são criados durante o processo de criação da imagem:

Pipelines de imagens da AMI

- EC2 instância (temporária)
- Systems Manager Inventory Association (por meio do Systems Manager State Manager, se `EnhancedImageMetadata` estiver habilitado) na EC2 instância
- Amazon EC2 AMI
- O Amazon EBS Snapshot associado ao Amazon AMI EC2

Pipelines de imagens de contêiner

- Contêiner Docker em execução em uma EC2 instância (temporário)
- A Systems Manager Inventory Association (por meio do Systems Manager State Manager) `EnhancedImageMetadata` está ativada) na EC2 instância
- Imagem do contêiner do docker
- Dockerfile

Depois que a imagem for criada, todos os recursos temporários serão excluídos.

Distribuição

EC2 O Image Builder pode distribuir AMIs ou armazenar imagens em qualquer AWS região. A imagem é copiada para cada região especificada na conta usada para criar a imagem.

Para imagens de saída da AMI, você pode definir as permissões de execução da AMI para controlar quais Contas da AWS têm permissão para executar EC2 instâncias com a AMI criada. Por exemplo, você pode tornar a imagem privada, pública ou compartilhar com contas específicas. Se você distribuir a AMI para outras regiões e definir permissões de execução para outras contas, as permissões de execução serão propagadas para todas as regiões nas quais a AMI é distribuída. AMIs

Você também pode usar sua AWS Organizations conta para impor limitações às contas dos membros para iniciar instâncias somente com instâncias aprovadas e em conformidade AMIs. Para obter mais informações, consulte Gerenciando o Contas da AWS em sua organização.

Para atualizar suas configurações de distribuição usando o console do Image Builder, siga as etapas para Criar uma versão de uma fórmula de imagem com o console, ou Crie uma nova versão de fórmula de contêiner com o console.

Compartilhar recursos

Para compartilhar componentes, receitas ou imagens com outras contas ou dentro delas AWS Organizations, consulteCompartilhe recursos do Image Builder com AWS RAM.

Compliance

Para os benchmarks do Center for Internet Security (CIS), o EC2 Image Builder usa o Amazon Inspector para realizar avaliações de exposição, vulnerabilidades e desvios das melhores práticas e padrões de conformidade. Por exemplo, o Image Builder avalia a acessibilidade não intencional da rede, a conectividade pública sem patches com CVEs a Internet e a ativação remota do login raiz. O Amazon Inspector é oferecido como um componente de teste que você pode escolher adicionar à sua fórmula de imagem. Para obter mais informações sobre o Amazon Inspector, consulte o Guia do usuário do Amazon Inspector. Para obter mais informações, consulte Center for Internet Security (CIS) Benchmarks.

O Image Builder fornece componentes de fortalecimento do STIG para ajudá-lo a compilar, de forma mais eficiente, imagens compatíveis para os padrões do STIG de base. Esses componentes do STIG verificam se há configurações incorretas e executam um script de correção. Não há cobranças adicionais pelo uso de componentes em conformidade com o STIG. Para obter uma lista completa dos componentes do STIG disponíveis por meio do Image Builder, consulte Componentes de hardening do STIG gerenciados pela Amazon para o Image Builder.

Versionamento semântico no Image Builder

O Image Builder usa versionamento semântico para organizar recursos e garantir que eles sejam exclusivos. IDs Essa versão semântica tem quatro nós:

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

É possível atribuir valores para os três primeiros e filtrar todos eles.

O controle de versionamento semântico está incluído no nome do recurso da Amazon (ARN) de cada objeto, no nível que se aplica a esse objeto da seguinte forma:

1. Versionless ARNs e Name ARNs não incluem valores específicos em nenhum dos nós. Os nós são totalmente excluídos ou são especificados como curingas, por exemplo: x.x.x.
2. A versão ARNs tem apenas os três primeiros nós:<major>. <minor>. <patch>
3. A versão de compilação ARNs tem todos os quatro nós e aponta para uma construção específica de uma versão específica de um objeto.

Atribuição: Para os primeiros três nós, você pode atribuir qualquer valor inteiro positivo, ou zero, com um limite superior de 2^30-1, ou 1073741823 para cada nó. O Image Builder atribui o número da compilação automaticamente ao quarto nó.

Padrão: é possível utilizar qualquer padrão numérico que atenda aos requisitos de atribuição para os nós que você pode atribuir. Por exemplo, você pode escolher um padrão de versão de software, como 1.0.0, ou uma data, como 2021.01.01.

Seleção: com o versionamento semântico, você tem a flexibilidade de utilizar curingas (x) para especificar as versões ou os nós mais recentes ao selecionar a imagem base ou os componentes para sua fórmula. Quando um curinga é usado em qualquer nó, todos os nós à direita do primeiro curinga também devem ser curingas.

Por exemplo, dadas as seguintes versões recentes: 2.2.4, 1.7.8 e 1.6.8, a seleção de versão usando curingas produz os seguintes resultados:

- `x.x.x` = 2.2.4
- `1.x.x` = 1.7.8
- `1.6.x` = 1.6.8
- `x.2.x` não é válido e produz um erro
- `1.x.8` não é válido e produz um erro

Usando referências de versão

As referências de versão são cadeias de caracteres ready-to-use ARN que incorporam padrões curinga com base na versão semântica do recurso que você criou ou recuperou. Em vez de escrever código personalizado para analisar ARNs e inserir curingas, o Image Builder faz esse trabalho para você.

Quando você cria ou recupera recursos do Image Builder, o Image Builder fornece automaticamente padrões de versão pré-construídos ARNs com caracteres curinga no objeto.

`latestVersionReferences` Isso elimina a necessidade de analisar e reconstruir manualmente ARNs quando você deseja referenciar recursos usando padrões de versão curinga.

Por exemplo, quando você cria um componente com versão`1.2.3`, o Image Builder retorna:

```
{
    "componentBuildVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/
my-component/1.2.3/1",
    "latestVersionReferences": {
        "latestVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/my-
component/x.x.x",
        "latestMajorVersionArn": "arn:aws:imagebuilder:us-
west-2:123456789012:component/my-component/1.x.x",
        "latestMinorVersionArn": "arn:aws:imagebuilder:us-
west-2:123456789012:component/my-component/1.2.x",
        "latestPatchVersionArn": "arn:aws:imagebuilder:us-
west-2:123456789012:component/my-component/1.2.3"
    }
}
```

Padrões de referência de versão disponíveis

O `latestVersionReferences` objeto contém quatro padrões de ARN:

- latestVersionArn (x.x.x) - Sempre resolve para a versão mais recente absoluta.
- atestMajorVersionArn (1.x.x) - Resolve para as versões secundárias e de patch mais recentes dentro de uma versão principal.
- latestMinorVersionArn (1.2.x) - Vai para a versão mais recente do patch em uma versão secundária específica.
- latestPatchVersionArn (1.2.3) - Faz referência a uma versão semântica específica e resolve para a versão de compilação mais recente para recursos que oferecem suporte a várias versões de compilação.

Recursos que retornam referências de versão

As referências de versão são retornadas por `Create` e `Get` APIs para todos os recursos do Image Builder versionados:

- Componentes -`CreateComponent`, `GetComponent`
- Receitas de imagens -`CreateImageRecipe`, `GetImageRecipe`
- Receitas de contêineres -`CreateContainerRecipe`, `GetContainerRecipe`
- Imagens -`CreateImage`, `GetImage`
- Fluxos de trabalho -`CreateWorkflow`, `GetWorkflow`

Nota: Para fluxos de trabalho gerenciados pelo Image Builder, somente `latestVersionArn (x.x.x)` é retornado, pois o Image Builder exige que você sempre use a versão mais recente dos fluxos de trabalho gerenciados pelo Image Builder.

Configurar compilação de imagens personalizadas com o Image Builder

Antes de criar imagens com o EC2 Image Builder, verifique se você atendeu aos seguintes pré-requisitos para criar um pipeline de imagens. Salvo indicação específica em contrário, esses pré-requisitos são necessários para todos os tipos de pipeline.

Pré-requisitos

- Perfil vinculado ao serviço do Image Builder
- Requisitos de configuração
- Repositório de contêiner para pipelines de imagens de contêiner
- Host dedicado para imagens do macOS
- Pré-requisitos do IAM
- Pré-requisitos do Systems Manager Agent

Depois de atender aos pré-requisitos, você pode gerenciar o EC2 Image Builder a partir de qualquer uma das interfaces a seguir.

- EC2 Console Image Builder
- Comandos do Image Builder no AWS CLI
- EC2 Referência da API Image Builder
- AWS SDKs e ferramentas

Perfil vinculado ao serviço do Image Builder

EC2 O Image Builder usa uma função vinculada ao serviço para conceder permissões a outros AWS serviços em seu nome. Não é necessário criar manualmente um perfil vinculado ao serviço. Quando você cria seu primeiro recurso do Image Builder no AWS Management Console, no AWS CLI, ou na AWS API, o Image Builder cria a função vinculada ao serviço para você. Para obter mais informações sobre o perfil vinculado ao serviço criada pelo Image Builder na sua conta, consulte Usar perfis do IAM vinculados ao serviço para o Image Builder.

Requisitos de configuração

- O Image Builder é compatível com AWS PrivateLink. Para obter mais informações sobre como configurar endpoints da VPC para o Image Builder, consulte Image Builder e AWS PrivateLink interface VPC endpoints.
- As instâncias que o Image Builder usa para criar imagens de contêiner devem ter acesso à Internet para baixá-las AWS CLI do Amazon S3 e baixar uma imagem base do repositório do Docker Hub, se aplicável. O Image Builder usa o AWS CLI para obter o Dockerfile da receita do contêiner, onde ele é armazenado como dados.
- As instâncias que o Image Builder usa para criar imagens e executar testes devem ter acesso ao serviço Systems Manager. Os requisitos de instalação dependem do seu sistema operacional.

  Para ver os requisitos de instalação da imagem base, escolha a guia que corresponde ao sistema operacional da imagem base.

  Linux

  Para instâncias do Amazon EC2 Linux, o Image Builder instala o Systems Manager Agent na instância de compilação, se ainda não estiver presente, e o remove antes de criar a imagem.

  Windows

  O Image Builder não instala o Systems Manager Agent nas instâncias de compilação do Amazon EC2 Windows Server. Se sua imagem base não veio pré-instalada com o Systems Manager Agent, você deve iniciar uma instância a partir da imagem de origem, instalar manualmente o Systems Manager na instância e criar uma nova imagem base a partir da sua instância.

  Para instalar manualmente o agente do Systems Manager em sua instância EC2 do Amazon Windows Server, consulte Instalar manualmente o agente do Systems Manager em EC2 instâncias do Windows Server no Guia AWS Systems Manager do usuário.

Repositório de contêiner para pipelines de imagens de contêiner

Para pipelines de imagens de contêiner, a fórmula define a configuração das imagens do Docker que são produzidas e armazenadas no repositório dos contêineres de destino. Você deve criar o repositório de destino antes de criar a fórmula do contêiner para sua imagem do Docker.

O Image Builder usa o Amazon ECR como seu repositório de destino para imagens de contêineres. Para criar um repositório do Amazon ECR, siga as etapas descritas em Criar um repositório no Manual do usuário do Amazon Elastic Container Registry.

Host dedicado para imagens do macOS

As instâncias do Amazon EC2 Mac exigem um host dedicado executado em um tipo de instância metálica. Antes de criar uma imagem personalizada do macOS, você deve alocar um host dedicado à sua conta. Para obter mais informações sobre instâncias do Mac e uma lista dos tipos de instância que oferecem suporte nativo ao sistema operacional macOS, consulte Instâncias do EC2 Amazon Mac no Guia do usuário da EC2 Amazon.

Depois de criar um host dedicado, você pode definir as configurações no recurso de configuração de infraestrutura da sua imagem. A configuração da infraestrutura inclui propriedades de posicionamento nas quais você pode especificar o host, o grupo de posicionamento do host ou a zona de disponibilidade para onde devem ir as instâncias executadas por meio de sua imagem.

Pré-requisitos do IAM

O perfil do IAM que você associa ao seu perfil de instância precisa ter permissões para executar os componentes de criação e teste incluídos na sua imagem. As seguintes políticas de perfil do IAM devem ser anexadas à função do IAM associada ao perfil de instância:

- EC2InstanceProfileForImageBuilder
- EC2InstanceProfileForImageBuilderECRContainerBuilds
- Amazon SSMManaged InstanceCore

Se você configurar o registro em log, o perfil de instância especificado na configuração da sua infraestrutura deverá ter `s3:PutObject` permissões para o bucket de destino (`arn:aws:s3:::BucketName/*`). Por exemplo:

JSON

```
{
 "Version":"2012-10-17",
 "Statement": [
  {
```

```
    "Sid": "UploadFileToS3Bucket",
    "Effect": "Allow",
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::bucket-name/*"
   }
  ]
}
```

Anexar política

As etapas a seguir orientam você pelo processo de anexar as políticas do IAM a um perfil do IAM para conceder as permissões anteriores.

1. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação à esquerda, escolha Políticas.
3. Filtre a lista de políticas com EC2InstanceProfileForImageBuilder
4. Selecione o bullet ao lado da política e, na lista suspensa Ações de política, selecione Anexar.
5. Selecione o nome do perfil do IAM ao qual a política de mascaramento será anexada.
6. Escolha Anexar política.
7. Repita as etapas de 3 a 6 para as SSMManaged InstanceCore políticas EC2InstanceProfileForImageBuilderECRContainerBuildse da Amazon.

> **Note**
>
> Se quiser copiar uma imagem criada com o Image Builder em outra conta, você deve criar o perfil do `EC2ImageBuilderDistributionCrossAccountRole` em todas as contas de destino e anexar a política gerenciada do Política Ec2ImageBuilderCrossAccountDistributionAccess à função. Para obter mais informações, consulte Compartilhe recursos do Image Builder com AWS RAM.

Pré-requisitos do Systems Manager Agent

EC2 O Image Builder executa o Agente AWS Systems Manager (Systems Manager) nas EC2 instâncias que ele executa para criar e testar sua imagem. O Image Builder coleta informações

adicionais sobre a instância usada durante a fase de compilação com o Systems Manager Inventory. Essas informações incluem o nome e a versão do sistema operacional (SO), bem como a lista de pacotes e suas respectivas versões, conforme relatado pelo seu sistema operacional.

Para optar por não coletar essas informações, selecione o método correspondente ao ambiente de sua preferência:

- Console do Image Builder: desmarque a caixa de seleção Habilitar a coleta de metadados aprimorada.
- AWS CLI: especifique a opção `--no-enhanced-image-metadata-enabled`.
- API Image Builder ou SDKs — Defina o `enhancedImageMetadataEnabled` parâmetro como`false`.

O Image Builder usa `RunCommand` para enviar ações à instância de compilação e teste como parte do fluxo de trabalho de compilação e teste de imagens. Você não pode optar por não usar o `RunCommand` para enviar ações para sua instância de compilação e teste.

Saiba como criar imagens personalizadas com os tutoriais do Image Builder.

Há muitas maneiras de criar imagens e componentes personalizados com o EC2 Image Builder. Os tutoriais ajudam você a aprender sobre os principais conceitos do Image Builder. Cada tutorial apresenta um caso de uso com etapas que você pode seguir pela primeira vez. As instruções usam padrões sempre que possível para ajudar você a aprender o processo geral. Depois de usar um dos tutoriais, você pode explorar outras maneiras de personalizar suas próprias imagens.

Compilar sua primeira imagem

Os tutoriais a seguir mostram como compilar sua primeira imagem com o assistente do console do Image Builder. Ao final do tutorial, você terá criado o conjunto de recursos do Image Builder a seguir. A etapa final do tutorial é limpar os recursos que foram criados.

- Uma fórmula de imagem para sua imagem de máquina da Amazon (AMI) ou uma fórmula de contêiner para sua imagem de contêiner.
- Um recurso de configuração de infraestrutura com configurações padrão.
- Um recurso de configurações de distribuição com configurações padrão que distribui a saída para a região de origem (sua conta na região que você usa para executar o assistente do console).
- Um pipeline de imagens que usa os recursos listados para compilar sua imagem de saída com os fluxos de trabalho padrão de compilação de imagens.
- Uma AMI de saída ou imagem de contêiner.

Tutoriais do assistente do console

- Assistente de pipeline: criar AMI
- Assistente de pipeline: criar imagem de contêiner

Criar um componente personalizado com parâmetros de entrada

O tutorial a seguir mostra como criar um componente personalizado que define os parâmetros de entrada e, em seguida, define os valores da sua fórmula do Image Builder.

Componente personalizado com parâmetros

Use os parâmetros do Systems Manager com o Image Builder

O tutorial a seguir mostra como criar um AWS Systems Manager parâmetro do Parameter Store e usá-lo em uma receita de imagem.

Use um parâmetro de imagem base em sua receita

Você também pode usar os parâmetros do Parameter Store nas configurações de distribuição da AMI para armazenar o ID da imagem de saída e em componentes personalizados. Para obter mais informações, consulte Criar e atualizar as configurações de distribuição da AMI para distribuições e Use os parâmetros do Systems Manager Parameter Store componentes personalizados.

Tutorial: criar um pipeline de imagens com a AMI de saída por meio do assistente do console do Image Builder

Este tutorial explica como criar um pipeline automatizado para criar e manter uma EC2 imagem personalizada do Image Builder usando o assistente de console Create image pipeline. Para ajudá-lo a percorrer as etapas com eficiência, as configurações padrão são usadas, quando disponíveis, e as seções opcionais são ignoradas.

Criar fluxo de trabalho de pipeline de imagens

- Etapa 1: especificar detalhes do pipeline
- Etapa 2: Escolher fórmula
- Etapa 3: Definir a configuração da infraestrutura - opcional
- Etapa 4: definir as configurações de distribuição - opcional
- Etapa 5: Revisão
- Etapa 6: limpar

Etapa 1: especificar detalhes do pipeline

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Para começar a criar seu pipeline, escolha Criar pipeline de imagem.
3. Na seção Geral, insira o nome do seu pipeline (obrigatório).

> **Tip**
>
> Por padrão, a coleta de metadados aprimorada está ativada. Para garantir a compatibilidade entre componentes e imagens básicas, deixe ativada.

4. Na seção Compilar schedule, você pode manter os padrões para as opções de schedule. Observe que a zona horária mostrada para o schedule padrão é Horário Universal Coordenado (UTC). Para obter mais informações sobre o horário UTC e encontrar a diferença do seu fuso horário, consulte Time Zone Abbreviations, Worldwide List (Abreviações de fuso horário: lista mundial).

   Em Configurações de atualização de dependência, escolha a opção Executar pipeline no horário programado se houver a opção de atualizações de dependência. Essa configuração faz com que seu pipeline verifique se há atualizações antes de iniciar a compilação. Se não houver atualizações, ele ignora a compilação programada do pipeline.

   > **Note**
   >
   > Para garantir que seu pipeline reconheça as atualizações e compilações de dependências conforme o esperado, você deve usar o controle de versionamento semântico (x.x.x) para sua imagem base e seus componentes. Para saber mais sobre o versionamento semântico dos recursos do Image Builder, consulte Versionamento semântico no Image Builder.

5. Para prosseguir para a próxima etapa, escolha Avançar.

Etapa 2: Escolher fórmula

1. O padrão do Image Builder é Usar a fórmula existente na seção Fórmula. Porém, na primeira vez, escolha a opção Criar nova fórmula.
2. Na seção Tipo de imagem, escolha a opção imagem de máquina da Amazon (AMI) para criar um pipeline de imagem que produzirá e distribuirá uma AMI.
3. Na seção Geral, insira as seguintes caixas obrigatórias:

   - Nome: o nome da sua fórmula

- Versão: a versão da sua fórmula (use o formato <major>.<minor>.<patch>, onde major, minor e patch são valores inteiros). As novas fórmulas geralmente começam com `1.0.0`.

4. Na seção Imagem de origem, mantenha os valores padrão para Selecionar imagem, Sistema operacional da imagem e Origem da imagem. Isso resulta em uma lista de Linux AMIs que são gerenciados pela Amazon. Para este tutorial, selecione a `Amazon Linux 2 x86` imagem.

   a. No menu suspenso Nome da imagem, escolha uma imagem.

   b. Mantenha o padrão para as opções de controle de versionamento automático (use a versão mais recente disponível do SO).

   > **Note**
   >
   > Essa configuração garante que seu pipeline use versionamento semântico para a imagem base, para detectar atualizações de dependências para trabalhos programados automaticamente. Para saber mais sobre o versionamento semântico dos recursos do Image Builder, consulte Versionamento semântico no Image Builder.

5. Na seção Configuração da instância, mantenha os valores padrão para o Systems Manager Agent. Isso faz com que o Image Builder mantenha o agente do Systems Manager após a conclusão da compilação e dos testes, para incluir o agente do Systems Manager em sua nova imagem.

   Mantenha o campo Dados do usuário em branco para este tutorial. Use esta propriedade para fornecer comandos ou um script de comando a executar na inicialização da instância de compilação. No entanto, ele substitui qualquer comando que o Image Builder possa ter adicionado para garantir que o Systems Manager seja instalado. Quando você usá-lo, certifique-se de que o agente do Systems Manager esteja pré-instalado na imagem de base ou que você inclua a instalação nos dados do usuário.

6. Na seção Componentes, você pode optar por não adicionar nenhum componente e continuar. Se você quiser adicionar componentes, no painel Construir componentes, escolha Adicionar componentes de construção e selecione na lista `Amazon managed` de filtros do proprietário do componente. Isso abre um painel de seleção no lado direito da interface do console, onde você pode navegar e filtrar os componentes que estão disponíveis.

   Para este tutorial, escolha um componente que atualize o Linux com as atualizações de segurança mais recentes, da seguinte forma:

a. Filtre os resultados inserindo a palavra `update` na barra de pesquisa localizada na parte superior do painel.

b. Marque a caixa de seleção para o componente de compilação `update-linux`.

c. Mantenha o padrão para as opções de controle de versão (use a versão mais recente disponível).

> **Note**
>
> Essa configuração garante que seu pipeline use versionamento semântico do componente selecionado, para detectar atualizações de dependências para trabalhos agendados automaticamente. Para saber mais sobre o versionamento semântico dos recursos do Image Builder, consulte Versionamento semântico no Image Builder.

d. Escolha Adicionar à receita para adicionar o componente à sua receita. Isso fecha o painel de seleção de componentes.

e. De volta ao painel Criar componentes, o componente que você adicionou é exibido.

7. Reordenar componentes (opcional)

Se você escolheu mais de um componente para incluir em sua imagem, você pode usar a drag-and-drop ação para reorganizá-los na ordem em que devem ser executados durante o processo de criação.

> **Note**
>
> Os componentes de endurecimento do CIS não seguem as regras padrão de ordenação de componentes nas fórmulas do Image Builder. Os componentes de endurecimento do CIS sempre são executados por último para garantir que os testes de benchmark sejam executados em relação à sua imagem de saída.

a. Repita as etapas anteriores para adicionar o `update-linux-kernel-5` componente à sua receita.

b. O componente que você acabou de adicionar tem um parâmetro de entrada para a versão do kernel. Para expandir as configurações das opções de versionamento ou dos parâmetros de entrada, você pode escolher a seta ao lado do nome da configuração. Para expandir

todas as configurações de todos os componentes selecionados, você pode ativar e desativar a opção Expandir tudo. Para obter mais informações sobre como usar parâmetros de entrada em seus componentes e configurá-los em suas fórmulas, consulte Tutorial: criar um componente personalizado com parâmetros de entrada.

c. Escolha um dos componentes e arraste-o para cima ou para baixo para alterar a ordem na qual os componentes serão executados.

d. Para remover o componente `update-linux-kernel-5`, escolha X no canto superior direito da caixa do componentes.

   Repita essa etapa para remover quaisquer outros componentes que você possa ter adicionado, deixando somente o `update-linux` componente selecionado.

8. Para prosseguir para a próxima etapa, escolha Avançar.

Etapa 3: Definir a configuração da infraestrutura - opcional

O Image Builder inicia EC2 instâncias em sua conta para personalizar imagens e executar testes de validação. As configurações da infraestrutura especificam os detalhes da infraestrutura para as instâncias que serão executadas na sua Conta da AWS durante o processo de criação.

Na seção Configuração da infraestrutura, as opções de configuração são padronizadas como `Create infrastructure configuration using service defaults`. Isso cria uma função do IAM e um perfil de instância associado para as instâncias de EC2 criação e teste usadas para configurar sua imagem. Para obter mais informações sobre as configurações de infraestrutura, consulte CreateInfrastructureConfigurationna Referência da API EC2 Image Builder.

Para este tutorial, estamos usando as configurações padrão.

> **Note**
>
> Para especificar uma sub-rede a ser usada em uma VPC privada, você pode criar sua própria configuração de infraestrutura personalizada ou usar configurações que você já criou.

- Para prosseguir para a próxima etapa, escolha Avançar.

Etapa 4: definir as configurações de distribuição - opcional

As configurações de distribuição incluem o nome da AMI de saída, configurações de região específicas para criptografia, permissões de lançamento e Contas da AWS organizações e unidades organizacionais (OUs) que podem iniciar a AMI de saída e configurações de licença.

Na seção Configurações de distribuição, as opções de configuração estão padronizadas com `Create distribution settings using service defaults`. Essa opção distribuirá a AMI de saída para a região atual. Para obter mais informações sobre as configurações de distribuição, consulte Gerenciar as configurações de distribuição do Image Builder.

Para este tutorial, estamos usando as configurações padrão.

- Para prosseguir para a próxima etapa, escolha Avançar.

Etapa 5: Revisão

A seção Revisar exibe todas as configurações que você definiu. Para editar informações em qualquer seção, escolha o botão Editar localizado no canto superior direito da seção da etapa. Por exemplo, se você quiser alterar o nome do pipeline, escolha o botão Editar no canto superior direito da seção Etapa 1: Detalhes do pipeline.

1. Depois de revisar suas configurações, escolha Criar pipeline para criar seu pipeline.
2. Você pode ver mensagens de sucesso ou falha na parte superior da página, à medida que seus recursos são criados para configurações de distribuição, configuração de infraestrutura, sua nova fórmula e o pipeline. Para ver detalhes de um recurso, incluindo o identificador do recurso, escolha Exibir detalhes.
3. Depois de visualizar os detalhes de um recurso, você pode ver detalhes de outros recursos escolhendo o tipo de recurso no painel de navegação. Por exemplo, para ver detalhes do seu novo pipeline, escolha Pipelines de imagem no painel de navegação. Se sua compilação for bem-sucedida, seu novo pipeline será exibido na lista Pipelines de imagem.

Etapa 6: limpar

Seu ambiente Image Builder, assim como sua casa, precisa de manutenção regular para ajudá-lo a encontrar o que precisa e concluir suas tarefas sem bagunça. Certifique-se de limpar regularmente os recursos temporários que você criou para testes. Caso contrário, você pode esquecer esses

recursos e, mais tarde, não se lembrar para que eles foram usados. E então, talvez você não possa se livrar deles com segurança.

> **Tip**
>
> Para evitar erros de dependência ao excluir recursos, certifique-se de excluir seus recursos na seguinte ordem:
>
> 1. Pipeline de imagens
> 2. Fórmula da imagem
> 3. Todos os recursos restantes

Siga estas etapas para limpar os recursos criados neste tutorial:

Exclua o pipeline

1. Para ver uma lista dos pipelines de compilação criados em sua conta, escolha Pipelines de imagem no painel de navegação.
2. Marque a caixa de seleção ao lado de Pipeline de nome para selecionar o pipeline que deseja excluir.
3. Na parte superior do painel Pipelines de imagem, no menu Ações, escolha Excluir.
4. Insira `Delete` para confirmar a exclusão e depois escolha Excluir.

Exclua a fórmula

1. Para ver uma lista das fórmulas criadas em sua conta, escolha Fórmulas de imagem no painel de navegação.
2. Marque a caixa de seleção ao lado de nome da fórmula para selecionar a fórmula que deseja excluir.
3. Na parte superior do painel Fórmulas de imagens, no menu Ações, escolha Excluir fórmula.
4. Insira `Delete` para confirmar a exclusão e depois escolha Excluir.

Excluir uma configuração de infraestrutura

1. Para ver uma lista das configurações de infraestrutura criadas em sua conta, escolha Configuração de infraestrutura no painel de navegação.
2. Marque a caixa de seleção ao lado de Nome da configuração para selecionar a configuração de infraestrutura que deseja excluir.
3. Na parte superior do painel Configurações de infraestrutura, escolha Excluir.
4. Insira `Delete` para confirmar a exclusão e depois escolha Excluir.

Excluir configurações de distribuição

1. Para ver uma lista das configurações de distribuição criadas em sua conta, escolha Configurações de distribuição no painel de navegação.
2. Marque a caixa de seleção ao lado de Nome da configuração para selecionar as configurações de distribuição que você criou para este tutorial.
3. Na parte superior do painel Configurações de distribuição, escolha Excluir.
4. Insira `Delete` para confirmar a exclusão e depois escolha Excluir.

Excluir a imagem

Siga estas etapas para verificar se você excluiu qualquer imagem criada no pipeline do tutorial. Não é provável que este tutorial crie uma imagem, a menos que tenha decorrido tempo suficiente desde a criação do pipeline para que ela seja executada, de acordo com o cronograma de compilação.

1. Para ver uma lista das imagens criadas em sua conta, escolha Imagens no painel de navegação.
2. Escolha a Versão da imagem que deseja remover. Isso abre a página Versões de compilação de imagens.
3. Marque a caixa de seleção ao lado da Versão da imagem que você deseja excluir. Você pode selecionar mais de uma versão de imagem de cada vez.
4. Na parte superior do painel Versões de compilação de imagens, escolha Excluir versão.
5. Insira `Delete` para confirmar a exclusão e depois escolha Excluir.

Tutorial: criar um pipeline de imagens com a imagem de contêiner do Docker de saída por meio do assistente do console do Image Builder

Este tutorial explica como criar um pipeline automatizado para criar e manter uma EC2 imagem personalizada do Docker do Image Builder usando o assistente de console Create image pipeline. Para ajudá-lo a percorrer as etapas com eficiência, as configurações padrão são usadas, quando disponíveis, e as seções opcionais são ignoradas.

Criar fluxo de trabalho de pipeline de imagens

- Etapa 1: especificar detalhes do pipeline
- Etapa 2: Escolher fórmula
- Etapa 3: Definir a configuração da infraestrutura - opcional
- Etapa 4: definir as configurações de distribuição - opcional
- Etapa 5: Revisão
- Etapa 6: limpar

Etapa 1: especificar detalhes do pipeline

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Para começar a criar seu pipeline, escolha Criar pipeline de imagem.
3. Na seção Geral, insira o nome do seu pipeline (obrigatório).
4. Na seção Compilar schedule, você pode manter os padrões para as opções de schedule. Observe que a zona horária mostrada para o schedule padrão é Horário Universal Coordenado (UTC). Para obter mais informações sobre o horário UTC e encontrar a diferença do seu fuso horário, consulte Time Zone Abbreviations, Worldwide List (Abreviações de fuso horário: lista mundial).

   Em Configurações de atualização de dependência, escolha a opção Executar pipeline no horário programado se houver a opção de atualizações de dependência. Essa configuração faz com que seu pipeline verifique se há atualizações antes de iniciar a compilação. Se não houver atualizações, ele ignora a compilação programada do pipeline.

> **Note**
>
> Para garantir que seu pipeline reconheça as atualizações e compilações de dependências conforme o esperado, você deve usar o controle de versionamento semântico (x.x.x) para sua imagem base e seus componentes. Para saber mais sobre o versionamento semântico dos recursos do Image Builder, consulte Versionamento semântico no Image Builder.

5. Para prosseguir para a próxima etapa, escolha Avançar.

Etapa 2: Escolher fórmula

1. O padrão do Image Builder é Usar a fórmula existente na seção Fórmula. Porém, na primeira vez, escolha a opção Criar nova fórmula.
2. Na seção Tipo de imagem, escolha a opção de imagem do Docker para criar um pipeline de contêineres que produzirá uma imagem do Docker e a distribuirá aos repositórios do Amazon ECR nas regiões de destino.
3. Na seção Geral, insira as seguintes caixas obrigatórias:
   - Nome: o nome da sua fórmula
   - Versão: a versão da sua fórmula (use o formato <major>.<minor>.<patch>, onde major, minor e patch são valores inteiros). As novas fórmulas geralmente começam com `1.0.0`.
4. Na seção Imagem de origem, mantenha os valores padrão para Selecionar imagem, Sistema operacional da imagem e Origem da imagem. Isso resulta em uma lista de imagens de contêiner do Amazon Linux 2, gerenciadas pela Amazon, para você escolher como sua imagem base.
   a. No menu suspenso Nome da imagem, escolha uma imagem.
   b. Mantenha o padrão para as opções de controle de versionamento automático (use a versão mais recente disponível do SO).

> **Note**
>
> Essa configuração garante que seu pipeline use versionamento semântico para a imagem base, para detectar atualizações de dependências para trabalhos programados automaticamente. Para saber mais sobre o versionamento semântico

dos recursos do Image Builder, consulte Versionamento semântico no Image Builder.

5. Na seção Componentes, você pode optar por não adicionar nenhum componente e continuar. Se você quiser adicionar componentes, no painel Criar componentes — Amazon Linux, você pode navegar pelos componentes listados na página. Use o controle de paginação no canto superior direito para navegar por outros componentes que estão disponíveis para seu sistema operacional de imagem base. Você também pode pesquisar componentes específicos ou criar seu próprio componente de compilação usando o Gerenciador de componentes.

   Para este tutorial, escolha um componente que atualize o Linux com as atualizações de segurança mais recentes, da seguinte forma:

   a. Filtre os resultados inserindo a palavra `update` na barra de pesquisa localizada na parte superior do painel.
   b. Marque a caixa de seleção para o componente de compilação `update-linux`.
   c. Role para baixo e, no canto superior direito da lista Componentes selecionados, escolha Expandir tudo.
   d. Mantenha o padrão para as opções de versionamento (use a versão mais recente do componente disponível).

   > **Note**
   >
   > Essa configuração garante que seu pipeline use versionamento semântico do componente selecionado, para detectar atualizações de dependências para trabalhos agendados automaticamente. Para saber mais sobre o versionamento semântico dos recursos do Image Builder, consulte Versionamento semântico no Image Builder.

   Se você selecionou um componente que tenha parâmetros de entrada, também verá os parâmetros nessa área. Os parâmetros não são abordados neste tutorial. Para obter mais informações sobre como usar parâmetros de entrada em seus componentes e configurá-los em suas fórmulas, consulte Tutorial: criar um componente personalizado com parâmetros de entrada.

   Reordenar componentes (opcional)

Se você escolheu mais de um componente para incluir em sua imagem, você pode usar a drag-and-drop ação para reorganizá-los na ordem em que devem ser executados durante o processo de criação.

> **Note**
>
> Os componentes de endurecimento do CIS não seguem as regras padrão de ordenação de componentes nas fórmulas do Image Builder. Os componentes de endurecimento do CIS sempre são executados por último para garantir que os testes de benchmark sejam executados em relação à sua imagem de saída.

1. Volte para cima até a lista de componentes disponíveis.
2. Marque a caixa de seleção do componente de compilação `update-linux-kernel-mainline` (ou qualquer outro componente de sua escolha).
3. Role para baixo até a lista Selected components para ver se há pelo menos dois resultados.
4. Os componentes recém-adicionados podem não ter seu versionamento expandido. Para expandir as opções de versionamento, você pode escolher a seta ao lado das opções de versionamento ou pode ativar e desativar a opção Expandir tudo para expandir o versionamento de todos os componentes selecionados.
5. Escolha um dos componentes e arraste-o para cima ou para baixo para alterar a ordem na qual os componentes serão executados.
6. Para remover o componente `update-linux-kernel-mainline`, escolha `X` no canto superior direito da caixa do componentes.
7. Repita a etapa anterior para remover quaisquer outros componentes que você possa ter adicionado, deixando somente o componente `update-linux` selecionado.

6. Na seção Modelo do Dockerfile, selecione a opção Usar exemplo. No painel Conteúdo, note as variáveis contextuais em que o Image Builder coloca informações de compilação ou scripts, com base na fórmula da imagem do contêiner.

   Por padrão, o Image Builder usa as variáveis contextuais a seguir em seu Dockerfile.

parentImage (obrigatória)

No momento da compilação, esta variável é resolvida na imagem de base da sua fórmula.

Exemplo:

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

environments (necessários se os componentes forem especificados)

Esta variável será resolvida em um script que executa componentes.

Exemplo:

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

components (opcional)

O Image Builder resolve scripts de componentes de compilação e teste para os componentes que a fórmula de contêiner inclui. Esta variável pode ser colocada em qualquer lugar no Dockerfile, depois da variável environments.

Exemplo:

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

7. Na seção Repositório de destino, especifique o nome do repositório Amazon ECR que você criou como pré-requisito para este tutorial. Esse repositório é usado como a configuração padrão para a configuração de distribuição na região em que o pipeline é executado (Região 1).

> **Note**
>
> O repositório de destino deve existir no Amazon ECR para todas as regiões de destino antes da distribuição.

8. Para prosseguir para a próxima etapa, escolha Avançar.

Etapa 3: Definir a configuração da infraestrutura - opcional

O Image Builder inicia EC2 instâncias em sua conta para personalizar imagens e executar testes de validação. As configurações da infraestrutura especificam os detalhes da infraestrutura para as instâncias que serão executadas na sua Conta da AWS durante o processo de criação.

Na seção Configuração da infraestrutura, as opções de configuração são padronizadas como `Create infrastructure configuration using service defaults`. Isso cria um perfil do IAM e um perfil de instância associado que são usados pelas instâncias de compilação para configurar sua imagens de contêiner. Além disso, você pode criar sua própria configuração de infraestrutura personalizada ou usar configurações que você já criou. Para obter mais informações sobre as configurações de infraestrutura, consulte CreateInfrastructureConfigurationna Referência da API EC2 Image Builder.

Para este tutorial, estamos usando as configurações padrão.

- Para prosseguir para a próxima etapa, escolha Avançar.

Etapa 4: definir as configurações de distribuição - opcional

As configurações de distribuição consistem nas regiões de destino e no nome do repositório Amazon ECR de destino. As imagens do Docker de saída são implantadas no repositório Amazon ECR nomeado em cada região.

Na seção Configurações de distribuição, as opções de configuração estão padronizadas com `Create distribution settings using service defaults`. Essa opção distribuirá a imagem do Docker de saída para o repositório Amazon ECR especificado em sua fórmula de contêiner para a região em que seu pipeline é executado (Região 1). Se você escolher `Create new distribution settings`, poderá substituir o repositório ECR da região atual e adicionar mais regiões para distribuição.

Para este tutorial, estamos usando as configurações padrão.

- Para prosseguir para a próxima etapa, escolha Avançar.

Etapa 5: Revisão

A seção Revisar exibe todas as configurações que você definiu. Para editar informações em qualquer seção, escolha o botão Editar localizado no canto superior direito da seção da etapa. Por

exemplo, se você quiser alterar o nome do pipeline, escolha o botão Editar no canto superior direito da seção Etapa 1: Detalhes do pipeline.

1. Depois de revisar suas configurações, escolha Criar pipeline para criar seu pipeline.
2. Você pode ver mensagens de sucesso ou falha na parte superior da página, à medida que seus recursos são criados para configurações de distribuição, configuração de infraestrutura, sua nova fórmula e o pipeline. Para ver detalhes de um recurso, incluindo o identificador do recurso, escolha Exibir detalhes.
3. Depois de visualizar os detalhes de um recurso, você pode ver detalhes de outros recursos escolhendo o tipo de recurso no painel de navegação. Por exemplo, para ver detalhes do seu novo pipeline, escolha Pipelines de imagem no painel de navegação. Se sua compilação for bem-sucedida, seu novo pipeline será exibido na lista Pipelines de imagem.

Etapa 6: limpar

Seu ambiente Image Builder, assim como sua casa, precisa de manutenção regular para ajudá-lo a encontrar o que precisa e concluir suas tarefas sem bagunça. Certifique-se de limpar regularmente os recursos temporários que você criou para testes. Caso contrário, você pode esquecer esses recursos e, mais tarde, não se lembrar para que eles foram usados. E então, talvez você não possa se livrar deles com segurança.

> **Tip**
>
> Para evitar erros de dependência ao excluir recursos, certifique-se de excluir seus recursos na seguinte ordem:
>
> 1. Pipeline de imagens
> 2. Fórmula da imagem
> 3. Todos os recursos restantes

Siga estas etapas para limpar os recursos criados neste tutorial:

Exclua o pipeline

1. Para ver uma lista dos pipelines de compilação criados em sua conta, escolha Pipelines de imagem no painel de navegação.

2. Marque a caixa de seleção ao lado de Pipeline de nome para selecionar o pipeline que deseja excluir.
3. Na parte superior do painel Pipelines de imagem, no menu Ações, escolha Excluir.
4. Insira `Delete` para confirmar a exclusão e depois escolha Excluir.

Excluir a fórmula do contêiner

1. Para ver uma lista das fórmulas de contêiner criadas em sua conta, escolha Fórmulas de contêiner no painel de navegação.
2. Marque a caixa de seleção ao lado de nome da fórmula para selecionar a fórmula que deseja excluir.
3. Na parte superior do painel Fórmulas de contêiner, no menu Ações, escolha Excluir fórmula.
4. Insira `Delete` para confirmar a exclusão e depois escolha Excluir.

Excluir uma configuração de infraestrutura

1. Para ver uma lista das configurações de infraestrutura criadas em sua conta, escolha Configuração de infraestrutura no painel de navegação.
2. Marque a caixa de seleção ao lado de Nome da configuração para selecionar a configuração de infraestrutura que deseja excluir.
3. Na parte superior do painel Configurações de infraestrutura, escolha Excluir.
4. Insira `Delete` para confirmar a exclusão e depois escolha Excluir.

Excluir configurações de distribuição

1. Para ver uma lista das configurações de distribuição criadas em sua conta, escolha Configurações de distribuição no painel de navegação.
2. Marque a caixa de seleção ao lado de Nome da configuração para selecionar as configurações de distribuição que você criou para este tutorial.
3. Na parte superior do painel Configurações de distribuição, escolha Excluir.
4. Insira `Delete` para confirmar a exclusão e depois escolha Excluir.

Excluir a imagem

Siga estas etapas para verificar se você excluiu qualquer imagem criada no pipeline do tutorial. Não é provável que este tutorial crie uma imagem, a menos que tenha decorrido tempo suficiente desde a criação do pipeline para que ela seja executada, de acordo com o cronograma de compilação.

1. Para ver uma lista das imagens criadas em sua conta, escolha Imagens no painel de navegação.
2. Escolha a Versão da imagem que deseja remover. Isso abre a página Versões de compilação de imagens.
3. Marque a caixa de seleção ao lado da Versão da imagem que você deseja excluir. Você pode selecionar mais de uma versão de imagem de cada vez.
4. Na parte superior do painel Versões de compilação de imagens, escolha Excluir versão.
5. Insira `Delete` para confirmar a exclusão e depois escolha Excluir.

Tutorial: criar um componente personalizado com parâmetros de entrada

Você pode gerenciar componentes do Image Builder, incluindo criar e definir parâmetros de componentes, diretamente do console do EC2 Image Builder, do AWS CLI, ou da API Image Builder ou SDKs. Nesta seção, abordaremos a criação e o uso de parâmetros em seu componente e a configuração dos parâmetros do componente por meio do console e dos AWS CLI comandos do Image Builder em tempo de execução.

> **Important**
>
> Os parâmetros do componente são valores de texto simples e estão logados em AWS CloudTrail. Recomendamos que você use AWS Secrets Manager ou o AWS Systems Manager Parameter Store para armazenar seus segredos. Para obter mais informações sobre o Secrets Manager, consulte O que é o Secrets Manager? no AWS Secrets Manager Guia do usuário. Para obter mais informações sobre o AWS Systems Manager Parameter Store, consulte AWS Systems Manager Parameter Store no AWS Systems Manager Guia do usuário.

Use parâmetros no documento do componente do YAML

Para compilar um componente, você deve fornecer um documento de componente de aplicação YAML ou JSON. O documento contém o código executado durante as fases e etapas que você define para tornar sua imagem personalizada. A fórmula que faz referência ao componente pode definir os parâmetros para personalizar os valores no runtime com valores padrão que entrarão em vigor se o parâmetro não for definido com um valor específico.

Criar um documento de componente com parâmetros de entrada

Esta seção mostra como definir e usar os parâmetros de entrada no documento do componente do YAML.

Para criar um documento do componente do aplicativo YAML que usa parâmetros e executa comandos em suas instâncias de compilação ou teste do Image Builder, siga as etapas que correspondem ao sistema operacional da imagem:

Linux

Crie um documento do componente do YAML

Use uma ferramenta de edição de arquivos para criar um arquivo do documento do componente. Os exemplos de documentação usam um arquivo chamado *hello-world-test.yaml*, que tem o seguinte conteúdo:

```
# 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**
>
> Use uma ferramenta como esse validador YAML on-line ou uma extensão lint YAML em seu ambiente de código para verificar se seu YAML está bem formado.

Windows

Crie um documento do componente do YAML

Use uma ferramenta de edição de arquivos para criar um arquivo do documento do componente. Os exemplos de documentação usam um arquivo chamado *`hello-world-test.yaml`*, que tem o seguinte conteúdo:

```
# 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**
>
> Use uma ferramenta como esse validador YAML on-line ou uma extensão lint YAML em seu ambiente de código para verificar se seu YAML está bem formado.

macOS

Crie um documento do componente do YAML

Use uma ferramenta de edição de arquivos para criar um arquivo do documento do componente. Os exemplos de documentação usam um arquivo chamado *hello-world-test.yaml*, que tem o seguinte conteúdo:

```
# 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**
>
> Use uma ferramenta como esse validador YAML on-line ou uma extensão lint YAML em seu ambiente de código para verificar se seu YAML está bem formado.

Para obter mais informações sobre as fases, as etapas e a sintaxe dos documentos de componente da aplicação AWSTOE , consulte Use documents in AWSTOE. Para obter mais informações sobre parâmetros e seus requisitos, consulte a seção Parâmetros da página Definir e referenciar variáveis no AWSTOE.

Criar um componente a partir do documento do componente do YAML

Seja qual for o método usado para criar um AWSTOE componente, o documento do componente do aplicativo YAML é sempre necessário como linha de base.

- Para criar um componente diretamente do documento YAML com o console do Image Builder, consulte Criar um componente personalizado com o console.
- Para criar um componente por meio do comando create-component do Image Builder na linha de comandos, consulte Crie um componente personalizado a partir do AWS CLI. Substitua o nome do documento do YAML nesses exemplos pelo nome do seu documento YAML Hello World (*`hello-world-test.yaml`*).

Definir parâmetros do componente em uma fórmula do Image Builder com o console

A configuração dos parâmetros do componente funciona da mesma forma para fórmulas de imagens e fórmulas de contêineres. Ao criar uma nova fórmula ou uma nova versão de uma fórmula, você escolhe quais componentes incluir nas listas Componentes de compilação e Componentes de teste. As listas de componentes incluem componentes que são aplicáveis ao sistema operacional básico que você escolheu para sua imagem.

Depois de selecionar um componente, ele é exibido na seção Componentes selecionados, diretamente abaixo das listas de componentes. São mostradas opções de configuração para cada componente selecionado. Se seu componente tiver parâmetros de entrada definidos, eles serão exibidos como uma seção expansível chamada Parâmetros de entrada.

As seguintes configurações de parâmetros são mostradas para cada parâmetro definido para seu componente:

- Nome do parâmetro (não editável): o nome do parâmetro.
- Descrição (não editável): a descrição do parâmetro
- Tipo (não editável): o tipo de dados para o valor do parâmetro.
- Valor: o valor do parâmetro. Se você estiver usando esse componente pela primeira vez nessa fórmula e um valor padrão tiver sido definido para o parâmetro de entrada, o valor padrão aparecerá na caixa Valor com texto acinzentado. Se nenhum outro valor for inserido, o Image Builder usará o valor padrão.

Use um parâmetro de imagem base em sua receita

Quando você cria uma receita para personalizações de imagens, há várias maneiras de identificar a imagem base com a qual você começa. Se você especificar o ID da Amazon Machine Image (AMI) para sua imagem base e essa imagem base for atualizada, seu ID de AMI poderá mudar e você precisará atualizar sua receita para corresponder.

Em vez de alterar sua receita toda vez que o ID da imagem base for alterado, você pode definir um AWS Systems Manager parâmetro de armazenamento de parâmetros (parâmetro SSM) para armazenar o valor do ID de AMI da imagem base e, em seguida, usar o parâmetro para especificar a imagem base em sua receita. Para AWS gerenciado AMIs, você pode usar um parâmetro público para a versão mais recente.

Este tutorial mostra o processo de criação de um parâmetro de ID de AMI e seu uso em uma receita de imagem. As etapas do Image Builder neste tutorial são baseadas em console.

Conteúdo

- Etapa 1: Localizar ou criar um parâmetro do Parameter Store
- Etapa 2: configurar as permissões do IAM (opcional)
- Etapa 3: criar uma receita de imagem que use o parâmetro

Etapa 1: Localizar ou criar um parâmetro do Parameter Store

O processo dessa etapa depende do tipo de AMI que você especifica para sua imagem base. Para AWS gerenciado AMIs, você pode usar um parâmetro público que se refere à versão atual. Alguns parâmetros podem não estar disponíveis em todos Regiões da AWS.

Para começar, abra a guia que corresponde à sua AMI.

AWS managed AMI

Se sua imagem base for uma AMI AWS gerenciada, você poderá usar parâmetros públicos para especificar a ID da AMI, em vez de criar seu próprio parâmetro. Para encontrar o parâmetro público para sua AMI, consulte Descobrindo parâmetros públicos no Guia do AWS Systems Manager usuário.

Custom AMI

Para criar um parâmetro de ID de AMI, siga as instruções para criar parâmetros do Parameter Store no Systems Manager com o console AWS CLI, ou PowerShell. Forneça os valores a seguir para garantir que o valor do parâmetro seja um ID de AMI.

Nível de parâmetros: `Standard`

Digite: `String`

Tipo de dados: Selecione`aws:ec2:image`. Quando você especifica esse tipo, o sistema valida o valor inserido para garantir que seja uma ID de AMI.

Valor: insira uma ID de AMI válida (por exemplo,***`ami-1234567890abcdef1`***).

Etapa 2: configurar as permissões do IAM (opcional)

Para usar um parâmetro do Systems Manager Parameter Store (parâmetro SSM), seja público ou privado, as seguintes ações do Systems Manager Parameter Store devem ser especificadas em uma função do IAM, com o parâmetro listado como um recurso. A função vinculada ao serviço Image Builder concede permissão para obter parâmetros públicos ou obter ou atualizar parâmetros privados que tenham um `/imagebuilder/` prefixo. Para parâmetros privados que não têm esse prefixo, você pode adicionar permissão à sua função de execução.

- `ssm:GetParameter`— Essa ação permite que você use um parâmetro SSM para especificar a imagem base em sua receita.

- `ssm:PutParameter`— Essa ação permite que você armazene o ID da AMI de saída em um parâmetro SSM durante a distribuição. A definição da política parece a mesma, mas este tutorial não inclui a ação put no exemplo de política.

1. Crie um papel personalizado (opcional)

   Ao criar um pipeline ou usar o comando create-image no AWS CLI, você só pode especificar uma função de execução do Image Builder. Se você tiver definido uma função de execução do fluxo de trabalho do Image Builder, você adicionaria quaisquer permissões de recursos adicionais a essa função. Caso contrário, você criaria um novo papel personalizado que inclua as permissões necessárias. Se você já tem uma função de execução personalizada definida, pode pular essa etapa.

   Siga o processo de criação de uma função para delegar permissões a um AWS serviço no Guia do AWS Identity and Access Management usuário.

2. Adicione permissões à sua função personalizada

   Para adicionar as permissões do parâmetro SSM à sua função personalizada, siga a seção Atualizar a política de permissões para um processo de função no Guia do AWS Identity and Access Management usuário.

   O exemplo de política a seguir mostra a `ssm:GetParameter` ação com um parâmetro criado na sua conta.

   JSON

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

Para obter mais informações sobre recursos de parâmetros públicos, consulte Chamando parâmetros públicos da AMI no Guia AWS Systems Manager do usuário.

Etapa 3: criar uma receita de imagem que use o parâmetro

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Escolha Receitas de imagem e, em seguida, escolha Criar receita de imagem na página da lista.
3. Preencha a seção Imagem base, da seguinte forma:
   a. Escolha a opção Usar AMI personalizada. Isso exibe campos adicionais nos quais você pode inserir a ID da AMI ou um parâmetro SSM que contém a ID da AMI.
   b. Escolha a opção de parâmetro SSM.
   c. No campo Parâmetro SSM, insira o nome do parâmetro ou o Amazon Resource Name (ARN) do parâmetro que você criou na Etapa 1. Se você digitar o nome, ele não terá o prefixo no console.
4. Conclua a configuração restante da receita conforme necessário.

> **Note**
>
> Se você definir a imagem principal por meio de outras interfaces, como a AWS CLI, o nome do parâmetro deverá ter um prefixo de `ssm:` (por exemplo,`ssm:/ImageBuilder-Tutorial/BaseAMI`.

Usar componentes para personalizar sua imagem do Image Builder

O Image Builder usa o aplicativo de gerenciamento de componentes AWS Task Orchestrator and Executor(AWSTOE) para orquestrar fluxos de trabalho complexos. Os componentes de criação e teste que funcionam com o AWSTOE aplicativo são baseados em documentos YAML que definem os scripts para personalizar ou testar sua imagem. Para imagens da AMI, o Image Builder instala componentes e o aplicativo de gerenciamento de AWSTOE componentes em suas instâncias de EC2 compilação e teste da Amazon. Para imagens de contêiner, os AWSTOE componentes e o aplicativo de gerenciamento de componentes são instalados dentro do contêiner em execução.

O Image Builder usa AWSTOE para realizar todas as atividades na instância. Não é necessária nenhuma configuração adicional para interagir AWSTOE quando você executa os comandos do Image Builder ou usa o console do Image Builder.

> **Note**
>
> Quando um componente gerenciado pela Amazon chega ao fim de sua vida útil de suporte, ele não é mais mantido. Cerca de quatro semanas antes que isso ocorra, todas as contas que estão usando o componente recebem uma notificação e uma lista das fórmulas afetadas em sua conta AWS Health Dashboard. Para saber mais AWS Health, consulte o Guia AWS Health do usuário.

Etapas do fluxo de trabalho para criar uma nova imagem

O fluxo de trabalho do Image Builder para criar novas imagens inclui os dois estágios distintos a seguir.

1. Estágio de construção (pré-instantâneo) — Durante o estágio de construção, você faz alterações na instância de EC2 compilação da Amazon que está executando sua imagem base, para criar a linha de base para sua nova imagem. Por exemplo, sua fórmula pode incluir componentes que instalam uma aplicação ou modificam as configurações do firewall do sistema operacional.

   As seguintes fases dos documentos de componente são executadas durante o estágio de compilação:

   - build

- validar

Depois que esse estágio for concluído com êxito, o Image Builder cria um snapshot ou imagem de contêiner que ele usa para o estágio de teste e além.

2. Estágio de teste (pós-captura instantânea) — Durante o estágio de teste, há algumas diferenças entre imagens que criam AMIs e imagens de contêineres. Para fluxos de trabalho da AMI, o Image Builder executa uma EC2 instância a partir do snapshot que ele criou como etapa final do estágio de criação. Os testes são executados na nova instância para validar as configurações e garantir que a instância esteja funcionando conforme o esperado. Para fluxos de trabalho de contêineres, os testes são executados na mesma instância usada para a compilação.

   A seguinte fase dos documentos de componente é executada para cada componente incluído na fórmula durante o estágio de teste de compilação da imagem:

   - teste

   Essa fase de componente se aplica aos tipos de componentes de compilação e teste. Depois que esse estágio for concluído com êxito, o Image Builder poderá criar e distribuir sua imagem final a partir do snapshot ou da imagem do contêiner.

> **Note**
>
> Embora a estrutura do AWSTOE aplicativo permita definir várias fases em um documento componente, o Image Builder tem regras rígidas sobre quais fases ele é executado e durante quais estágios ele as executa. Para que um componente seja executado durante o estágio de compilação da imagem, o documento do componente deve definir pelo menos uma dessas fases: `build` ou `validate`. Para que um componente seja executado durante o estágio de teste da imagem, o documento do componente deve definir a fase `test`, e nenhuma outra fase.
>
> Como o Image Builder executa os estágios de forma independente, o encadeamento de referências em documentos componentes não pode ultrapassar os limites do estágio. Você não pode encadear um valor de uma fase que é executada no estágio de construção para uma fase que é executada no estágio de teste. No entanto, você pode definir parâmetros de entrada para o destino pretendido e transmitir valores por meio da linha de comando. Para obter mais informações sobre como definir parâmetros de componentes em suas fórmulas do Image Builder, consulteTutorial: criar um componente personalizado com parâmetros de entrada.

Para ajudar na solução de problemas em sua instância de compilação ou teste,AWSTOE crie uma pasta de log que contém o documento de entrada e os arquivos de log para monitorar o que está acontecendo sempre que um componente é executado. Se você configurou um bucket do Amazon S3 na configuração do pipeline, os registros também são gravados lá. Para obter mais informações sobre documentos YAML e saída de log, consulteUse a estrutura de documentos de AWSTOE componentes para componentes personalizados.

> **Tip**
>
> Quando você tem muitos componentes para monitorar, a marcação ajuda a identificar um componente ou versão específica com base nas tags que você atribuiu a ele. Para obter mais informações sobre como marcar seus recursos usando os comandos do Image Builder no AWS CLI, consulte a Marcar recursos seção deste guia.

Esta seção aborda como listar, visualizar, criar e importar componentes usando o console do Image Builder ou os comandos da AWS CLI.

Tópicos

- Listar e visualizar detalhes do componente
- Use AWS Marketplace componentes para personalizar sua imagem
- Usar componentes gerenciados para personalizar sua imagem do Image Builder
- Desenvolver componentes personalizados para imagens do Image Builder
- Como o Image Builder usa o AWS Task Orchestrator and Executor aplicativo para gerenciar componentes

Listar e visualizar detalhes do componente

Esta seção descreve como você pode encontrar informações e visualizar detalhes dos componentes que você usa em suas receitas do EC2 Image Builder.

Detalhes do componente

- Listar componentes do Image Builder
- Listar as versões de construção de componentes do AWS CLI
- Obtenha detalhes do componente no AWS CLI
- Obtenha detalhes da política de componentes no AWS CLI

Listar componentes do Image Builder

É possível usar um dos métodos a seguir para listar e filtrar componentes do Image Builder.

Console de gerenciamento da AWS

Para exibir uma lista de componentes no Console de gerenciamento da AWS, siga estas etapas:

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Selecione Componentes no painel de navegação. Por padrão, o Image Builder mostra uma lista dos componentes que sua conta possui.
3. Opcionalmente, você pode filtrar pela propriedade do componente. Para ver os componentes que você não possui, mas aos quais tem acesso, amplie a lista suspensa do tipo de proprietário e selecione um dos valores. A lista de tipos de proprietário está localizada na barra de pesquisa, ao lado da caixa de texto de pesquisa. Você pode selecionar um dos seguintes valores:

   - AWS Marketplace— Componentes associados diretamente a uma assinatura de AWS Marketplace produto.
   - Início rápido (gerenciado pela Amazon): componentes publicamente disponíveis que a Amazon cria e mantém.
   - De minha propriedade: componentes que você criou. Esta é a seleção padrão.
   - Compartilhado comigo: componentes que outras pessoas criaram e compartilharam com você a partir de suas contas.
   - Gerenciado por terceiros — Componentes que um terceiro possui e nos AWS Marketplace quais você se inscreveu.

AWS CLI

O exemplo a seguir mostra como usar o comando list-components para retornar uma lista de componentes do Image Builder que sua conta possui.

```
aws imagebuilder list-components
```

Opcionalmente, você pode filtrar pela propriedade do componente. O atributo proprietário define quem é o proprietário dos componentes que deseja listar. Por padrão, essa solicitação retorna uma lista dos componentes que sua conta possui. Para filtrar os resultados por proprietário do

componente, especifique um dos valores a seguir com o parâmetro `--owner` ao executar o comando list-components.

Valores do proprietário do componente

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

Os exemplos a seguir mostram o comando list-components com o parâmetro `--owner` para filtrar os resultados.

```
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
```

Listar as versões de construção de componentes do AWS CLI

O exemplo a seguir mostra como usar o comando list-component-build-versions para listar as versões de compilação do componente que têm uma versão semântica específica. Para saber mais sobre o versionamento semântico dos recursos do Image Builder, consulte Versionamento semântico no 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"
            }
        }
    ]
}
```

Obtenha detalhes do componente no AWS CLI

O exemplo a seguir mostra como usar o comando get-component para obter detalhes do componente quando você especifica o nome do recurso da Amazon (ARN) do componente.

```
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": {}
    }
}
```

Obtenha detalhes da política de componentes no AWS CLI

O exemplo a seguir mostra como usar o comando get-component-policy para obter detalhes de uma política de componente ao especificar o ARN do componente.

```
aws imagebuilder get-component-policy --component-arn arn:aws:imagebuilder:us-
west-2:123456789012:component/example-component/1.0.1
```

Use AWS Marketplace componentes para personalizar sua imagem

Além de uma grande seleção de imagens criadas por Independent Software Vendors (ISVs), ele AWS Marketplace oferece componentes que você pode usar para personalizar suas próprias

imagens do Image Builder. Você deve assinar esses AWS Marketplace componentes antes de poder usá-los em sua receita de imagem para criar uma nova imagem.

Quando você especifica um AWS Marketplace componente em uma receita de imagem, o Image Builder valida a assinatura e executa verificações de dependência para garantir que você tenha os recursos necessários para usá-lo. Quando a validação é bem-sucedida, o Image Builder cria downloads seguros para o componente e seus artefatos para uso nas compilações do pipeline de imagem.

Descubra os AWS Marketplace componentes

Você pode descobrir componentes AWS Marketplace de software para usar em suas receitas na página Discover products no console do Image Builder, da seguinte forma.

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. No painel de navegação, escolha Descobrir produtos na AWS Marketplaceseção.
3. Selecione a guia Components (Componentes). Essa guia lista todos os AWS Marketplace produtos que usam a opção de entrega que inclui componentes associados em AWS Marketplace.
4. Para pesquisar produtos de software específicos que incluam componentes, você pode inserir parte do nome na barra de pesquisa ou filtrar por `StatusOperating System`,`Publisher`, ou`Categories`. A barra de pesquisa também contém controles de paginação para seus resultados.

Resultados

Cada AWS Marketplace produto tem seu próprio painel de detalhes que inclui as seguintes informações.

O nome e o logotipo do AWS Marketplace produto

O nome do produto de software está vinculado aos detalhes do produto em AWS Marketplace. Você pode selecionar o link para saber mais sobre o produto em AWS Marketplace. Como alternativa, você pode ver um resumo das opções de assinatura e se inscrever diretamente nos resultados da pesquisa com o botão Exibir opções de assinatura, caso já tenha feito sua pesquisa.

Versão

Ele contém a versão principal do componente.

Sistema operacional

O sistema operacional no qual o componente foi projetado para ser executado.

Editora

O editor do componente. Isso está vinculado à página de detalhes do editor em AWS Marketplace. A página de detalhes do editor é aberta em uma nova guia do seu navegador.

Categorias

Uma ou mais categorias de AWS Marketplace produtos que se aplicam ao componente.

Status

Mostra se você está inscrito neste produto. Se você não estiver inscrito, poderá escolher Exibir opções de assinatura para ver um resumo das opções de assinatura do AWS Marketplace produto e, opcionalmente, assinar diretamente do console do Image Builder.

Componentes associados

Se o AWS Marketplace produto tiver uma ou mais versões incluídas em sua assinatura, elas serão mostradas na seção Componentes associados. A seção é reduzida inicialmente e exibe uma contagem dos componentes associados. Você pode expandir a seção para ver mais detalhes.

> **Note**
>
> O componente Center for Internet Security (CIS) associado ao produto de AWS Marketplace imagem não é exibido nos resultados do Discover products. Se você assinar seu produto de imagem, o componente associado será mostrado na página Assinaturas e como um componente de terceiros na caixa de diálogo Criar receita de imagem. Para obter mais informações sobre o componente, consulteComponentes de fortalecimento do CIS.

Inscrever-se em AWS Marketplace componentes

Depois de encontrar um AWS Marketplace produto com componentes que você deseja usar em suas receitas, você pode assiná-lo diretamente do console do Image Builder, da seguinte forma, ou você pode se inscrever no AWS Marketplace console.

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. No painel de navegação, escolha Descobrir produtos na AWS Marketplaceseção.
3. Selecione a guia Components (Componentes). Essa guia lista todos os AWS Marketplace produtos que usam a opção de entrega que inclui componentes associados em AWS Marketplace.
4. Para pesquisar um AWS Marketplace produto específico, insira parte do nome na barra de pesquisa. Se você conhece o editor, mas não sabe o nome exato do produto ou como soletrá-lo, você também pode filtrar por `Publisher` para obter uma lista dos produtos que o editor tem disponíveis.
5. Selecione o produto que você deseja assinar na lista de resultados e escolha Exibir opções de assinatura. Isso mostra um resumo das opções de assinatura do AWS Marketplace produto.
6. Selecione Inscrever-se para assinar o produto sem sair do console do Image Builder. Você é notificado de que a assinatura está sendo processada. Depois de se inscrever, o status é atualizado para`Subscribed`.

Para obter mais informações sobre os AWS Marketplace produtos nos quais você está inscrito atualmente, consulte as etapas do console descritas emAWS Marketplace assinaturas no Image Builder.

Use um AWS Marketplace componente em uma receita de imagem do Image Builder

Você pode usar AWS Marketplace componentes em suas receitas de imagem do Image Builder da mesma forma que usa outros tipos de componentes. Para a maioria dos componentes associados a um produto de AWS Marketplace imagem, a categoria de propriedade é`AWS Marketplace`. Por exemplo, para usar um componente de compilação de um AWS Marketplace produto que você assinou, escolha Adicionar componentes de compilação e `AWS Marketplace` selecione na lista. Isso abre um painel de seleção no lado direito da interface do console que lista AWS Marketplace os componentes.

> **Note**
>
> Se você estiver procurando pelo componente de fortalecimento do CIS, selecione`Third party managed`, na lista de proprietários em vez de. `AWS Marketplace`

Para obter mais informações sobre como selecionar, organizar e configurar parâmetros para seus componentes, consulteCriar uma nova versão de uma fórmula de imagem.

Usar componentes gerenciados para personalizar sua imagem do Image Builder

Os componentes gerenciados são criados AWS, às vezes em parceria com uma organização terceirizada, como o Center for Internet Security (CIS), por exemplo. Quando você usa componentes gerenciados em suas receitas de imagem ou contêiner, a Amazon fornece as versões mais recentes dos componentes que têm patches e outras atualizações aplicadas. Para obter uma lista de componentes ou obter informações sobre componentes, consulteListar e visualizar detalhes do componente.

A lista a seguir de componentes AWS gerenciados em destaque inclui um componente que está disponível para você usar ao assinar o CIS reforçado por AMIs meio do.AWS Marketplace

Componentes em destaque

- Instalação de aplicativo de componente gerenciado pelo pacote de distribuidor para imagens do Image Builder para Windows
- Componentes de fortalecimento do CIS
- Componentes de hardening do STIG gerenciados pela Amazon para o Image Builder

Instalação de aplicativo de componente gerenciado pelo pacote de distribuidor para imagens do Image Builder para Windows

AWS Systems Manager O Distributor ajuda você a empacotar e publicar software em nós AWS Systems Manager gerenciados. Você pode empacotar e publicar seu próprio software ou usar o Distributor para encontrar e publicar pacotes de software de agente fornecidos por AWS. Para obter mais informações sobre o Systems Manager Distributor, consulteAWS Systems Manager Distributor no Manual do usuário do AWS Systems Manager.

Componentes gerenciados para Distributor

Os seguintes componentes gerenciados do Image Builder usam o AWS Systems Manager Distributor para instalar pacotes de aplicativos em instâncias do Windows.

- O componente gerenciado `distributor-package-windows` usa o AWS Systems Manager Distributor para instalar pacotes de aplicações que você especifica em sua instância Windows de compilação de imagem. Para configurar parâmetros ao incluir esse componente em sua fórmula, consulte Configurar `distributor-package-windows` como um componente independente.
- O `aws-vss-components-windows` componente usa o AWS Systems Manager Distributor para instalar o `AwsVssComponents` pacote na sua instância de criação de imagem do Windows. Para configurar parâmetros ao incluir esse componente em sua fórmula, consulte Configurar `aws-vss-components-windows` como um componente independente.

Para obter mais informações sobre como usar componentes gerenciados em sua fórmula do Image Builder, consulte Criar uma nova versão de uma fórmula de imagem para fórmulas de imagens ou Criar uma nova versão de fórmula de contêiner para fórmulas de contêiner. Para obter mais informações sobre o *`AwsVssComponents`* pacote, consulte Criar um snapshot consistente com o aplicativo VSS no Guia do usuário da Amazon. EC2

Pré-requisitos

Antes de usar os componentes do Image Builder que dependem do Systems Manager Distributor para instalar pacotes de aplicativos, você deve garantir que os seguintes pré-requisitos sejam atendidos.

- Os componentes do Image Builder que usam o Systems Manager Distributor para instalar pacotes de aplicativos na sua instância precisam de permissão para chamar a API do Systems Manager. Antes de usar os componentes em uma fórmula do Image Builder, crie a política do IAM e a função que concedam permissão. Para configurar permissões, consulte Configurar permissões de Systems Manager Distributor.

> **Note**
>
> Atualmente, o Image Builder não é compatível com pacotes do Systems Manager Distributor que reinicializam a instância. Por exemplo, os pacotes de distribuidor `AWSNVMe`,

`AWSPVDrivers` e `AwsEnaNetworkDriver` reinicializam a instância e, portanto, não são permitidos.

Configurar permissões de Systems Manager Distributor

O componente `distributor-package-windows` e outros componentes que o usam, como o `aws-vss-components-windows`, exigem permissão adicional na instância de compilação para serem executados. A instância de compilação deve ser capaz de chamar a API do Systems Manager para iniciar uma instalação do Distributor e pesquisar o resultado.

Siga esses procedimentos em Console de gerenciamento da AWS para criar uma política e uma função personalizadas do IAM que concedam permissão para que os componentes do Image Builder instalem pacotes do Systems Manager Distributor a partir da instância de compilação.

Etapa 1: Criar uma política

Crie uma política do IAM para permissões do Distributor.

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Políticas e, em seguida, Criar política.
3. Na página Criar política, escolha a aba JSON e, em seguida, substitua o conteúdo padrão pela seguinte política JSON, substituindo partição, região e ID da conta conforme necessário ou usando curingas.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDistributorSendCommand",
            "Effect": "Allow",
            "Action": "ssm:SendCommand",
            "Resource": [
                "arn:aws:ssm:*::document/AWS-ConfigureAWSPackage",
                "arn:aws:ec2:*:111122223333:instance/*"
            ]
        },
        {
```

```
            "Sid": "AllowGetCommandInvocation",
            "Effect": "Allow",
            "Action": "ssm:GetCommandInvocation",
            "Resource": "*"
        }
    ]
}
```

4. Escolha Revisar política.
5. Em Name (Nome), insira um nome para identificar a política, como *InvokeDistributor* ou outro nome que preferir.
6. (Opcional) Para Description (Descrição), informe a descrição do propósito da função.
7. Selecione Criar política.

Etapa 2: Criar uma função

Crie um perfil do IAM para permissões do Distributor.

1. A partir do painel de navegação do console do IAM, escolha Funções e Criar função.
2. Em Select type of trusted entity (Selecionar o tipo de entidade confiável), escolha AWS service (Serviço da AWS).
3. Imediatamente em Escolher o serviço que usará essa função, escolha e EC2, em seguida, escolha Avançar: Permissões.
4. Em Selecione seu caso de uso, escolha e EC2, em seguida, escolha Avançar: Permissões.
5. Na lista de políticas, marque a caixa de seleção ao lado da Amazon SSMManaged InstanceCore. (Digite `SSM` na caixa de texto se você precisar restringir a lista.)
6. Nessa lista de políticas, escolha a caixa ao lado de EC2InstanceProfileForImageBuilder. (Digite `ImageBuilder` na caixa de texto se você precisar restringir a lista.)
7. Escolha Next: Tags (Próximo: tags).
8. (Opcional) Adicione um ou mais pares de chave-valor de tag para organizar, rastrear ou controlar o acesso para esta função e selecione Próximo: revisar.
9. Em Role name (Nome da função), insira um nome para a nova função, como *InvokeDistributor* ou outro nome que você preferir.
10. (Opcional) Para Role description (Descrição da função), substitua o texto padrão pela descrição do propósito dessa função.
11. Selecione Create role. O sistema faz com que você retorne para a página Roles.

Etapa 3: Anexar a política à função

A etapa final para configurar suas permissões de Distribuidor é anexar a política do IAM ao perfil do IAM.

1. A partir da página Perfis no console do IAM, escolha o nome do perfil que você acabou de criar. A função Summary page (Página de resumo) é aberta.
2. Escolha Attach policies (Anexar políticas).
3. Pesquise a política que você criou no procedimento anterior e selecione a caixa de seleção ao lado do nome.
4. Escolha Anexar política.

Use essa função no recurso de configuração de infraestrutura do Image Builder para qualquer imagem que inclua componentes que usam o Systems Manager Distributor. Para obter mais informações, consulte Criar uma configuração de infraestrutura.

Configurar **`distributor-package-windows`** como um componente independente

Para usar o componente `distributor-package-windows` em uma fórmula, defina os seguintes parâmetros que configuram o pacote a ser instalado.

> **Note**
>
> Antes de usar o componente `distributor-package-windows` em uma fórmula, você deve garantir que todos os Pré-requisitos sejam atendidos.

- Ação (Obrigatório): especifique se deseja instalar ou desinstalar o pacote. Os valores válidos são `Install` e `Uninstall`. O valor padrão é `Install`.
- PackageName(Obrigatório) — O nome do pacote do Distribuidor a ser instalado ou desinstalado. Para obter uma lista de nomes de pacotes válidos, consulte Encontre pacotes de distribuidor.
- PackageVersion(Opcional) — A versão do pacote do Distribuidor a ser instalada. PackageVersion usa como padrão a versão recomendada.
- AdditionalArguments(Opcional) — Uma string JSON que contém os parâmetros adicionais a serem fornecidos ao script para instalar, desinstalar ou atualizar um pacote. Para obter mais informações, consulte AdditionalArguments na seção Entradas de aws:ConfigurePackage da página de referência do plug-in de documentos do Systems Manager Command.

Configurar **aws-vss-components-windows** como um componente independente

Ao usar o componente `aws-vss-components-windows` em uma fórmula, você pode, opcionalmente, definir o parâmetro `PackageVersion` para usar uma versão específica do pacote `AwsVssComponents`. Quando você omite esse parâmetro, o componente usa como padrão a versão recomendada do pacote `AwsVssComponents`.

> **Note**
>
> Antes de usar o componente `aws-vss-components-windows` em uma fórmula, você deve garantir que todos os Pré-requisitos sejam atendidos.

Encontre pacotes de distribuidor

A Amazon e terceiros fornecem pacotes públicos que você pode instalar com o Systems Manager Distributor.

Para ver os pacotes disponíveis no Console de gerenciamento da AWS, faça login no AWS Systems Manager console e escolha Distribuidor no painel de navegação. A página do Distribuidor mostra todos os pacotes disponíveis para você. Para obter mais informações sobre como listar pacotes disponíveis com o AWS CLI, consulte Exibir pacotes (linha de comando) no Guia AWS Systems Manager do usuário.

Você também pode criar seus próprios pacotes privados do Systems Manager Distributor. Para obter mais informações, consulte Criar um pacote no Guia do usuário do AWS Systems Manager.

Componentes de fortalecimento do CIS

O CIS: Centro de segurança na internet é uma organização sem fins lucrativos dirigida pela comunidade. Seus especialistas em segurança cibernética trabalham juntos para desenvolver diretrizes de segurança de TI que protejam organizações públicas e privadas contra ameaças cibernéticas. Seu conjunto globalmente reconhecido de melhores práticas, conhecido como CIS Benchmarks, ajuda organizações de TI em todo o mundo a configurar seus sistemas com segurança. Para artigos populares, postagens em blogs, podcasts, webinars e whitepapers, consulte CIS Insights no site Center for Internet Security.

Referências do CIS

O CIS cria e mantém um conjunto de diretrizes de configuração, conhecido como CIS Benchmarks, que fornecem as melhores práticas de configuração para tecnologias específicas, incluindo sistemas operacionais, plataformas de nuvem, aplicativos, bancos de dados e muito mais. CIS Benchmarks são reconhecidos como um padrão da indústria por organizações e normas como PCI DSS, HIPAA, computação em nuvem SRG do DoD, FISMA, DFARS e FEDRAMP. Para saber mais, consulte CIS Benchmarks no site do Center for Internet Security.

Componentes de fortalecimento do CIS

Ao assinar uma imagem reforçada do CIS em AWS Marketplace, você também obtém acesso ao componente de proteção associado que executa um script para aplicar as diretrizes do CIS Benchmarks de nível 1 para sua configuração. A organização CIS possui e mantém os componentes de fortalecimento do CIS para garantir que eles reflitam as diretrizes mais recentes.

> **Note**
>
> Os componentes de endurecimento do CIS não seguem as regras padrão de ordenação de componentes nas fórmulas do Image Builder. Os componentes de endurecimento do CIS sempre são executados por último para garantir que os testes de benchmark sejam executados em relação à sua imagem de saída.

Componentes de hardening do STIG gerenciados pela Amazon para o Image Builder

Os Guias de Implementação Técnica de Segurança (STIGs) são os padrões de fortalecimento da configuração criados pela Agência de Sistemas de Informação de Defesa (DISA) para proteger sistemas de informação e software. Para que seus sistemas estejam em conformidade com os padrões STIG, você deve instalar, definir e testar uma variedade de configurações de segurança.

O Image Builder fornece componentes de fortalecimento do STIG para ajudá-lo a compilar, de forma mais eficiente, imagens compatíveis para os padrões do STIG de base. Esses componentes do STIG verificam se há configurações incorretas e executam um script de correção. Não há cobranças adicionais pelo uso de componentes em conformidade com o STIG.

> **Important**
>
> Com poucas exceções, os componentes de fortalecimento do STIG não instalam pacotes de terceiros. Se pacotes de terceiros já estiverem instalados na instância e se houver algo

relacionado STIGs que o Image Builder suporta para esse pacote, o componente de proteção os aplica.

Esta página lista tudo o STIGs que o Image Builder suporta e que são aplicados às EC2 instâncias que o Image Builder executa quando você cria e testa uma nova imagem. Se quiser aplicar configurações adicionais do STIG à sua imagem, você pode criar um componente personalizado para configurá-la. Para obter mais informações sobre os componentes personalizados e como criá-los, consulte Usar componentes para personalizar sua imagem do Image Builder.

Quando você cria uma imagem, os componentes de endurecimento do STIG registram se os suportados foram aplicados ou STIGs ignorados. Recomendamos que você revise os logs do Image Builder para suas imagens que usam componentes de fortalecimento do STIG. Para obter mais informações sobre como acessar e revisar os logs do Image Builder, consulte Solucionar problemas em compilação de pipelines.

Níveis de conformidade

- Alto (categoria I)

  O risco mais grave. Inclui qualquer vulnerabilidade que possa resultar em perda de confidencialidade, disponibilidade ou integridade.
- Médio (categoria II)

  Inclui qualquer vulnerabilidade que possa resultar em perda de confidencialidade, disponibilidade ou integridade, mas os riscos podem ser mitigados.
- Baixo (categoria III)

  Qualquer vulnerabilidade que degrade medidas de proteção contra perda de confidencialidade, disponibilidade ou integridade.

Tópicos

- Componentes de fortalecimento do STIG do Windows
- Log do histórico de versões do STIG para Windows
- Componentes de fortalecimento do STIG do Linux
- Log do histórico de versões do STIG para Linux
- Componente validador de conformidade do SCAP

Componentes de fortalecimento do STIG do Windows

AWSTOE Os componentes de proteção do Windows STIG foram projetados para servidores autônomos e aplicam a Política de Grupo Local. Os componentes de proteção compatíveis com STIG são instalados InstallRoot do Departamento de Defesa (DoD) na infraestrutura do Windows para baixar, instalar e atualizar os certificados do DoD. Eles também removem certificados desnecessários para manter a conformidade com o STIG. Atualmente, as linhas de base do STIG são compatíveis para as seguintes versões do Windows Server: 2012 R2, 2016, 2019 e 2022.

Esta seção lista as configurações atuais de cada um dos componentes de fortalecimento do STIG do Windows, seguidas por um log do histórico de versões.

STIG do Windows baixa (categoria III)

A lista a seguir contém configurações do STIG que o componente de fortalecimento aplica à sua infraestrutura. Se uma configuração compatível não for aplicável à sua infraestrutura, o componente de fortalecimento ignora essa configuração e segue em frente. Por exemplo, algumas configurações STIG podem não se aplicar a servidores autônomos. Políticas específicas da organização também podem afetar quais configurações o componente de fortalecimento aplica, como um requisito para que os administradores revisem as configurações do documento.

Para obter uma lista completa do Windows STIGs, consulte a Biblioteca de STIGs Documentos. Para obter informações sobre como visualizar a lista completa, consulte STIG Viewing Tools.

- Windows Server 2022 STIG versão 2 versão 4

  V-254335, V-254336, V-254337, V-254338, V-254351, V-254357, V-254363 e V-254481
- Windows Server 2019 STIG versão 3 versão 4

  V-205691, V-205819, V-205858, V-205859, V-205860, V-205870, V-205871 e V-205923
- Windows Server 2016 STIG versão 2 versão 10

  V-224916, V-224917, V-224918, V-224919, V-224931, V-224942 e V-225060
- Windows Server 2012 R2 MS STIG Versão 3 Release 5

  V-225250, V-225318, V-225319, V-225324, V-225327, V-225328, V-225330, V-225332, V-225333, V-225334, V-225335, 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-225412, V-225459, V-225460, V-225462, V-225468, V-225473, V-225476, V-225476, V-225476, V-225479, V-225480, V-225481,

V-225482, V-225483, V-225484, V-225485, V-225488, V-225489, V-225490, V-225511, V-225514, V-225525, V-225526, V-225536 e V-225537

- Microsoft.NET Framework 4.0 STIG versão 2 versão 6

  Nenhuma configuração STIG é aplicada ao Microsoft .NET Framework para vulnerabilidades de Categoria III.

- Firewall do Windows STIG versão 2 versão 2

  V-241994, V-241995, V-241996, V-241999, V-242000, V-242001, V-242006, V-242007 e V-242008

- Internet Explorer 11 STIG Versão 2 Versão 5

  V-223016, V-223056 e V-223078

- Microsoft Edge STIG versão 2 versão 2 (somente Windows Server 2022)

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

Windows STIG Medium (categoria III)

A lista a seguir contém configurações do STIG que o componente de fortalecimento aplica à sua infraestrutura. Se uma configuração compatível não for aplicável à sua infraestrutura, o componente de fortalecimento ignora essa configuração e segue em frente. Por exemplo, algumas configurações STIG podem não se aplicar a servidores autônomos. Políticas específicas da organização também podem afetar quais configurações o componente de fortalecimento aplica, como um requisito para que os administradores revisem as configurações do documento.

Para obter uma lista completa do Windows STIGs, consulte a Biblioteca de STIGs Documentos. Para obter informações sobre como visualizar a lista completa, consulte STIG Viewing Tools.

> **Note**
>
> Os componentes de proteção do Windows STIG Medium incluem todas as configurações STIG listadas que AWSTOE se aplicam aos componentes do Windows STIG Low Hardening, além das configurações STIG listadas especificamente para vulnerabilidades da Categoria II.

- Windows Server 2022 STIG versão 2 versão 4

Inclui todas as configurações do STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades da Categoria III (Baixa), além de:

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-254289, 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-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-254364, V-254364 365, V-254366, 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-254382, V-254383, V-254384, V-254431, V-254432, V-254433, V-254435, V-254436, V-254438, V-254439, V-254442, V-254443, V-254444, V-254445, V-254449, V-254450, V-254451, V-254452, V-254453, V-254454, V-254455, V-254456, V-254459, V-254460, V-254461, V-254462, V-254463, V-254464, V-254468, V-254470, V-254471, V-254472, V-254473, V-254477, V-254478, V-254479, V-254480, V-254482, V-254483, V-254484, V-254485, V-254485 254486, V-254487, 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-254507, V-254508, V-254509, V-254510, V-254511 e V-254512

- Windows Server 2019 STIG versão 3 versão 4

  Inclui todas as configurações do STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades da Categoria III (Baixa), além de:

  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-205644, V-205648, V-205649, V-205650, V-205651, V-205651, V-205651 2, 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-205682, V-205683, V-205683 5684, V-205685, V-205686, 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-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-205765, V-205766, V-205767, 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-205784, V-205795, V-205795, V-205796, V-205797, V-205798, 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-205824 25, 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-205865, 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 e V-257503

- Windows Server 2016 STIG versão 2 versão 10

  Inclui todas as configurações do STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades da Categoria III (Baixa), além de:

  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-224869, V-224870, V-224871, V-224872, V-224873, V-224881, V-224882, V-224883, V-224884, V-224885, V-224886, V-224887, V-224888, V-224889, V-224890, V-224891, V-224892, V-224893, V-224894, V-224895, 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-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-224947, 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-225015, V-225015 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-225040 25041, V-225042, V-225043, V-225047, V-225049, V-225050, V-225051, V-225052, V-225055, V-225056, V-225057, V-225058, V-225059, V-225061, V-225062, V-225063, V-225064, V-225065, V-225066, V-225067, V-225068, V-225069, V-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, V-236000 e V-257502

- Windows Server 2012 R2 MS STIG Versão 3 Release 5

  Inclui todas as configurações do STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades da Categoria III (Baixa), além de:

  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-225275 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-225291, V-225291, V-225291, 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-225304, V-225305, V-225314, V-225315, V-225315, V-225315 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-225389, 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-225411, 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-225478, 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-225518, 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 38, 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-225559, V-225560, V-225561, V-225562, V-225563, V-225564, V-225565, V-225566, V-225566, V-225567, V-225568, V-225569, V-225570, V-225571, V-225572, V-225573 e V-225574

- Microsoft.NET Framework 4.0 STIG versão 2 versão 6

  Inclui todas as configurações do STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades da Categoria III (Baixa), além do V-225238

- Firewall do Windows STIG versão 2 versão 2

  Inclui todas as configurações do STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades da Categoria III (Baixa), além de:

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

- Internet Explorer 11 STIG Versão 2 Versão 5

  Inclui todas as configurações do STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades da Categoria III (Baixa), além de:

  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-223054 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-227430, 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-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-223123, V-223124, V-223125, V-223126, V-223127, V-223128, V-223129, V-223130, V-223131, V-223131 23132, 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, V-223148, V-223148 149, V-250540 e V-250541

- Microsoft Edge STIG versão 2 versão 2 (somente Windows Server 2022)

  Inclui todas as configurações do STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades da Categoria III (Baixa), além de:

  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-235737, 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-235760, V-235761, V-235763, V-235764, V-235766, V-235767, V-235768, V-235769, V-235770, V-235771, V-235772, V-235773, V-235774 e V-246736

- Microsoft Defender STIG Versão 2 Versão 4

  Inclui todas as configurações do STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades da Categoria III (Baixa), além de:

  V-213427, V-213429, V-213430, V-213431, V-213432, V-213433, V-213434, V-213435, V-213436, V-213437, V-213438, V-213439, V-213440, V-213441, V-213442, V-213443, V-213444, V-213445, V-213446, V-213447, V-213448, V-213449, V-213450, V-213451, V-213455, V-213464, V-213465 e V-213466

STIG do Windows High (categoria I)

A lista a seguir contém configurações do STIG que o componente de fortalecimento aplica à sua infraestrutura. Se uma configuração compatível não for aplicável à sua infraestrutura, o componente de fortalecimento ignora essa configuração e segue em frente. Por exemplo, algumas configurações STIG podem não se aplicar a servidores autônomos. Políticas específicas da organização também

podem afetar quais configurações o componente de fortalecimento aplica, como um requisito para que os administradores revisem as configurações do documento.

Para obter uma lista completa do Windows STIGs, consulte a Biblioteca de STIGs Documentos. Para obter informações sobre como visualizar a lista completa, consulte STIG Viewing Tools.

> **Note**
>
> Os componentes de proteção do Windows STIG High incluem todas as configurações STIG listadas que AWSTOE se aplicam aos componentes de fortalecimento do Windows STIG Low e do Windows STIG Medium, além das configurações STIG listadas especificamente para vulnerabilidades da Categoria I.

- Windows Server 2022 STIG versão 2 versão 4

  Inclui todas as configurações do STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades das Categorias II e III (Média e Baixa), além de:

  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 e V-254500
- Windows Server 2019 STIG versão 3 versão 4

  Inclui todas as configurações do STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades das Categorias II e III (Média e Baixa), além de:

  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 e V-205919
- Windows Server 2016 STIG versão 2 versão 10

  Inclui todas as configurações do STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades das Categorias II e III (Média e Baixa), além de:

  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 e V-225079
- Windows Server 2012 R2 MS STIG Versão 3 Release 5

  Inclui todas as configurações do STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades das Categorias II e III (Média e Baixa), além de:

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

- Microsoft.NET Framework 4.0 STIG versão 2 versão 6

  Inclui todas as configurações do STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades das Categorias II e III (Média e Baixa) do Microsoft .NET Framework. Nenhuma configuração adicional do STIG se aplica a vulnerabilidades da Categoria I.

- Firewall do Windows STIG versão 2 versão 2

  Inclui todas as configurações do STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades das Categorias II e III (Média e Baixa), além de:

  V-241992, V-241997 e V-242002

- Internet Explorer 11 STIG Versão 2 Versão 5

  V-252910

- Microsoft Edge STIG versão 2 versão 2 (somente Windows Server 2022)

  Inclui todas as configurações do STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades das Categorias II e III (Média e Baixa), além de:

  V-235758 e V-235759

- Microsoft Defender STIG Versão 2 Versão 4

  Inclui todas as configurações do STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades das Categorias II e III (Média e Baixa), além de:

  V-213426, V-213452 e V-213453

Log do histórico de versões do STIG para Windows

Esta seção registra o histórico de versões do componente de fortalecimento do Windows para as atualizações trimestrais do STIG. Para ver as alterações e as versões publicadas de um trimestre, escolha o título para expandir as informações.

Alterações no terceiro trimestre de 2025 - 9/04/2025 (sem alterações):

Não houve alterações no componente STIGS do Windows para a versão do terceiro trimestre de 2025.

Alterações no segundo trimestre de 2025 - 26/06/2025:

Versões atualizadas do STIG e aplicou o STIGS para a versão do segundo trimestre de 2025 da seguinte forma:

STIG-Build-Windows-Low versão 2025.2.x

- Windows Server 2022 STIG versão 2 versão 4
- Windows Server 2019 STIG versão 3 versão 4
- Windows Server 2016 STIG versão 2 versão 10
- Windows Server 2012 R2 MS STIG Versão 3 Release 5
- Microsoft.NET Framework 4.0 STIG versão 2 versão 6
- Firewall do Windows STIG versão 2 versão 2
- Internet Explorer 11 STIG Versão 2 Versão 5
- Microsoft Edge STIG versão 2 versão 2 (somente Windows Server 2022)

STIG-Build-Windows-Medium versão 2025.2.x

- Windows Server 2022 STIG versão 2 versão 4
- Windows Server 2019 STIG versão 3 versão 4
- Windows Server 2016 STIG versão 2 versão 10
- Windows Server 2012 R2 MS STIG Versão 3 Release 5
- Microsoft.NET Framework 4.0 STIG versão 2 versão 6
- Firewall do Windows STIG versão 2 versão 2
- Internet Explorer 11 STIG Versão 2 Versão 5
- Microsoft Edge STIG versão 2 versão 2 (somente Windows Server 2022)
- Defender STIG Versão 2 Versão 4

STIG-Build-Windows-High versão 2025.2.x

- Windows Server 2022 STIG versão 2 versão 4

- Windows Server 2019 STIG versão 3 versão 4
- Windows Server 2016 STIG versão 2 versão 10
- Windows Server 2012 R2 MS STIG Versão 3 Release 5
- Microsoft.NET Framework 4.0 STIG versão 2 versão 6
- Firewall do Windows STIG versão 2 versão 2
- Internet Explorer 11 STIG Versão 2 Versão 5
- Microsoft Edge STIG versão 2 versão 2 (somente Windows Server 2022)
- Defender STIG Versão 2 Versão 4

Alterações no primeiro trimestre de 2025 - 05/04/2025:

STIGS atualizado para Internet Explorer 11 STIG versão 2 versão 5 para todos os componentes do STIG para a versão do primeiro trimestre de 2025.

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

Alterações no quarto trimestre de 2024 - 02/04/2025:

Versões atualizadas do STIG e aplicou STIGS para a versão do quarto trimestre de 2024 da seguinte forma:

STIG-Build-Windows-Low versão 2024.4.0

- Windows Server 2022 STIG versão 2 versão 2
- Windows Server 2019 STIG versão 3 versão 2
- Windows Server 2016 STIG versão 2 versão 9
- Windows Server 2012 R2 MS STIG Versão 3 Release 5
- Microsoft .NET Framework 4.0 STIG Versão 2 Release 2
- Firewall do Windows STIG versão 2 versão 2
- Internet Explorer 11 STIG Versão 2 Versão 5
- Microsoft Edge STIG versão 2 versão 2 (somente Windows Server 2022)

STIG-Build-Windows-Medium versão 2024.4.0

- Windows Server 2022 STIG versão 2 versão 2
- Windows Server 2019 STIG versão 3 versão 2
- Windows Server 2016 STIG versão 2 versão 9
- Windows Server 2012 R2 MS STIG Versão 3 Release 5
- Microsoft .NET Framework 4.0 STIG Versão 2 Release 2
- Firewall do Windows STIG versão 2 versão 2
- Internet Explorer 11 STIG Versão 2 Versão 5
- Microsoft Edge STIG versão 2 versão 2 (somente Windows Server 2022)
- Defender STIG Versão 2 Versão 4

STIG-Build-Windows-High versão 2024.4.0

- Windows Server 2022 STIG versão 2 versão 2
- Windows Server 2019 STIG versão 3 versão 2
- Windows Server 2016 STIG versão 2 versão 9
- Windows Server 2012 R2 MS STIG Versão 3 Release 5
- Microsoft .NET Framework 4.0 STIG Versão 2 Release 2
- Firewall do Windows STIG versão 2 versão 2
- Internet Explorer 11 STIG Versão 2 Versão 5
- Microsoft Edge STIG versão 2 versão 2 (somente Windows Server 2022)
- Defender STIG Versão 2 Versão 4

Alterações no terceiro trimestre de 2024 - 10/04/2023 (sem alterações):

Não houve alterações no componente STIGS do Windows para a versão do terceiro trimestre de 2024.

Alterações no segundo trimestre de 2024 (10/5/2024; sem alterações):

Não houve alterações no componente STIGS do Windows para o lançamento do segundo trimestre de 2024.

Alterações no primeiro trimestre de 2024 (6/2/2024; sem alterações):

Não houve alterações no componente STIGS do Windows para o lançamento do primeiro trimestre de 2024.

Alterações no quarto trimestre de 2023 (4/12/2023; sem alterações):

Não houve alterações no componente STIGS do Windows para o lançamento do quarto trimestre de 2023.

Alterações no terceiro trimestre de 2023 - 04/10/2023 (sem alterações):

Não houve alterações no componente STIGS do Windows para o release do terceiro trimestre de 2023.

Alterações no segundo trimestre de 2023 - 03/05/2023 (sem alterações):

Não houve alterações no componente STIGS do Windows para o release do segundo trimestre de 2023.

Alterações no primeiro trimestre de 2023 - 27/03/2023 (sem alterações):

Não houve alterações no componente STIGS do Windows para o release do primeiro trimestre de 2023.

Alterações no quarto trimestre de 2022 - 01/02/2023:

Versões do STIG atualizadas e STIGS aplicados para o release do quarto trimestre de 2022 da seguinte forma:

STIG-Build-Windows-Low versão 2022.4.x

- Windows Server 2022 STIG versão 1 release 1
- Windows Server 2019 STIG Versão 2 Release 5
- Windows Server 2016 STIG Versão 2 Release 5
- Windows Server 2012 R2 MS STIG Versão 3 Release 5
- Microsoft .NET Framework 4.0 STIG Versão 2 Release 2
- Windows Firewall STIG Versão 2 Release 1
- Internet Explorer 11 STIG Versão 2 Release 3

- Microsoft Edge STIG Versão 1 Release 6 (somente Windows Server 2022)

STIG-Build-Windows-Medium versão 2022.4.x

- Windows Server 2022 STIG versão 1 release 1
- Windows Server 2019 STIG Versão 2 Release 5
- Windows Server 2016 STIG Versão 2 Release 5
- Windows Server 2012 R2 MS STIG Versão 3 Release 5
- Microsoft .NET Framework 4.0 STIG Versão 2 Release 2
- Windows Firewall STIG Versão 2 Release 1
- Internet Explorer 11 STIG Versão 2 Release 3
- Microsoft Edge STIG Versão 1 Release 6 (somente Windows Server 2022)
- Defender STIG Versão 2 Release 4 (somente Windows Server 2022)

STIG-Build-Windows-High versão 2022.4.x

- Windows Server 2022 STIG versão 1 release 1
- Windows Server 2019 STIG Versão 2 Release 5
- Windows Server 2016 STIG Versão 2 Release 5
- Windows Server 2012 R2 MS STIG Versão 3 Release 5
- Microsoft .NET Framework 4.0 STIG Versão 2 Release 2
- Windows Firewall STIG Versão 2 Release 1
- Internet Explorer 11 STIG Versão 2 Release 3
- Microsoft Edge STIG Versão 1 Release 6 (somente Windows Server 2022)
- Defender STIG Versão 2 Release 4 (somente Windows Server 2022)

Alterações no terceiro trimestre de 2022 - 30/09/2022 (sem alterações):

Não houve alterações no componente STIGS do Windows para o release do terceiro trimestre de 2022.

Alterações no segundo trimestre de 2022 - 02/08/2022:

Versões do STIG atualizadas e STIGS aplicados para o release do segundo trimestre de 2022.

STIG-Build-Windows-Low versão 1.5.x

- Windows Server 2019 STIG Versão 2 Release 4
- Windows Server 2016 STIG Versão 2 Release 4
- Windows Server 2012 R2 MS STIG Versão 3 Release 3
- Microsoft .NET Framework 4.0 STIG Versão 2 Release 1
- Windows Firewall STIG Versão 2 Release 1
- Internet Explorer 11 STIG Versão 1 Release 19

STIG-Build-Windows-Medium versão 1.5.x

- Windows Server 2019 STIG Versão 2 Release 4
- Windows Server 2016 STIG Versão 2 Release 4
- Windows Server 2012 R2 MS STIG Versão 3 Release 3
- Microsoft .NET Framework 4.0 STIG Versão 2 Release 1
- Windows Firewall STIG Versão 2 Release 1
- Internet Explorer 11 STIG Versão 1 Release 19

STIG-Build-Windows-High versão 1.5.x

- Windows Server 2019 STIG Versão 2 Release 4
- Windows Server 2016 STIG Versão 2 Release 4
- Windows Server 2012 R2 MS STIG Versão 3 Release 3
- Microsoft .NET Framework 4.0 STIG Versão 2 Release 1
- Windows Firewall STIG Versão 2 Release 1
- Internet Explorer 11 STIG Versão 1 Release 19

Alterações no primeiro trimestre de 2022 - 02/08/2022 (sem alterações):

Não houve alterações no componente STIGS do Windows para o release do primeiro trimestre de 2022.

Alterações no quarto trimestre de 2021 - 20/12/2021:

Versões do STIG atualizadas e STIGS aplicados para o release do quarto trimestre de 2021.

STIG-Build-Windows-Low versão 1.5.x

- Windows Server 2019 STIG Versão 2 Release 3
- Windows Server 2016 STIG Versão 2 Release 3
- Windows Server 2012 R2 MS STIG Versão 3 Release 3
- Microsoft .NET Framework 4.0 STIG Versão 2 Release 1
- Windows Firewall STIG Versão 2 Release 1
- Internet Explorer 11 STIG Versão 1 Release 19

STIG-Build-Windows-Medium versão 1.5.x

- Windows Server 2019 STIG Versão 2 Release 3
- Windows Server 2016 STIG Versão 2 Release 3
- Windows Server 2012 R2 MS STIG Versão 3 Release 3
- Microsoft .NET Framework 4.0 STIG Versão 2 Release 1
- Windows Firewall STIG Versão 2 Release 1
- Internet Explorer 11 STIG Versão 1 Release 19

STIG-Build-Windows-High versão 1.5.x

- Windows Server 2019 STIG Versão 2 Release 3
- Windows Server 2016 STIG Versão 2 Release 3
- Windows Server 2012 R2 MS STIG Versão 3 Release 3
- Microsoft .NET Framework 4.0 STIG Versão 2 Release 1
- Windows Firewall STIG Versão 2 Release 1
- Internet Explorer 11 STIG Versão 1 Release 19

Alterações no terceiro trimestre de 2021 - 30/09/2021:

Versões do STIG atualizadas e STIGS aplicados para o release do terceiro trimestre de 2021.

STIG-Build-Windows-Low versão 1.4.x

- Windows Server 2019 STIG Versão 2 Release 2
- Windows Server 2016 STIG Versão 2 Release 2

- Windows Server 2012 R2 MS STIG Versão 3 Release 2
- Microsoft .NET Framework 4.0 STIG Versão 2 Release 1
- Windows Firewall STIG Versão 1 Release 7
- Internet Explorer 11 STIG Versão 1 Release 19

STIG-Build-Windows-Medium versão 1.4.x

- Windows Server 2019 STIG Versão 2 Release 2
- Windows Server 2016 STIG Versão 2 Release 2
- Windows Server 2012 R2 MS STIG Versão 3 Release 2
- Microsoft .NET Framework 4.0 STIG Versão 2 Release 1
- Windows Firewall STIG Versão 1 Release 7
- Internet Explorer 11 STIG Versão 1 Release 19

STIG-Build-Windows-High versão 1.4.x

- Windows Server 2019 STIG Versão 2 Release 2
- Windows Server 2016 STIG Versão 2 Release 2
- Windows Server 2012 R2 MS STIG Versão 3 Release 2
- Microsoft .NET Framework 4.0 STIG Versão 2 Release 1
- Windows Firewall STIG Versão 1 Release 7
- Internet Explorer 11 STIG Versão 1 Release 19

Componentes de fortalecimento do STIG do Linux

Esta seção contém informações sobre os componentes de fortalecimento do STIG do Linux, seguidas por um log do histórico de versões. Se a distribuição do Linux não tiver configurações de STIG próprias, o componente de fortalecimento aplicará as configurações do RHEL.

Os componentes Linux têm parâmetros de entrada opcionais que ajudam você a personalizar os seguintes comportamentos para sua instância Linux.

- InstallPackages (string) Se o valor for`No`, o componente não instalará nenhum pacote de software adicional. Se o valor for`Yes`, o componente instala pacotes de software adicionais que são necessários para máxima conformidade. O padrão é `No`.

- SetDoDConsentBanner (string) Se o valor for`No`, o banner de consentimento do DoD não é exibido quando você se conecta a uma instância que tem um dos componentes do STIG Linux instalado. Se o valor for`Yes`, o banner de consentimento do DoD será exibido antes de você fazer login quando você se conectar a uma instância que tenha um dos componentes do STIG Linux instalado. Você deve reconhecer o banner antes de fazer login. O padrão é `No`.

  Para ver um exemplo do banner de consentimento, consulte o Aviso de Privacidade e Consentimento do Departamento de Defesa que aparece quando você acessa o site dos Serviços de Documentos da DLA.

O componente de fortalecimento aplica as configurações do STIG compatíveis à infraestrutura com base na distribuição do Linux, da seguinte forma:

Configurações do Red Hat Enterprise Linux (RHEL) 7 STIG

- RHEL 7
- CentOS 7
- Amazon Linux (2AL2)

Configurações do RHEL 8 STIG

- RHEL 8
- CentOS 8

Configurações do RHEL 9 STIG

- RHEL 9
- CentOS Stream 9

Linux STIG Low (categoria III)

A lista a seguir contém configurações do STIG que o componente de fortalecimento aplica à sua infraestrutura. Se uma configuração compatível não for aplicável à sua infraestrutura, o componente de fortalecimento ignora essa configuração e segue em frente. Por exemplo, algumas configurações STIG podem não se aplicar a servidores autônomos. Políticas específicas da organização também podem afetar quais configurações o componente de fortalecimento aplica, como um requisito para que os administradores revisem as configurações do documento.

Para obter uma lista completa, consulte a Biblioteca de STIGs documentos. Para obter informações sobre como visualizar a lista completa, consulte STIG Viewing Tools.

RHEL 7 STIG Versão 3 Versão 15

- RHEL 7/CentOS 7/ AL2

  V-204452, V-204576 e V-204605

RHEL 8 STIG Versão 2 Versão 4

- RHEL 8/CentOS 8

  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-230496, V-230497, V-230498, V-230499 e V-244527

RHEL 9 STIG Versão 2 Versão 5

- RHEL 9/CentOS Stream 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 e V-258173

Amazon Linux 2023 STIG versão 1 versão 1

V-274141

SLES 12 STIG Versão 3 Versão 3

V-217108, V-217113, V-217140, V-217198, V-217209, V-217211, V-217212, V-217213, V-217214, V-217215, V-217216, V-217236, V-217237, V-217238, V-217239, V-217282 e V-255915

SLES 15 STIG Versão 2 Versão 5

V-234811, V-234850, V-234868, V-234868, V-234873, V-234873, V-234905, V-234907, V-234908, V-234909, V-234933, V-234934, V-234935, V-234936, V-234955, V-234963, V-234967 e V-255921

Ubuntu 18.04 STIG versão 2 versão 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-219327, V-219332 e V-219333

Ubuntu 20.04 STIG Versão 2 Versão 3

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 e V-238373

Ubuntu 22.04 STIG Versão 2 Versão 5

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

Ubuntu 24.04 STIG Versão 1 Versão 2

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

Linux STIG Medium (categoria II)

A lista a seguir contém configurações do STIG que o componente de fortalecimento aplica à sua infraestrutura. Se uma configuração compatível não for aplicável à sua infraestrutura, o componente de fortalecimento ignora essa configuração e segue em frente. Por exemplo, algumas configurações STIG podem não se aplicar a servidores autônomos. Políticas específicas da organização também podem afetar quais configurações o componente de fortalecimento aplica, como um requisito para que os administradores revisem as configurações do documento.

Para obter uma lista completa, consulte a Biblioteca de STIGs documentos. Para obter informações sobre como visualizar a lista completa, consulte STIG Viewing Tools.

> **Note**
>
> Os componentes de fortalecimento Linux STIG Medium incluem todas as configurações STIG listadas que AWSTOE se aplicam aos componentes Linux STIG Low Hardening, além das configurações STIG listadas especificamente para vulnerabilidades da Categoria II.

RHEL 7 STIG Versão 3 Versão 15

Inclui todas as configurações STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades da Categoria III (Baixa) para esta distribuição do Linux, além de:

- 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-204418, V-204420, V-204422, V-204423, V-204426, 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-204511, V-204511 12, V-204514, 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-204546, V-204546, V-204546, V-204546, V-204546, V-204546 47, V-204548, V-204549, V-204550, V-204551, V-204552, V-204553, V-204554, V-204555, V-204556, V-204557, V-204558, V-204559, V-204560, V-204562, V-204563, V-204564, V-204565, V-204566, V-204567, V-204568, V-204572, V-204579, V-204584, V-204585, V-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 4610, 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-255927, e V-256970

RHEL 8 STIG Versão 2 Versão 4

Inclui todas as configurações STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades da Categoria III (Baixa) para esta distribuição do Linux, além de:

- RHEL 8/CentOS 8

  V-230222, 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-230256, V-230257, V-230257, V-230259, V-230260, V-230261, V-230262, V-230266, V-230267, V-230268, V-230271, V-230273, V-230275, V-230276, V-230277, V-230278, V-230279, V-230280, V-230282, V-230286, 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-230318, V-230319, V-230320, V-230321, V-230322, V-230324, V-230325, V-230326, V-230327, V-230330, V-230333, V-230335, V-230337, V-230339, V-230341, V-230343, V-230345, V-230347, V-230348, V-230352, V-230353, V-230354, V-230356, V-230357, V-230357, V-230358, V-230359, V-230359 230360, V-230361, V-230362, V-230363, V-230365, V-230366, V-230368, V-230369, V-230370, V-230372, V-230373, V-230375, V-230376, V-230377, V-230378, V-230380, V-230382, V-230383, V-230383, V-230385, V-230386, V-230387, V-230389, V-230390, V-230392, V-230393, V-230394, V-230396, V-230397, V-230398, V-230399, V-230400, V-230401, V-230402, V-230403, V-230404, V-230405, V-230406,

V-230406, V-230407, V-230408, V-230409, V-230410, V-230411, V-230412, V-230413, V-230418, V-230419, V-230421, V-230422, V-230423, V-230424, V-230425, V-230426, V-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-230448, V-230449, V-230455, V-230456, V-230462, V-230463, V-230463, V-230464, V-230465 V-230466, V-230467, V-230470, V-230471, V-230472, V-230473, V-230474, V-230475, V-230478, V-230480, V-230481, V-230482, V-230483, V-230488, V-230489, V-230493, V-230502, V-230503, V-230503, V-230505, V-23050, V-230506, V-230507, V-230523, V-230524, 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-244519, V-244523, V-244524, V-244525, V-244526 244528, V-244531, V-244533, V-244535, V-244536, V-244538, V-244539, 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-251707, V-251708, V-251709, V-251710, V-251711, V-251713, V-251714, V-251715, V-251716, V-251717, V-251718, V-256974, V-257258 e V-274877

RHEL 9 STIG Versão 2 Versão 5

Inclui todas as configurações STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades da Categoria III (Baixa) para esta distribuição do Linux, além de:

- RHEL 9/CentOS Stream 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-257800, V-257801, V-257802, V-257803, V-257804 257805, V-257806, V-257807, V-257809, V-257810, V-257811, V-257812, V-257813, V-257814, V-257815, V-257816, V-257817, V-257818, V-257825, V-257827, V-257828, V-257829 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-257887, 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 908, V-257909, V-257910, 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-257926, V-257926, V-257926, V-257926, V-257925, V-257926, V-257926, V-257926, V-257926, V-257926, V-257925, V-257926, V-257926, V-257926, V-257926, V-257925, V-257926, V-257926, V-257926, V-257926, V-25257927, 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-257954, V-257957, V-257957, V-257957, V-257953, V-257954, V-257957, V-257957, V-257957 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-257973, 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-257997, V-257998, V-257999, V-258000, V-258001, V-258002, V-258003, V-258004, 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-258046, V-258049, V-258052, V-258054, V-258055, V-258056, V-258057, V-258060, V-258063, V-258064, V-258065, V-258066, V-258077, V-258071, V-258072, V-258073, V-258074, V-258075, V-258077, V-258079, V-258080, V-258080, V-258081, V-258082, V-258083, V-258084, V-258085, V-258088, V-258089, V-258090, V-258091, V-258092, V-258093, V-258095, V-258097, V-258098, V-258099, V-258100, V-258101, V-258102, V-258103 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-258122, V-258123, V-258123, V-258123, V-258123 4, 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-258151, V-258151, V-258151, V-258152, V-258153, V-258154, V-258156, V-258157, V-258158, V-258159, V-258161, V-258162, V-258163, V-258164, V-258165, V-258166, V-258167, V-258168, V-258168, V-258168, V-258168 8169, 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-258186, V-258186, V-258186 7, 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-258205, V-258206, V-258207, V-258208, V-258209, V-258210, V-258210 8211, V-258212, V-258213, V-258214, V-258215, V-258216, V-258217, V-258218, V-258219, V-258221, V-258222, V-258223, V-258224, V-258225, V-258226, V-258227, V-258228, V-258229, V-258229, V-258229 8232, V-258233, V-258234, V-258237, V-258239, V-258240, V-270174, V-270175, V-270176, V-270177, V-272488 e V-274878

Amazon Linux 2023 STIG versão 1 versão 1

Inclui todas as configurações STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades da Categoria III (Baixa) para esta distribuição do Linux, além de:

V-273995, V-274000, V-274001, V-274002, V-274003, V-274004, V-274005, V-274006, V-274008, V-274018, V-274011, V-274012, V-274013, V-274014, V-274017, V-274018, V-274019, V-274020, V-274020, V-274021, V-274022, V-274023, V-274024, V-274026, V-274027, V-274028, V-274030,

V-274031, V-274032, V-274033, V-274034, V-274035, V-274036, V-274037, V-274040, V-274041, V-274042, V-274044, V-274044, V-274042, V-274044, V-274044, V-274042, V-274044, V-274044, V-274042, V-274044, V-274044, V-274042, V-274044, V-274044, V-274042, V-274044, V-274044, V-V-274045, V-274047, V-274048, V-274049, V-274050, V-274051, V-274053, V-274054, V-274059, V-274061, V-274062, V-274065, V-274069, V-274070, V-274071, V-274072, V-274073, V-274074, V-274074, V-2740, V-2775, V-274076, V-274077, V-274078, V-274079, V-274081, V-274082, V-274083, V-274084, V-274085, V-274086, V-274087, V-274088, V-274089, V-274090, V-274091, V-274092, V-274093, V-274094, V-274095, V-274096, V-274097, V-274098, V-274099, V-274100, V-274101, V-274102, V-274103, V-274104, V-274105, V-274106, V-274107, V-274108, V-274109, V-274110, V-274111, V-274112, V-274113, V-274114 V-274115, V-274116, V-274117, V-274119, V-274120, V-274121, V-274122, V-274123, V-274124, V-274125, V-274126, V-274127, V-274128, V-274129, V-274130, V-274131, V-274132, V-274133, V-274134, V-274135, V-274136, V-274137, V-274138, V-274139, V-274140, V-274142, V-274143, V-274144, V-274145, V-274147, V-274149, V-274151, V-274152, V-274154, V-274155, V-274155, V-274155 6, V-274157, V-274160, V-274161, V-274162, V-274164, V-274165, V-274166, V-274167, V-274168, V-274169, V-274170, V-274173, V-274174, V-274177, V-274181, V-274182, V-274185 e V-274187

SLES 12 STIG Versão 3 Versão 3

Inclui todas as configurações STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades da Categoria III (Baixa) para esta distribuição do Linux, além de:

V-217102, V-217105, V-217110, V-217116, V-217117, V-217118, V-217119, V-217120, V-217121, V-217122, V-217124, V-217125, V-217126, V-217127, V-217128, V-217130, V-217134, V-217138 17143, V-217147, V-217152, V-217153, V-217154, V-217155, V-217156, V-217158, V-217161, V-217163, V-217166, V-217167, V-217168, V-217169, V-217170, V-217171, V-2171 182, V-217183, V-217188, V-217190, V-217191, V-217194, V-217195, V-217196, V-217197, V-217200, V-217201, V-217202, V-217203, V-217204, V-217205, V-217206, V-217207, V-217208, V-217210, V-217217, V-217218, V-217223, V-217227, V-217230, V-217240, V-217241, V-217242, V-217243, V-217244, V-217245, V-217246, V-217247, V-217248, V-217249, V-217250, V-217251, V-217252, V-217253, V-217254, V-217254, V-217254, V-217254, V-217254 17255, V-217257, V-217258, V-217260, V-217265, V-217266, V-217267, V-217269, V-217272, V-217273, V-217274, V-217275, V-217276, V-217277, V-217278, V-217279, V-21727 17280, V-217283, V-217284, V-217286, V-217287, V-217288, V-217289, V-217290, V-217291, V-217292, V-217293, V-217294, V-217295, V-217296, V-217299, V-217300, V-217300, V-21717301, V-217302, V-233308, V-237606, V-237607, V-237608, V-237609, V-237610, V-237611, V-237612, V-237613, V-237614, V-237615, V-237616, V-237617, V-237618, V-237619, V-237620, V-237621, V-237622, V-237623, V-251720, V-251722, V-255914 e V-256981

SLES 15 STIG Versão 2 Versão 5

Inclui todas as configurações STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades da Categoria III (Baixa) para esta distribuição do Linux, além de:

V-234802, V-234807, V-234808, V-234809, V-234813, V-234815, V-234817, V-234821, V-234822, V-234823, V-234825, V-234827, V-234828, V-234829, V-234830, V-234832, V-234833, V-234834, V-234834, V-234834 234835, V-234836, V-234837, V-234838, V-234839, V-234840, V-234841, V-234842, V-234843, V-234844, V-234845, V-234848, V-234854, V-234855, V-234856, V-234857, V-234858, V-234861, V-234862, V-234862, V-234862 234863, V-234869, V-234870, V-234875, V-234880, V-234881, V-234882, V-234883, V-234884, V-234885, V-234886, V-234887, V-234888, V-234889, V-234891, V-234895, V-234896, V-234897, V-234899, V-234899, V-234899 234900, V-234901, V-234902, V-234903, V-234904, V-234906, V-234910, V-234911, V-234912, V-234913, V-234914, V-234918, V-234924, V-234928, V-234932, V-234937, V-234938, V-234939, V-234940, V-234941, V-234942, V-234943, V-234944, V-234945, V-234946, V-234947, V-234948, V-234949, V-234950, V-234951, V-234952, V-234954, V-234956, V-234957, V-234958, V-234959, V-234961, V-234962, V-234964, V-234966, V-234969, V-234973, V-234975, V-234976, V-234977, V-234978, V-234979, V-234981, V-234982, V-234983, V-234991, V-235002, V-235003, V-235007, V-235008, V-235009, V-235010, V-235013, V-235014, V-235015, V-235016, V-235017, V-235018, V-235019, V-235020, V-235021, V-235022, V-235023, V-235024, V-235025, V-235026, V-235028, V-235029, V-235030, V-251724, V-255920, V-256983, e V-274879

Ubuntu 18.04 STIG versão 2 versão 15

Inclui todas as configurações STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades da Categoria III (Baixa) para esta distribuição do Linux, além de:

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-2193, V-22219194, 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-219209, V-219209 213, V-219214, V-219215, V-219216, V-219217, V-219218, V-219219, V-219221, V-219222, V-219223, V-219224, V-219225, V-219226, V-219227, V-219228, V-219228, V-219228 29, 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-219269, V-219270, V-219271, V-219272, V-219273, V-219274, V-219275, V-219276, V-219277, V-219281, V-219287, V-219291, V-219291, V-219296, 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, V-219335, V-219335, V-22235 336, V-219337, V-219338, V-219339, V-219342, V-219344, V-233779, V-233780 e V- 255906

Ubuntu 20.04 STIG Versão 2 Versão 3

Inclui todas as configurações STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades da Categoria III (Baixa) para esta distribuição do Linux, além de:

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-238236, V-238236, V-238236, 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-238299, V-238300, V-238301, V-238303, V-238304, V-238309, V-238310, V-238315, V-238316, V-238317, V-238318, V-238319, V-238319, V-238319, V-238320, V-238324, V-238325, V-238329, V-238330, 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-238360, V-238369, V-238370, V-238371, V-238376, V-238377, V-238378, V-25151505, V-255912, V-274852 e V-274853

Ubuntu 22.04 STIG Versão 2 Versão 5

Inclui todas as configurações STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades da Categoria III (Baixa) para esta distribuição do Linux, além de:

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-260494, 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-260514, 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-2605547 3, 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-260590, 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-260610, 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 26, 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, V-260649, V-274862, V-274864 e V-274866

Ubuntu 24.04 STIG Versão 1 Versão 2

Inclui todas as configurações STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades da Categoria III (Baixa) para esta distribuição do Linux, além de:

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-270699, V-270700, V-270701, V-270702, V-270703, V-270704, V-270705, V-270715, V-270716, V-270718, V-270720, V-270721, V-270722, V-270723, V-270724, V-270725, V-270725, V-270725 26, V-270727, V-270728, V-270729, 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-270766, 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-270787, 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-270801, V-270801 2, V-270803, V-270804, V-270805, V-270806, V-270807, V-270808, V-270809, V-270810, V-270811, V-270812, V-270813, V-270814, V-270821, V-270822, V-270823, V-270824, V-270825, V-270826, V-270827, V-270828, V-270829, V-270829 0830, V-270831, V-270832, V-274870, V-274871, V-274872 e V-274873

Linux STIG High (categoria I)

A lista a seguir contém configurações do STIG que o componente de fortalecimento aplica à sua infraestrutura. Se uma configuração compatível não for aplicável à sua infraestrutura, o componente de fortalecimento ignora essa configuração e segue em frente. Por exemplo, algumas configurações STIG podem não se aplicar a servidores autônomos. Políticas específicas da organização também podem afetar quais configurações o componente de fortalecimento aplica, como um requisito para que os administradores revisem as configurações do documento.

Para obter uma lista completa, consulte a Biblioteca de STIGs documentos. Para obter informações sobre como visualizar a lista completa, consulte STIG Viewing Tools.

> **Note**
>
> Os componentes de fortalecimento Linux STIG High incluem todas as configurações STIG listadas que AWSTOE se aplicam aos componentes de fortalecimento Linux STIG Low e Linux STIG Medium, além das configurações STIG listadas que se aplicam especificamente às vulnerabilidades da Categoria I.

RHEL 7 STIG Versão 3 Versão 15

Inclui todas as configurações STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades das Categorias II e III (Média e Baixa) para esta distribuição do Linux, além de:

- 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 e V-204621

RHEL 8 STIG Versão 2 Versão 4

Inclui todas as configurações STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades das Categorias II e III (Média e Baixa) para esta distribuição do Linux, além de:

- RHEL 8/CentOS 8

  V-230221, V-230223, V-230226, V-230264, V-230265, V-230283, V-230284, V-230487, V-230492, V-230529, V-230530, V-230531, V-230533, V-230558, V-244540 e V-268322

RHEL 9 STIG Versão 2 Versão 5

Inclui todas as configurações STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades das Categorias II e III (Média e Baixa) para esta distribuição do Linux, além de:

- RHEL 9/CentOS Stream 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 e V-258238

Amazon Linux 2023 STIG versão 1 versão 1

Inclui todas as configurações STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades das Categorias II e III (Média e Baixa) para esta distribuição do Linux, além de:

V-273996, V-273997, V-273999, V-274007, V-274038, V-274039, V-274046, V-274052, V-274057 e V-274153

SLES 12 STIG Versão 3 Versão 3

Inclui todas as configurações STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades das Categorias II e III (Média e Baixa) para esta distribuição do Linux, além de:

V-217101, V-217139, V-217141, V-217142, V-217159, V-217160, V-217264, V-217268, V-222386 e V-251721

SLES 15 STIG Versão 2 Versão 5

Inclui todas as configurações STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades das Categorias II e III (Média e Baixa) para esta distribuição do Linux, além de:

V-234800, V-234804, V-234818, V-234852, V-234859, V-234860, V-234898, V-234984, V-234985, V-234988, V-234989, V-234990, V-235031, V-235032 e V-251725

Ubuntu 18.04 STIG versão 2 versão 15

Inclui todas as configurações STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades das Categorias II e III (Média e Baixa) para esta distribuição do Linux, além de:

V-219157, V-219158, V-219177, V-219212, V-219308, V-219314, V-219316, V-251507 e V-264388

Ubuntu 20.04 STIG Versão 2 Versão 3

Inclui todas as configurações STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades das Categorias II e III (Média e Baixa) para esta distribuição do Linux, além de:

V-238201, V-238218, V-238219, V-238326, V-238327, V-238380 e V-251504

Ubuntu 22.04 STIG Versão 2 Versão 5

Inclui todas as configurações STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades das Categorias II e III (Média e Baixa) para esta distribuição do Linux, além de:

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

Ubuntu 24.04 STIG Versão 1 Versão 2

Inclui todas as configurações STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades das Categorias II e III (Média e Baixa) para esta distribuição do Linux, além de:

V-270647, V-270648, V-270665, V-270666, V-270708, V-270711, V-270712, V-270713, V-270714, V-270717 e V-270736

Log do histórico de versões do STIG para Linux

Esta seção registra o histórico de versões do componente do Linux. Para ver as alterações e as versões publicadas de um trimestre, escolha o título para expandir as informações.

Alterações no terceiro trimestre de 2025 - 09/04/2025:

Foi adicionado suporte ao sistema operacional SUSE Linux Enterprise Server (SLES) e ao Amazon Linux 2023. Atualizou as seguintes versões do STIG e aplicou o STIGS para a versão do terceiro trimestre de 2025 para todos os níveis de conformidade ()low/medium/high:

- RHEL 7 STIG Versão 3 Versão 15
- RHEL 8 STIG Versão 2 Versão 4
- RHEL 9 STIG Versão 2 Versão 5
- Amazon Linux 2023 STIG versão 1, versão 1
- SLES 12 STIG Versão 3 Versão 3
- SLES 15 STIG Versão 2 Versão 5
- Ubuntu 18.04 STIG versão 2 versão 15
- Ubuntu 20.04 STIG Versão 2 Versão 3
- Ubuntu 22.04 STIG Versão 2 Versão 5
- Ubuntu 24.04 STIG Versão 1 Versão 2

Alterações no segundo trimestre de 2025 - 26/06/2025:

Atualizou as seguintes versões do STIG e aplicou o STIGS para a versão do segundo trimestre de 2025:

STIG-Build-Linux-Low versão 2025.2.x

- RHEL 7 STIG Versão 3 Versão 15

- RHEL 8 STIG Versão 2 Versão 3
- RHEL 9 STIG Versão 2 Versão 4
- Ubuntu 18.04 STIG versão 2 versão 15
- Ubuntu 20.04 STIG Versão 2 Versão 2
- Ubuntu 22.04 STIG Versão 2 Versão 4
- Ubuntu 24.04 STIG Versão 1 Versão 1

STIG-Build-Linux-Medium versão 2025.2.x

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

STIG-Build-Linux-High versão 2025.2.x

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

Alterações no primeiro trimestre de 2025 - 11/04/2025:

Atualizou as seguintes versões do STIG, aplicou o STIGS na versão do primeiro trimestre de 2025 e adicionou suporte ao Ubuntu 24.04:

STIG-Build-Linux-Low versão 2025.1.x

- RHEL 7 STIG Versão 3 Versão 15
- RHEL 8 STIG Versão 2 Versão 2
- RHEL 9 STIG Versão 2 Versão 3
- Ubuntu 18.04 STIG versão 2 versão 15
- Ubuntu 20.04 STIG Versão 2 Versão 2
- Ubuntu 22.04 STIG Versão 2 Versão 3
- Ubuntu 24.04 STIG Versão 1 Versão 1

STIG-Build-Linux-Medium versão 2025.1.x

- RHEL 7 STIG Versão 3 Versão 15
- RHEL 8 STIG Versão 2 Versão 2
- RHEL 9 STIG Versão 2 Versão 3
- Ubuntu 18.04 STIG versão 2 versão 15
- Ubuntu 20.04 STIG Versão 2 Versão 2
- Ubuntu 22.04 STIG Versão 2 Versão 3
- Ubuntu 24.04 STIG Versão 1 Versão 1

STIG-Build-Linux-High versão 2025.1.x

- RHEL 7 STIG Versão 3 Versão 15
- RHEL 8 STIG Versão 2 Versão 2
- RHEL 9 STIG Versão 2 Versão 3
- Ubuntu 18.04 STIG versão 2 versão 15
- Ubuntu 20.04 STIG Versão 2 Versão 2
- Ubuntu 22.04 STIG Versão 2 Versão 3
- Ubuntu 24.04 STIG Versão 1 Versão 1

Alterações no quarto trimestre de 2024 - 12/10/2024:

Atualizou as seguintes versões do STIG, aplicou o STIGS para a versão do quarto trimestre de 2024 e adicionou informações sobre dois novos parâmetros de entrada para os componentes Linux:

STIG-Build-Linux-Low versão 2024.4.x

- RHEL 7 STIG Versão 3 Versão 15
- RHEL 8 STIG Versão 2 Versão 1
- RHEL 9 STIG Versão 2 Versão 2
- Ubuntu 18.04 STIG versão 2 versão 15
- Ubuntu 20.04 STIG Versão 2 Versão 1
- Ubuntu 22.04 STIG Versão 2 Versão 2

STIG-Build-Linux-Medium versão 2024.4.x

- RHEL 7 STIG Versão 3 Versão 15
- RHEL 8 STIG Versão 2 Versão 1
- RHEL 9 STIG Versão 2 Versão 2
- Ubuntu 18.04 STIG versão 2 versão 15
- Ubuntu 20.04 STIG Versão 2 Versão 1
- Ubuntu 22.04 STIG Versão 2 Versão 2

STIG-Build-Linux-High versão 2024.4.x

- RHEL 7 STIG Versão 3 Versão 15
- RHEL 8 STIG Versão 2 Versão 1
- RHEL 9 STIG Versão 2 Versão 2
- Ubuntu 18.04 STIG versão 2 versão 15
- Ubuntu 20.04 STIG Versão 2 Versão 1
- Ubuntu 22.04 STIG Versão 2 Versão 2

Alterações no terceiro trimestre de 2024 - 10/04/2024 (sem alterações):

Não houve alterações no componente Linux STIGS para a versão do terceiro trimestre de 2024.

Alterações no segundo trimestre de 2024 (10/5/2024):

Versões do STIG atualizadas e STIGS aplicados para o release do segundo trimestre de 2024. Compatibilidade com RHEL 9, CentOS Stream 9 e Ubuntu 22.04 adicionada, conforme a seguir:

STIG-Build-Linux-Low versão 2024.2.x

- RHEL 7 STIG Versão 3 Release 14
- RHEL 8 STIG Versão 1 Release 14
- RHEL 9 STIG Versão 1 Release 3
- Ubuntu 18.04 STIG Versão 2 Release 14
- Ubuntu 20.04 STIG Versão 1 Release 12
- Ubuntu 22.04 STIG Versão 1 Release 1

STIG-Build-Linux-Medium versão 2024.2.x

- RHEL 7 STIG Versão 3 Release 14
- RHEL 8 STIG Versão 1 Release 14
- RHEL 9 STIG Versão 1 Release 3
- Ubuntu 18.04 STIG Versão 2 Release 14
- Ubuntu 20.04 STIG Versão 1 Release 12
- Ubuntu 22.04 STIG Versão 1 Release 1

STIG-Build-Linux-High versão 2024.2.x

- RHEL 7 STIG Versão 3 Release 14
- RHEL 8 STIG Versão 1 Release 14
- RHEL 9 STIG Versão 1 Release 3
- Ubuntu 18.04 STIG Versão 2 Release 14
- Ubuntu 20.04 STIG Versão 1 Release 12
- Ubuntu 22.04 STIG Versão 1 Release 1

Alterações no primeiro trimestre de 2024 (6/2/2024):

Versões do STIG atualizadas e STIGS aplicados para o lançamento do primeiro trimestre de 2024 da seguinte forma:

STIG-Build-Linux-Low versão 2024.1.x

- RHEL 7 STIG Versão 3 Release 14
- RHEL 8 STIG Versão 1 Release 13
- Ubuntu 18.04 STIG Versão 2 Release 13
- Ubuntu 20.04 STIG Versão 1 Release 11

STIG-Build-Linux-Medium versão 2024.1.x

- RHEL 7 STIG Versão 3 Release 14
- RHEL 8 STIG Versão 1 Release 13
- Ubuntu 18.04 STIG Versão 2 Release 13
- Ubuntu 20.04 STIG Versão 1 Release 11

STIG-Build-Linux-High versão 2024.1.x

- RHEL 7 STIG Versão 3 Release 14
- RHEL 8 STIG Versão 1 Release 13
- Ubuntu 18.04 STIG Versão 2 Release 13
- Ubuntu 20.04 STIG Versão 1 Release 11

Alterações no quarto trimestre de 2023 (7/12/2023):

Versões do STIG atualizadas e STIGS aplicados para o release do quarto trimestre de 2023 da seguinte forma:

STIG-Build-Linux-Low versão 2023.4.x

- RHEL 7 STIG Versão 3 Release 13
- RHEL 8 STIG Versão 1 Release 12

- Ubuntu 18.04 STIG Versão 2 Release 12
- Ubuntu 20.04 STIG Versão 1 Release 10

STIG-Build-Linux-Medium versão 2023.4.x

- RHEL 7 STIG Versão 3 Release 13
- RHEL 8 STIG Versão 1 Release 12
- Ubuntu 18.04 STIG Versão 2 Release 12
- Ubuntu 20.04 STIG Versão 1 Release 10

STIG-Build-Linux-High versão 2023.4.x

- RHEL 7 STIG Versão 3 Release 13
- RHEL 8 STIG Versão 1 Release 12
- Ubuntu 18.04 STIG Versão 2 Release 12
- Ubuntu 20.04 STIG Versão 1 Release 10

Alterações no terceiro trimestre de 2023 - 04/10/2023:

Versões do STIG atualizadas e STIGS aplicados para o release do terceiro trimestre de 2023 da seguinte forma:

STIG-Build-Linux-Low versão 2023.3.x

- RHEL 7 STIG Versão 3 Release 12
- RHEL 8 STIG Versão 1 Release 11
- Ubuntu 18.04 STIG Versão 2 Release 11
- Ubuntu 20.04 STIG Versão 1 Release 9

STIG-Build-Linux-Medium versão 2023.3.x

- RHEL 7 STIG Versão 3 Release 12
- RHEL 8 STIG Versão 1 Release 11
- Ubuntu 18.04 STIG Versão 2 Release 11

- Ubuntu 20.04 STIG Versão 1 Release 9

STIG-Build-Linux-High versão 2023.3.x

- RHEL 7 STIG Versão 3 Release 12
- RHEL 8 STIG Versão 1 Release 11
- Ubuntu 18.04 STIG Versão 2 Release 11
- Ubuntu 20.04 STIG Versão 1 Release 9

Alterações no segundo trimestre de 2023 - 03/05/2023:

Versões do STIG atualizadas e STIGS aplicados para o release do segundo trimestre de 2023 da seguinte forma:

STIG-Build-Linux-Low versão 2023.2.x

- RHEL 7 STIG Versão 3 Release 11
- RHEL 8 STIG Versão 1 Release 10
- Ubuntu 18.04 STIG Versão 2 Release 11
- Ubuntu 20.04 STIG Versão 1 Release 8

STIG-Build-Linux-Medium versão 2023.2.x

- RHEL 7 STIG Versão 3 Release 11
- RHEL 8 STIG Versão 1 Release 10
- Ubuntu 18.04 STIG Versão 2 Release 11
- Ubuntu 20.04 STIG Versão 1 Release 8

STIG-Build-Linux-High versão 2023.2.x

- RHEL 7 STIG Versão 3 Release 11
- RHEL 8 STIG Versão 1 Release 10
- Ubuntu 18.04 STIG Versão 2 Release 11
- Ubuntu 20.04 STIG Versão 1 Release 8

Alterações no primeiro trimestre de 2023 - 27/03/2023:

Versões do STIG atualizadas e STIGS aplicados para o release do primeiro trimestre de 2023 da seguinte forma:

STIG-Build-Linux-Low versão 2023.1.x

- RHEL 7 STIG Versão 3 Release 10
- RHEL 8 STIG Versão 1 Release 9
- Ubuntu 18.04 STIG Versão 2 Release 10
- Ubuntu 20.04 STIG Versão 1 Release 7

STIG-Build-Linux-Medium versão 2023.1.x

- RHEL 7 STIG Versão 3 Release 10
- RHEL 8 STIG Versão 1 Release 9
- Ubuntu 18.04 STIG Versão 2 Release 10
- Ubuntu 20.04 STIG Versão 1 Release 7

STIG-Build-Linux-High versão 2023.1.x

- RHEL 7 STIG Versão 3 Release 10
- RHEL 8 STIG Versão 1 Release 9
- Ubuntu 18.04 STIG Versão 2 Release 10
- Ubuntu 20.04 STIG Versão 1 Release 7

Alterações no quarto trimestre de 2022 - 01/02/2023:

Versões do STIG atualizadas e STIGS aplicados para o release do quarto trimestre de 2022 da seguinte forma:

STIG-Build-Linux-Low versão 2022.4.x

- RHEL 7 STIG Versão 3 Release 9
- RHEL 8 STIG Versão 1 Release 8
- Ubuntu 18.04 STIG Versão 2 Release 9

- Ubuntu 20.04 STIG Versão 1 Release 6

STIG-Build-Linux-Medium versão 2022.4.x

- RHEL 7 STIG Versão 3 Release 9
- RHEL 8 STIG Versão 1 Release 8
- Ubuntu 18.04 STIG Versão 2 Release 9
- Ubuntu 20.04 STIG Versão 1 Release 6

STIG-Build-Linux-High versão 2022.4.x

- RHEL 7 STIG Versão 3 Release 9
- RHEL 8 STIG Versão 1 Release 8
- Ubuntu 18.04 STIG Versão 2 Release 9
- Ubuntu 20.04 STIG Versão 1 Release 6

Alterações no terceiro trimestre de 2022 - 30/09/2022 (sem alterações):

Não houve alterações no componente STIGS do Linux para o release do terceiro trimestre de 2022.

Alterações no segundo trimestre de 2022 - 02/08/2022:

Suporte do Ubuntu introduzido, versões do STIG atualizadas e STIGS aplicados para o release do segundo trimestre de 2022 da seguinte forma:

STIG-Build-Linux-Low versão 2022.2.x

- RHEL 7 STIG Versão 3 Release 7
- RHEL 8 STIG Versão 1 Release 6
- Ubuntu 18.04 STIG Versão 2 Release 6 (novo)
- Ubuntu 20.04 STIG Versão 1 Release 4 (novo)

STIG-Build-Linux-Medium versão 2022.2.x

- RHEL 7 STIG Versão 3 Release 7

- RHEL 8 STIG Versão 1 Release 6
- Ubuntu 18.04 STIG Versão 2 Release 6 (novo)
- Ubuntu 20.04 STIG Versão 1 Release 4 (novo)

STIG-Build-Linux-High versão 2022.2.x

- RHEL 7 STIG Versão 3 Release 7
- RHEL 8 STIG Versão 1 Release 6
- Ubuntu 18.04 STIG Versão 2 Release 6 (novo)
- Ubuntu 20.04 STIG Versão 1 Release 4 (novo)

Alterações no primeiro trimestre de 2022 - 26/04/2022:

Refatorado para incluir melhor suporte para contêineres. Combinou o AL2 script anterior com o RHEL 7. Versões do STIG atualizadas e STIGS aplicados para o release do primeiro trimestre de 2022 da seguinte forma:

STIG-Build-Linux-Low versão 3.6.x

- RHEL 7 STIG Versão 3 Release 6
- RHEL 8 STIG Versão 1 Release 5

STIG-Build-Linux-Medium versão 3.6.x

- RHEL 7 STIG Versão 3 Release 6
- RHEL 8 STIG Versão 1 Release 5

STIG-Build-Linux-High versão 3.6.x

- RHEL 7 STIG Versão 3 Release 6
- RHEL 8 STIG Versão 1 Release 5

Alterações no quarto trimestre de 2021 - 20/12/2021:

Versões do STIG atualizadas e STIGS aplicados para o release do quarto trimestre de 2021 da seguinte forma:

STIG-Build-Linux-Low versão 3.5.x

- RHEL 7 STIG Versão 3 Release 5
- RHEL 8 STIG Versão 1 Release 4

STIG-Build-Linux-Medium versão 3.5.x

- RHEL 7 STIG Versão 3 Release 5
- RHEL 8 STIG Versão 1 Release 4

STIG-Build-Linux-High versão 3.5.x

- RHEL 7 STIG Versão 3 Release 5
- RHEL 8 STIG Versão 1 Release 4

Alterações no terceiro trimestre de 2021 - 30/09/2021:

Versões do STIG atualizadas e STIGS aplicados para o release do terceiro trimestre de 2021 da seguinte forma:

STIG-Build-Linux-Low versão 3.4.x

- RHEL 7 STIG Versão 3 Release 4
- RHEL 8 STIG Versão 1 Release 3

STIG-Build-Linux-Medium versão 3.4.x

- RHEL 7 STIG Versão 3 Release 4
- RHEL 8 STIG Versão 1 Release 3

STIG-Build-Linux-High versão 3.4.x

- RHEL 7 STIG Versão 3 Release 4
- RHEL 8 STIG Versão 1 Release 3

Componente validador de conformidade do SCAP

O Protocolo de Automação de Conteúdo de Segurança (SCAP) é um conjunto de padrões que os profissionais de TI podem usar para identificar vulnerabilidades de segurança de aplicativos para fins de conformidade. O Verificador de Conformidade do SCAP (SCC) é uma ferramenta de verificação validada pelo SCAP e lançada pelo Naval Information Warfare Center (NIWC) Atlantic. Para obter mais informações, consulte Verificador de conformidade (SCC) do Protocolo de Automação de Conteúdo de Segurança (SCAP) no site da NIWC Atlantic.

Os `scap-compliance-checker-linux` componentes AWSTOE `scap-compliance-checker-windows` e baixam e instalam o scanner SCC nas instâncias de criação e teste do pipeline. Quando o scanner é executado, ele executa varreduras de configuração autenticadas usando os benchmarks DISA SCAP e fornece um relatório que inclui as seguintes informações. AWSTOE também grava as informações nos registros do seu aplicativo.

- Configurações STIG que são aplicadas à instância.
- Uma pontuação de conformidade geral para a instância.

Recomendamos que você execute a validação SCAP como etapa final do processo de compilação, para garantir que você relate resultados precisos da validação de conformidade.

> **Note**
>
> Você pode revisar os relatórios com uma das Ferramentas de Visualização do STIG. Essas ferramentas estão disponíveis on-line por meio do DoD Cyber Exchange.

As seções a seguir descrevem os benchmarks incluídos nos componentes de validação SCAP.

scap-compliance-checker-windows versão 2024.03.0

O `scap-compliance-checker-windows` componente é executado nas EC2 instâncias que o Image Builder cria para criar e testar a imagem. AWSTOE registra o relatório e a pontuação que o aplicativo SCC produz.

O componente executa as seguintes etapas do fluxo de trabalho:

1. Baixa e instala o aplicativo SCC.

2. Importa os benchmarks de conformidade.
3. Executa a validação usando o aplicativo SCC.
4. Salva o relatório de conformidade e a pontuação localmente no desktop da instância de compilação.
5. Registra a pontuação de conformidade do relatório local nos arquivos de log do AWSTOE aplicativo.

> **Note**
>
> AWSTOE atualmente oferece suporte à validação de conformidade com SCAP para Windows Server 2012 R2 MS, 2016, 2019 e 2022.

O componente verificador de conformidade SCAP para Windows inclui os seguintes referenciais:

Versão SCC: 5.10

Benchmarks do quarto trimestre de 2023:

- u_MS_Defender_Antivirus_V2R5_STIG_SCAP_1-2_Benchmark
- U_MS_ DotNet _Framework_4-0_V2R2_STIG_SCAP_1-2_Benchmark
- U_MS_ _V2R6_STIG_SCAP_1-2_Referência IE11
- U_MS_Windows_2012_e_2012_R2_DC_V3R5_STIG_SCAP_1-2_Benchmark
- U_MS_Windows_2012_e_2012_R2_MS_V3R5_STIG_SCAP_1-2_Benchmark
- u_MS_Windows_Defender_Firewall_V2R3_STIG_SCAP_1-2_Benchmark
- U_MS_Windows_Server_2016_V2R7_STIG_SCAP_1-2_Benchmark
- U_MS_Windows_Server_2019_V3R2_STIG_SCAP_1-2_Benchmark
- U_MS_Windows_Server_2022_V2R2_STIG_SCAP_1-2_Benchmark
- U_CAN_Ubuntu_20-04_LTS_V1R10_STIG_SCAP_1-2_Benchmark
- U_RHEL_7_V3R15_STIG_SCAP_1-3_Referência
- U_RHEL_8_V1R13_STIG_SCAP_1-3_Referência
- U_RHEL_9_V2R1_STIG_SCAP_1-3_Referência

scap-compliance-checker-linux versão 2021.04.0

O `scap-compliance-checker-linux` componente é executado nas EC2 instâncias que o Image Builder cria para criar e testar a imagem. AWSTOE registra o relatório e a pontuação que o aplicativo SCC produz.

O componente executa as seguintes etapas do fluxo de trabalho:

1. Baixa e instala o aplicativo SCC.
2. Importa os benchmarks de conformidade.
3. Executa a validação usando o aplicativo SCC.
4. Salva o relatório de conformidade e a pontuação localmente, no seguinte local na instância de compilação: `/opt/scc/SCCResults`.
5. Registra a pontuação de conformidade do relatório local nos arquivos de log do AWSTOE aplicativo.

> **Note**
>
> AWSTOE atualmente suporta validação de conformidade SCAP para RHEL 7/8 e Ubuntu 18.04/20.04. Atualmente, o aplicativo SCC é compatível com a arquitetura para validação x86.

O componente verificador de conformidade SCAP para Linux inclui os seguintes referenciais:

Versão SCC: 5.10

Benchmarks do quarto trimestre de 2023:

- U_CAN_Ubuntu_20-04_LTS_V1R10_STIG_SCAP_1-2_Benchmark
- U_RHEL_7_V3R15_STIG_SCAP_1-3_Referência
- U_RHEL_8_V1R13_STIG_SCAP_1-3_Referência
- U_RHEL_9_V2R1_STIG_SCAP_1-3_Referência
- u_MS_Defender_Antivirus_V2R5_STIG_SCAP_1-2_Benchmark
- U_MS_ DotNet _Framework_4-0_V2R2_STIG_SCAP_1-2_Benchmark
- U_MS_ _V2R6_STIG_SCAP_1-2_Referência IE11
- U_MS_Windows_2012_e_2012_R2_DC_V3R5_STIG_SCAP_1-2_Benchmark

- U_MS_Windows_2012_e_2012_R2_MS_V3R5_STIG_SCAP_1-2_Benchmark
- u_MS_Windows_Defender_Firewall_V2R3_STIG_SCAP_1-2_Benchmark
- U_MS_Windows_Server_2016_V2R7_STIG_SCAP_1-2_Benchmark
- U_MS_Windows_Server_2019_V3R2_STIG_SCAP_1-2_Benchmark
- U_MS_Windows_Server_2022_V2R2_STIG_SCAP_1-2_Benchmark

Histórico de versões do SCAP

A tabela a seguir descreve as alterações importantes feitas no ambiente e configurações SCAP descritos neste documento.

Alteração	Descrição	Data
Atualizações do SCAP do primeiro trimestre de 2025	• scap-compliance-checker-windows versão 2024.03.0 (versão SCC: 5.10) • scap-compliance-checker-windows versão 2025.01.0 (Versão SCC: 5.10)	11 de abril de 2025
Atualizações do SCAP do quarto trimestre de 2023	• scap-compliance-checker-windows versão 2023.04.0 (versão SCC: 5.8) • scap-compliance-checker-windows versão 2023.04.0 (versão SCC: 5.8)	20 de dezembro de 2021
Atualizações do SCAP do terceiro trimestre de 2023	• scap-compliance-checker-windows versão 2023.03.0 (Versão SCC: 5.7.2) • scap-compliance-checker-linux versão 2023.03.0 (Versão SCC: 5.7.2)	13 de novembro de 2023
		20 de dezembro de 2021

Alteração	Descrição	Data
Componentes SCAP adicionados	Os componentes SCAP a seguir foram introduzidos: • scap-compliance-checker-linuxVersão criada 2021.04.0 (versão SCC: 5.4.2) • scap-compliance-checker-linuxVersão criada 2021.04.0 (versão SCC: 5.4.2)	

Desenvolver componentes personalizados para imagens do Image Builder

Você pode criar componentes próprios para personalizar imagens do Image Builder de acordo com suas especificações exatas. Use as etapas a seguir para desenvolver componentes personalizados para sua imagem ou fórmulas de contêiner do Image Builder.

1. Se você quiser desenvolver seu documento de componente e validá-lo localmente, você pode instalar o aplicativo AWS Task Orchestrator and Executor(AWSTOE) e configurá-lo em sua máquina local. Para obter mais informações, consulte Configuração manual para desenvolver componentes personalizados com AWSTOE.
2. Crie um documento de componente que use a estrutura do documento de AWSTOE componente. Para obter mais informações sobre o framework de documentos, consulte Use a estrutura de documentos de AWSTOE componentes para componentes personalizados.
3. Especifique o documento de componente ao criar um componente personalizado. Para obter mais informações, consulte Criar um componente personalizado com o Image Builder.

Tópicos

- Criar um documento de componente YAML para componentes personalizados no Image Builder
- Criar um componente personalizado com o Image Builder

Criar um documento de componente YAML para componentes personalizados no Image Builder

Para compilar um componente, você deve fornecer um documento de componente de aplicação YAML ou JSON. O documento contém o código executado durante as fases e etapas que você define para tornar sua imagem personalizada.

Alguns dos exemplos nesta seção criam um componente de construção que chama o módulo de `UpdateOS` ação no aplicativo de gerenciamento de AWSTOE componentes. O módulo atualiza o sistema operacional. Para obter mais informações sobre o módulo de ação `UpdateOS`, consulte UpdateOS.

O exemplo do sistema operacional macOS usa o módulo de ação `ExecuteBash` para instalar e verificar o utilitário `wget`. O módulo de ação `UpdateOS` não é compatível com o macOS. Para obter mais informações sobre o módulo de ação `ExecuteBash`, consulte ExecuteBash. Para obter mais informações sobre as fases, as etapas e a sintaxe dos documentos de componente da aplicação AWSTOE , consulte Use documents in AWSTOE.

> **Note**
>
> O Image Builder determina o tipo de componente com base nas fases definidas no documento do componente, conforme a seguir:
>
> - Compilação: esse é o tipo de componente padrão. Qualquer coisa que não seja classificada como componente de teste é um componente de compilação. Esse tipo de componente é executado durante o estágio de compilação da imagem. Se esse componente de compilação tiver uma fase de `test` definida, essa fase será executada durante o estágio de teste.
> - Teste: para se qualificar como um componente de teste, o documento do componente deve incluir somente uma fase, denominada `test`. Para testes relacionados às configurações de componentes de compilação, recomendamos que você não use um componente de teste independente. Em vez disso, use a fase `test` no componente de compilação associado.
>
> Para obter mais informações sobre como o Image Builder usa estágios e fases para gerenciar o fluxo de trabalho de componentes em seu processo de compilação, consulte Usar componentes para personalizar sua imagem do Image Builder.

Para criar um componente do aplicativo YAML para um aplicativo de amostra, siga as etapas que correspondem à sua plataforma de sistema operacional de imagem.

Linux

Criar um arquivo de componente YAML

Use uma ferramenta de edição de arquivos para criar o documento do componente. Os exemplos de documentação usam um arquivo chamado *update-linux-os.yaml*, com o seguinte conteúdo:

```
# 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**
>
> Use uma ferramenta como esse validador YAML on-line ou uma extensão lint YAML em seu ambiente de código para verificar se seu YAML está bem formado.

Windows

Criar um arquivo de componente YAML

Use uma ferramenta de edição de arquivos para criar o documento do componente. Os exemplos de documentação usam um arquivo chamado *update-windows-os.yaml*, com o seguinte conteúdo:

```
# 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**
>
> Use uma ferramenta como esse validador YAML on-line ou uma extensão lint YAML em seu ambiente de código para verificar se seu YAML está bem formado.

macOS

Criar um arquivo de componente YAML

Use uma ferramenta de edição de arquivos para criar o documento do componente. Os exemplos de documentação usam um arquivo chamado *wget-macos.yaml*, com o seguinte conteúdo:

```
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**
>
> Use uma ferramenta como esse validador YAML on-line ou uma extensão lint YAML em seu ambiente de código para verificar se seu YAML está bem formado.

Criar um componente personalizado com o Image Builder

Depois de concluir o documento do componente, você pode usá-lo para criar um componente personalizado que suas fórmulas do Image Builder possam usar. Você pode criar um componente personalizado no console do Image Builder SDKs, na API ou na linha de comando. Para obter mais informações sobre como criar um componente personalizado com parâmetros de entrada e usá-lo em suas fórmulas, consulte Tutorial: criar um componente personalizado com parâmetros de entrada.

As seções a seguir mostram como criar componentes com o console ou a AWS CLI.

Conteúdo

- Criar um componente personalizado com o console
- Crie um componente personalizado a partir do AWS CLI
- Importe um script para criar um componente a partir do AWS CLI
- Gerenciamento automático de versões de compilação
- Usando referências de versão

Criar um componente personalizado com o console

Para criar um componente do AWSTOE aplicativo a partir do console do Image Builder, siga estas etapas:

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Selecione Componentes no painel de navegação. Em seguida, selecione Criar componente.
3. Na página Criar componente, em Detalhes do componente, insira o seguinte:

   a. Sistema operacional da imagem (OS). Especifique o sistema operacional com o qual o componente é compatível.

   b. Categoria de componente. No menu suspenso, selecione o tipo de componente de compilação ou teste que você está criando.

   c. Nome do componente. Insira um nome para o componente.

   d. Versão do componente. Insira o número da versão do componente.

   e. Descrição. Fornece uma descrição opcional para ajudá-lo a identificar o componente.

   f. Descrição da alteração. Fornece uma descrição opcional para ajudá-lo a entender as alterações feitas nessa versão do componente.
4. Na seção Documento de definição, a opção padrão é Definir conteúdo do documento. O documento do componente define as ações que o Image Builder executa nas instâncias de construção e teste para criar sua imagem.

   Na caixa Conteúdo, insira o conteúdo do documento do componente YAML. Para começar com um exemplo do Hello World para Linux, escolha a opção Usar exemplo. Para saber mais sobre como criar um documento do componente YAML ou copiar e colar o exemplo do UpdateOS dessa página, consulte. Criar um documento de componente YAML para componentes personalizados no Image Builder
5. Depois de inserir os detalhes do componente, selecione Criar componente.

> **Note**
>
> Para ver seu novo componente ao criar ou atualizar uma fórmula, aplique o filtro De minha propriedade à lista de componentes da compilação ou teste. O filtro está localizado na parte superior da lista de componentes, ao lado da caixa de pesquisa.

6. Para excluir um componente, a partir da página Componentes, marque a caixa de seleção ao lado do componente que você deseja excluir. No menu suspenso Ações, selecione Excluir componente.

Atualizar um componente

Para criar uma nova versão do componente, siga estas etapas:

1. Dependendo de onde você começa:
   - Na página da lista de Componentes: marque a caixa de seleção ao lado do nome do componente e selecione Criar nova versão no menu Ações.
   - Na página de detalhes do componente: escolha o botão Criar nova versão no canto superior direito do cabeçalho.
2. As informações do componente já estão preenchidas com os valores atuais quando a página Criar componente é exibida. Siga as etapas de criação de um componente para atualizar o componente. Isso garante que você insira uma versão semântica exclusiva na versão do Componente. Para saber mais sobre o versionamento semântico dos recursos do Image Builder, consulte Versionamento semântico no Image Builder.

Crie um componente personalizado a partir do AWS CLI

Nesta seção, você aprenderá a configurar e usar os comandos do Image Builder no AWS CLI para criar um componente de AWSTOE aplicativo, da seguinte forma.

- Carregue o documento do componente do YAML em um bucket do S3 que você possa referenciar a partir da linha de comandos.
- Crie o componente do AWSTOE aplicativo com o create-component comando.
- Liste as versões do componente com o comando list-components e um filtro de nome para ver quais versões já existem. Você pode usar a saída para determinar qual deve ser a próxima versão para atualizações.

Para criar um componente de AWSTOE aplicativo a partir de um documento YAML de entrada, siga as etapas que correspondem à sua plataforma de sistema operacional de imagem.

Linux

Armazene seu documento do componente do aplicativo no Amazon S3

Você pode usar um bucket do S3 como repositório para o documento de origem do componente do AWSTOE aplicativo. Para armazenar seu documento do componente, siga estas etapas:

- Carregue o documento no Amazon S3

  Se seu documento for menor que 64 KB, você pode ignorar esta etapa. Documentos com 64 KB ou mais devem ser armazenados no Amazon S3.

  ```
  aws s3 cp update-linux-os.yaml s3://amzn-s3-demo-destination-bucket/my-path/update-linux-os.yaml
  ```

Crie um componente a partir do documento do YAML

Para simplificar o create-component comando que você usa no AWS CLI, crie um arquivo JSON que contenha todos os parâmetros do componente que você deseja passar para o comando. Inclua a localização do documento *update-linux-os.yaml* que você criou anteriormente. O par de chave–valor `uri` contém a referência do arquivo.

> **Note**
>
> A convenção de nomenclatura para os valores de dados no arquivo JSON segue o padrão especificado para os parâmetros de solicitação de operação da API Image Builder. Para revisar os parâmetros da solicitação de comando da API, consulte o CreateComponentcomando na Referência da API EC2 Image Builder.
> Para fornecer os valores dos dados como parâmetros da linha de comando, consulte os nomes dos parâmetros especificados na AWS CLI Referência de comando.

1. Criar um arquivo JSON de entrada da CLI

   Use uma ferramenta de edição de arquivos para criar um arquivo chamado *create-update-linux-os-component.json*. Inclua o seguinte conteúdo:

   ```
   {
    "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. Criar o componente

   Use o comando a seguir para criar o componente, fazendo referência ao nome do arquivo JSON que você criou na etapa anterior:

   ```
   aws imagebuilder create-component --cli-input-json file://create-update-linux-
   os-component.json
   ```

   > **Note**
   >
   > - É necessário incluir a notação `file://` no início do caminho do arquivo JSON.
   > - O caminho para o arquivo JSON deve seguir a convenção apropriada para o sistema operacional de base no qual você está executando o comando. Por exemplo, o Windows usa a barra invertida (\) para se referir ao caminho do diretório, enquanto o Linux e o macOS usam a barra (/).

Windows

Armazene seu documento do componente do aplicativo no Amazon S3

Você pode usar um bucket do S3 como repositório para o documento de origem do componente do AWSTOE aplicativo. Para armazenar seu documento do componente, siga estas etapas:

- Carregue o documento no Amazon S3

  Se seu documento for menor que 64 KB, você pode ignorar esta etapa. Documentos com 64 KB ou mais devem ser armazenados no Amazon S3.

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

Crie um componente a partir do documento do YAML

Para simplificar o create-component comando que você usa no AWS CLI, crie um arquivo JSON que contenha todos os parâmetros do componente que você deseja passar para o comando. Inclua a localização do documento *update-windows-os.yaml* que você criou anteriormente. O par de chave–valor `uri` contém a referência do arquivo.

> **Note**
>
> A convenção de nomenclatura para os valores de dados no arquivo JSON segue o padrão especificado para os parâmetros de solicitação de operação da API Image Builder. Para revisar os parâmetros da solicitação de comando da API, consulte o CreateComponentcomando na Referência da API EC2 Image Builder.
> Para fornecer os valores dos dados como parâmetros da linha de comando, consulte os nomes dos parâmetros especificados na AWS CLI Referência de comando..

1. Criar um arquivo JSON de entrada da CLI

   Use uma ferramenta de edição de arquivos para criar um arquivo chamado *create-update-windows-os-component.json*. Inclua o seguinte conteúdo:

```
{
 "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**
>
> - É necessário incluir a notação `file://` no início do caminho do arquivo JSON.
> - O caminho para o arquivo JSON deve seguir a convenção apropriada para o sistema operacional de base no qual você está executando o comando. Por exemplo, o Windows usa a barra invertida (\) para se referir ao caminho do diretório, enquanto o Linux e o macOS usam a barra (/).

2. Criar o componente

   Use o comando a seguir para criar o componente, fazendo referência ao nome do arquivo JSON que você criou na etapa anterior:

   ```
   aws imagebuilder create-component --cli-input-json file://create-update-windows-os-component.json
   ```

   > **Note**
   >
   > - É necessário incluir a notação `file://` no início do caminho do arquivo JSON.
   > - O caminho para o arquivo JSON deve seguir a convenção apropriada para o sistema operacional de base no qual você está executando o comando. Por exemplo, o Windows usa a barra invertida (\) para se referir ao caminho do diretório, enquanto o Linux e o macOS usam a barra (/).

macOS

Armazene seu documento do componente do aplicativo no Amazon S3

Você pode usar um bucket do S3 como repositório para o documento de origem do componente do AWSTOE aplicativo. Para armazenar seu documento do componente, siga estas etapas:

- Carregue o documento no Amazon S3

  Se seu documento for menor que 64 KB, você pode ignorar esta etapa. Documentos com 64 KB ou mais devem ser armazenados no Amazon S3.

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

Crie um componente a partir do documento do YAML

Para simplificar o create-component comando que você usa no AWS CLI, crie um arquivo JSON que contenha todos os parâmetros do componente que você deseja passar para o comando. Inclua a localização do documento *wget-macos.yaml* que você criou anteriormente. O par de chave–valor `uri` contém a referência do arquivo.

> **Note**
>
> A convenção de nomenclatura para os valores de dados no arquivo JSON segue o padrão especificado para os parâmetros de solicitação de operação da API Image Builder. Para revisar os parâmetros da solicitação de comando da API, consulte o CreateComponentcomando na Referência da API EC2 Image Builder.
> Para fornecer os valores dos dados como parâmetros da linha de comando, consulte os nomes dos parâmetros especificados na AWS CLI Referência de comando.

1. Criar um arquivo JSON de entrada da CLI

   Use uma ferramenta de edição de arquivos para criar um arquivo chamado *install-wget-macos-component.json*. Inclua o seguinte conteúdo:

   ```
   {
    "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. Criar o componente

   Use o comando a seguir para criar o componente, fazendo referência ao nome do arquivo JSON que você criou na etapa anterior:

   ```
   aws imagebuilder create-component --cli-input-json file://install-wget-macos-component.json
   ```

   > **Note**
   >
   > - É necessário incluir a notação `file://` no início do caminho do arquivo JSON.
   > - O caminho para o arquivo JSON deve seguir a convenção apropriada para o sistema operacional de base no qual você está executando o comando. Por exemplo, o Windows usa a barra invertida (\) para se referir ao caminho do diretório, enquanto o Linux e o macOS usam a barra (/).

AWSTOE controle de versão de componentes para atualizações do AWS CLI

AWSTOE os nomes e versões dos componentes são incorporados no Amazon Resource Name (ARN) do componente, após o prefixo do componente. Cada nova versão de um componente tem seu próprio ARN exclusivo. As etapas para criar uma nova versão são exatamente iguais às etapas para criar um novo componente, desde que a versão semântica seja exclusiva para aquele nome de componente. Para saber mais sobre o versionamento semântico dos recursos do Image Builder, consulte Versionamento semântico no Image Builder.

Para garantir que você atribua a próxima versão lógica, primeiro obtenha uma lista das versões existentes do componente que você deseja alterar. Use o list-components comando com o AWS CLI, e filtre pelo nome.

Neste exemplo, você filtra pelo nome do componente que você criou nos exemplos anteriores do Linux. Para listar o componente que você criou, use o valor do parâmetro `name` do arquivo JSON que você usou no comando create-component.

```
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"
        }
    ]
}
```

Com base nos resultados, você pode determinar qual deve ser a próxima versão.

Importe um script para criar um componente a partir do AWS CLI

Em alguns cenários, talvez seja mais fácil começar com um script preexistente. Para este cenário, você pode usar o exemplo a seguir.

Este exemplo pressupõe que você tem um arquivo chamado *`import-component.json`* (conforme mostrado). Observe que o arquivo faz referência direta a um PowerShell script chamado `AdminConfig.ps1` that já foi enviado para*`amzn-s3-demo-source-bucket`*. No momento, o `SHELL` é compatível com o componente `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"
}
```

Para criar o componente usando um script importado, execute o comando a seguir.

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

Gerenciamento automático de versões de compilação

Quando você cria um componente com o mesmo nome e versão semântica de um componente existente, o Image Builder incrementa automaticamente a versão de compilação (por exemplo, de `/1` para `/2/3`, para e assim por diante). Isso permite que você faça atualizações iterativas em seus componentes sem gerenciar manualmente os números de versão, o que é especialmente útil em CI/CD pipelines e infrastructure-as-code implantações. Se o conteúdo do componente for idêntico à versão de compilação anterior, o Image Builder retornará `ResourceAlreadyExistsException` para evitar que componentes duplicados consumam sua cota de serviço.

Usando referências de versão

Quando você cria ou recupera um componente, o Image Builder fornece automaticamente padrões de versão pré-construídos ARNs com caracteres curinga no objeto. latestVersionReferences Essas referências facilitam o uso das versões mais recentes de seus componentes em receitas e pipelines sem análise ARNs manual.

Escolhendo a referência de versão correta

- latestVersionArn (x.x.x) - Sempre use a versão mais recente absoluta do componente.
- atestMajorVersionArn (1.x.x) - Use as versões secundárias e de patch mais recentes em uma versão principal.

- latestMinorVersionArn (1.2.x) - Use somente a versão mais recente do patch.
- latestPatchVersionArn (1.2.3) - Faça referência a uma versão semântica específica, mas obtenha a versão de compilação mais recente.

> **Note**
>
> Para evitar cobranças inesperadas, certifique-se de limpar os recursos e pipelines que você criou com base nos exemplos deste guia. Para obter mais informações sobre exclusão de recursos no Image Builder, consulte Excluir recursos desatualizados ou não utilizados do Image Builder.

Como o Image Builder usa o AWS Task Orchestrator and Executor aplicativo para gerenciar componentes

EC2 O Image Builder usa o aplicativo AWS Task Orchestrator and Executor (AWSTOE) para orquestrar fluxos de trabalho complexos, modificar configurações do sistema e testar suas imagens sem a necessidade de scripts ou códigos adicionais de DevOps. Essa aplicação gerencia e executa componentes que implementam seu esquema de documento declarativo.

AWSTOE é um aplicativo independente que o Image Builder instala em suas instâncias de compilação e teste quando você cria uma imagem. Você também pode instalá-lo manualmente nas EC2 instâncias para criar seus próprios componentes personalizados. Ele não requer nenhuma configuração adicional e também pode ser executado no ambiente on-premises.

Conteúdo

- AWSTOE downloads
- Regiões do compatíveis
- AWSTOE referência de comando
- Configuração manual para desenvolver componentes personalizados com AWSTOE
- Use a estrutura de documentos de AWSTOE componentes para componentes personalizados
- Módulos de ação compatíveis com o gerenciador de componentes do AWSTOE
- Configurar a entrada para o comando de AWSTOE execução

AWSTOE downloads

Para instalar AWSTOE, escolha o link de download para sua arquitetura e plataforma. Se você se conectar a um VPC endpoint para seu serviço (Image Builder, por exemplo), ele deverá ter uma política de endpoint personalizada anexada que inclua acesso ao bucket do S3 para downloads. AWSTOE Caso contrário, suas instâncias de compilação e teste não poderão baixar o script bootstrap (`bootstrap.sh`) e instalar o AWSTOE aplicativo. Para obter mais informações, consulte Criar uma política de endpoint da VPC o Image Builder.

> **Important**
>
> AWS está eliminando gradualmente o suporte para as versões 1.0 e 1.1 do TLS. Para acessar o bucket do S3 para AWSTOE downloads, seu software cliente deve usar o TLS versão 1.2 ou posterior. Para obter mais informações, consulte esta publicação do blog de Segurança da AWS.

Arquitetura	Plataforma	Link para fazer download	Exemplo
386	AL 2 e 2023 RHEL 7, 8 e 9 Ubuntu 16.04, 18.04, 20.04, 22.04 e 24.04 CentOS 7 e 8 SUSE 12 e 15	`https://awstoe- <region>.s3.com/latest/linux/386/awstoe`	https://awstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/linux/386/awstoe
AMD64	AL 2 e 2023 RHEL 7, 8 e 9 Ubuntu 16.04, 18.04, 20.04, 22.04 e 24.04 CentOS 7 e 8	`https://awstoe- <region>.s3.com/latest/linux/amd64/awstoe`	https://awstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/linux/amd64/awstoe

Arquitetura	Plataforma	Link para fazer download	Exemplo
	CentOS Stream 8 SUSE 12 e 15		
AMD64	macOS 10.14.x (Mojave), 10.15.x (Catalina), 11.x (Big Sur), 12.x (Monterey)	`https://awstoe-region.s3.amazonaws.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 e 2022	`https://awstoe-<region>.s3.amazonaws.com/latest/windows/amd64/awstoe.exe`	https://awstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/windows/amd64/awstoe.exe
ARM64	AL 2 e 2023 RHEL 7, 8 e 9 Ubuntu 16.04, 18.04, 20.04, 22.04 e 24.04 CentOS 7 e 8 CentOS Stream 8 SUSE 12 e 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

Regiões do compatíveis

AWSTOE é suportado como um aplicativo independente nas seguintes regiões.

Região da AWS nome	Região da AWS
Leste dos EUA (Ohio)	us-east-2

Região da AWS nome	Região da AWS
Leste dos EUA (Norte da Virgínia)	us-east-1
AWS GovCloud (Leste dos EUA)	us-gov-east-1
AWS GovCloud (Oeste dos EUA)	us-gov-west-1
Oeste dos EUA (N. da Califórnia)	us-west-1
Oeste dos EUA (Oregon)	us-west-2
África (Cidade do Cabo)	af-south-1
Ásia-Pacífico (Hong Kong)	ap-east-1
Ásia-Pacífico (Osaka)	ap-northeast-3
Ásia-Pacífico (Seul)	ap-northeast-2
Ásia Pacífico (Mumbai)	ap-south-1
Ásia-Pacífico (Hyderabad)	ap-south-2
Ásia-Pacífico (Singapura)	ap-southeast-1
Ásia-Pacífico (Sydney)	ap-southeast-2
Ásia-Pacífico (Jacarta)	ap-southeast-3
Ásia-Pacífico (Tóquio)	ap-northeast-1
Canadá (Central)	ca-central-1
Europa (Frankfurt)	eu-central-1
Europa (Zurique)	eu-central-2
Europa (Estocolmo)	eu-north-1
Europa (Milão)	eu-south-1

Região da AWS nome	Região da AWS
Europa (Espanha)	eu-south-2
Europa (Irlanda)	eu-west-1
Europa (Londres)	eu-west-2
Europa (Paris)	eu-west-3
Israel (Tel Aviv)	il-central-1
Oriente Médio (Emirados Árabes Unidos)	me-central-1
Oriente Médio (Bahrein)	me-south-1
América do Sul (São Paulo)	sa-east-1
China (Pequim)	cn-north-1
China (Ningxia)	cn-northwest-1

AWSTOE referência de comando

AWSTOE é um aplicativo de gerenciamento de componentes de linha de comando executado em EC2 instâncias da Amazon. Quando o Image Builder inicia uma instância de EC2 compilação ou teste, ele é instalado AWSTOE na instância. Em seguida, ele executa AWSTOE comandos no AWS CLI para instalar ou validar os componentes especificados na imagem ou na receita do contêiner.

> **Note**
>
> Alguns módulos de AWSTOE ação exigem permissões elevadas para serem executados em um servidor Linux. Para usar permissões elevadas, prefixe a sintaxe do comando com sudo ou execute o comando sudo su uma vez ao fazer login antes de executar os comandos vinculados abaixo. Para obter mais informações sobre módulos de AWSTOE ação, consulteMódulos de ação compatíveis com o gerenciador de componentes do AWSTOE.

run

Use o run comando para executar os scripts do documento YAML para um ou mais documentos componentes.

validar

Use o validate comando para validar a sintaxe do documento YAML para um ou mais documentos componentes.

comando awstoe run

Este comando executa os scripts do documento do componente YAML na ordem em que são incluídos no arquivo de configuração especificado pelo parâmetro `--config` ou na lista de documentos do componente especificada pelo parâmetro `--documents`.

> **Note**
>
> Você deve especificar exatamente um dos seguintes parâmetros, nunca os dois:
> --config
> --documents

Sintaxe

```
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]
```

Parâmetros e opções

Parâmetros

--config ***./config-example.json***

Forma abreviada: -c *./config-example.json*

O arquivo de configuração (condicional). Esse parâmetro contém a localização do arquivo JSON que contém as definições de configuração dos componentes que esse comando está executando. Se você especificar as configurações de run comando em um arquivo de configuração, não deverá especificar o `--documents` parâmetro. Para obter mais informações sobre a configuração de entrada, consulte Configurar a entrada para o comando de AWSTOE execução.

Localizações válidas incluem:

- Um caminho de arquivo local (*`./config-example.json`*)
- Uma URI do S3 (`s3://`*`bucket/key`*)

--cw-ignore-failures

Forma abreviada: N/A

Ignore as falhas de CloudWatch registro dos registros.

--cw-log-group

Forma abreviada: N/A

O `LogGroup` nome dos CloudWatch Logs.

--cw-log-region

Forma abreviada: N/A

A AWS região que se aplica aos CloudWatch registros.

--cw-log-stream

Forma abreviada: N/A

O `LogStream` nome dos CloudWatch registros, que direciona para AWSTOE onde transmitir o `console.log` arquivo.

--document-s3-proprietário do bucket

Forma abreviada: N/A

O ID da conta do proprietário do bucket para documentos baseados em URI do S3.

--documentos ***`./doc-1.yaml,./doc-n.yaml`***

Forma abreviada: -d *`./doc-1.yaml`*,*`./doc-n`*

Os documentos do componente (condicionais). Esse parâmetro contém uma lista separada por vírgulas dos locais dos arquivos para execução dos documentos do componente YAML. Se você especificar documentos YAML para o run comando usando o `--documents` parâmetro, não deverá especificar o `--config` parâmetro.

Localizações válidas incluem:

- caminhos de arquivo locais (*`./component-doc-example.yaml`*).
- S3 URIs (`s3://`*`bucket/key`*).
- Versão de compilação do componente Image Builder ARNs (arn:aws:imagebuilder:us-west-/ /2021.12.02/1). *`2:123456789012 my-example-component`*

> **Note**
>
> Não há espaços entre os itens na lista, somente vírgulas.

--execution-id

Forma abreviada: -i

Esse é o ID exclusivo que se aplica à execução do comando run atual. Este ID é incluído nos nomes dos arquivos de saída e log, para identificar esses arquivos de forma exclusiva e vinculá-los à execução atual do comando. Se essa configuração for omitida, AWSTOE gera um GUID.

--log-directory

Forma abreviada: -l

O diretório de destino onde AWSTOE armazena todos os arquivos de log da execução desse comando. Este arquivo está localizado no diretório a seguir, por padrão: `TOE_<DATETIME>_<EXECUTIONID>`. Se você não especificar o diretório de log, AWSTOE usa o diretório de trabalho atual (`.`).

--log-s3-nome do bucket

Forma abreviada: -b

Se os registros do componente estiverem armazenados no Amazon S3 (recomendado), AWSTOE carrega os registros do aplicativo do componente para o bucket do S3 nomeado nesse parâmetro.

--log-s3-proprietário do bucket

Forma abreviada: N/A

Se os registros do componente estiverem armazenados no Amazon S3 (recomendado), esse é o ID da conta do proprietário do bucket em que AWSTOE grava os arquivos de log.

--log-s3-key-prefix

Forma abreviada: -k

Se os registros do componente estiverem armazenados no Amazon S3 (recomendado), esse é o prefixo da chave de objeto do S3 para a localização do log no bucket.

--parâmetros ***name1*** =***value1***, ***name2*** =***value2***...

Forma abreviada: N/A

Os parâmetros são variáveis mutáveis definidas no documento do componente, com configurações que o aplicativo de chamada pode fornecer em tempo de execução.

--phases

Forma abreviada: -p

Uma lista separada por vírgulas que especifica quais fases devem ser executadas a partir dos documentos do componente YAML. Se um documento do componente incluir fases adicionais, elas não serão executadas.

--diretório estadual

Forma abreviada: -s

O caminho do arquivo em que os arquivos de rastreamento de estado são armazenados.

--version

Forma abreviada: -v

Especifica a versão do aplicativo do componente.

Opções

--help

Forma abreviada: -h

Exibe um manual de ajuda para usar as opções do aplicativo de gerenciamento de componentes.

--trace

Forma abreviada: -t

Ativa o registro detalhado no console do.

comando awstoe validate

Ao executar esse comando, ele valida a sintaxe do documento YAML para cada um dos documentos de componente especificados pelo parâmetro `--documents`.

Sintaxe

```
awstoe validate [--document-s3-bucket-owner <owner>]
      --documents <file path,file path,...> [--help] [--trace]
```

Parâmetros e opções

Parâmetros

--document-s3-proprietário do bucket

Forma abreviada: N/A

ID da conta de origem dos documentos baseados em URI do S3 fornecidos.

--documentos ***./doc-1.yaml,./doc-n.yaml***

Forma abreviada: -d *./doc-1.yaml*,*./doc-n*

Os documentos do componente (obrigatório). Esse parâmetro contém uma lista separada por vírgulas dos locais dos arquivos para execução dos documentos do componente YAML. Localizações válidas incluem:

- caminhos de arquivo locais (*./component-doc-example.yaml*)
- S3 URIs () `s3://`*bucket/key*
- Versão de compilação do componente Image Builder ARNs (arn:aws:imagebuilder:us-west-/2021.12.02/1) *2:123456789012 my-example-component*

> **Note**
>
> Não há espaços entre os itens na lista, somente vírgulas.

Opções

--help

Forma abreviada: -h

Exibe um manual de ajuda para usar as opções do aplicativo de gerenciamento de componentes.

--trace

Forma abreviada: -t

Ativa o registro detalhado no console do.

Configuração manual para desenvolver componentes personalizados com AWSTOE

O aplicativo AWS Task Orchestrator and Executor (AWSTOE) é um aplicativo independente que cria, valida e executa comandos dentro de uma estrutura de definição de componentes. AWS os serviços podem ser usados AWSTOE para orquestrar fluxos de trabalho, instalar software, modificar configurações do sistema e testar compilações de imagens.

Siga estas etapas para instalar manualmente o AWSTOE aplicativo e usá-lo como um aplicativo independente para desenvolver componentes personalizados. O Image Builder cuida dessas etapas para você, se você usar o console ou os AWS CLI comandos do Image Builder para criar componentes personalizados. Para obter mais informações, consulte Criar componentes personalizados com o Image Builder.

Verifique a assinatura do download da AWSTOE instalação

Esta seção descreve o processo recomendado para verificar a validade do download da instalação em sistemas operacionais baseados AWSTOE em Linux, macOS e Windows.

Tópicos

- Verificar a assinatura do download de instalação do AWSTOE no Linux ou no macOS

- Verifique a assinatura do download da instalação do AWSTOE no Windows

Verificar a assinatura do download de instalação do AWSTOE no Linux ou no macOS

Este tópico descreve o processo recomendado para verificar a validade do download da instalação para sistemas operacionais baseados AWSTOE em Linux ou macOS.

Sempre que baixar um aplicativo da Internet, recomendamos que você autentique a identidade do publicador do software. Além disso, verifique se o aplicativo não foi alterado ou corrompido desde que foi publicado. Isso protege você contra a instalação de uma versão do aplicativo que contenha um vírus ou outro código mal-intencionado.

Se depois de executar as etapas neste tópico, você determinar que o software do aplicativo AWSTOE está alterado ou corrompido, não execute o arquivo de instalação. Em vez disso, entre em contato Suporte para obter mais informações sobre suas opções de suporte, consulte Suporte.

AWSTOE arquivos para sistemas operacionais baseados em Linux e macOS são assinados usando GnuPG uma implementação de código aberto do padrão Pretty Good Privacy (OpenPGP) para assinaturas digitais seguras. GnuPG(também conhecido comoGPG) fornece verificação de autenticação e integridade por meio de uma assinatura digital. A Amazon EC2 publica uma chave pública e assinaturas que você pode usar para verificar as ferramentas Amazon CLI EC2 baixadas. Para obter mais informações sobre PGP e GnuPG (GPG), consulte http://www.gnupg.org.

A primeira etapa é estabelecer confiança com o fornecedor do software. Faça download da chave pública do fornecedor do software, verifique se o proprietário da chave pública é quem afirma ser e, em seguida, adicione a chave pública ao seu keyring. O keyring é um conjunto de chaves públicas conhecidas. Após estabelecer a autenticidade da chave pública, você pode usá-la para verificar a assinatura do aplicativo.

Tópicos

- Como instalar as ferramentas do GPG
- Como autenticar e importar a chave pública
- Verificar a assinatura do pacote

Como instalar as ferramentas do GPG

Se o seu sistema operacional for Linux, Unix ou macOS, as ferramentas do GPG provavelmente já estarão instaladas. Para testar se as ferramentas estão instaladas no sistema, digite gpg em um

prompt de comando. Se as ferramentas do GPG estiverem instaladas, um prompt de comando do GPG será exibido. Se as ferramentas do GPG não estiverem instaladas, uma mensagem de erro será exibida informando que o comando não pode ser encontrado. Você pode instalar o pacote GnuPG a partir de um repositório.

Para instalar as ferramentas do GPG, selecione o sistema operacional que corresponde à sua instância.

Debian-based Linux

Em um terminal, execute o comando a seguir:

```
apt-get install gnupg
```

Red Hat–based Linux

Em um terminal, execute o comando a seguir:

```
yum install gnupg
```

macOS

Em um terminal, execute o comando a seguir:

```
brew install gnupg
```

Como autenticar e importar a chave pública

A próxima etapa do processo é autenticar a chave AWSTOE pública e adicioná-la como uma chave confiável em seu GPG chaveiro.

Para autenticar e importar a chave AWSTOE pública

1. Obtenha uma cópia de nossa compilação de chave pública do GPG de uma das seguintes maneiras:

   - Baixe a chave em

     https://awstoe - **<region>** .s3. **<region>**.amazonaws. com/assets/awstoe.gpg. Por exemplo, .https://awstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/assets/awstoe.gpg

- Copie a chave do texto abaixo e cole-a em um arquivo chamado `awstoe.gpg`. Certifique-se de incluir tudo o que está a seguir:

```
-----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. Em um prompt de comando no diretório em que você salvou awstoe.gpg, use o comando a seguir para importar a chave AWSTOE pública para o seu chaveiro.

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

O comando retorna resultados semelhantes a:

```
gpg: key F5AEBC52: public key "AWSTOE <awstoe@amazon.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
```

Anote o valor da chave, ele será necessário na próxima etapa. No exemplo anterior, o valor da chave é `F5AEBC52`.

3. Verifique a impressão digital, executando o comando a seguir, substituindo chave-valor pelo valor da etapa anterior:

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

Esse comando retorna resultados semelhantes a:

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

Além disso, a cadeia de caracteres da impressão digital deve ser idêntica a `F6DD E01C 869F D639 15E5 5742 DEBD C156 F5AE BC52` mostrado acima no exemplo anterior. Compare a impressão digital da chave retornada à publicada nesta página. Elas devem corresponder. Se eles não corresponderem, não instale o script de AWSTOE instalação e entre em contato Suporte.

Verificar a assinatura do pacote

Depois de instalar as ferramentas `GPG`, autentique e importe a chave pública AWSTOE e verifique se a chave pública é confiável, você estará pronto para verificar a assinatura do script de instalação.

Para verificar a assinatura do script de instalação

1. Em um prompt de comando, execute o comando a seguir para baixar o binário do aplicativo:

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

   Por exemplo:

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

   Os valores compatíveis para **`architecture`** podem ser `amd64`, `386` e `arm64`.

2. Em um prompt de comando, execute o comando a seguir para baixar o arquivo de assinatura para o binário do aplicativo correspondente a partir do mesmo caminho de prefixo de chave do S3:

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

Por exemplo:

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

Os valores compatíveis para **`architecture`** podem ser `amd64`, `386` e `arm64`.

3. Verifique a assinatura executando o comando a seguir em um prompt de comando no diretório em que você salvou `awstoe.sig` e no arquivo AWSTOE de instalação. Ambos os arquivos devem estar presentes.

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

A saída deve parecer com algo semelhante ao seguinte:

```
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
```

Se a saída contém a frase `Good signature from "AWSTOE <awstoe@amazon.com>"`, isso significa que a assinatura foi confirmada com êxito e você pode dar continuidade à execução do script de instalação do AWSTOE .

Se a saída inclui a frase `BAD signature`, verifique se você executou o procedimento corretamente. Se você continuar a receber essa resposta, não execute o arquivo de instalação que baixou anteriormente e entre em contato com o Suporte.

Veja a seguir os detalhes sobre as advertências que talvez sejam exibidas:

- AVISO: esta chave não está certificada com uma assinatura confiável! Não há indicação de que a assinatura pertença ao proprietário. Idealmente, você visitaria um AWS escritório e receberia a chave pessoalmente. No entanto, é mais provável você baixá-la de um site. Nesse caso, o site é um AWS site.
- gpg: em última análise, nenhuma chave confiável encontrada. Isso significa que a chave específica não é "essencialmente confiável" por você ou por outras pessoas que você confia.

Para obter mais informações, consulte http://www.gnupg.org.

Verifique a assinatura do download da instalação do AWSTOE no Windows

Este tópico descreve o processo recomendado para verificar a validade do arquivo de instalação do AWS Task Orchestrator and Executor aplicativo em sistemas operacionais baseados em Windows.

Sempre que baixar um aplicativo da Internet, recomendamos que você autentique a identidade do fornecedor do software e verifique se o aplicativo não foi alterado ou corrompido desde que foi publicado. Isso protege você contra a instalação de uma versão do aplicativo que contenha um vírus ou outro código mal-intencionado.

Se depois de executar as etapas neste tópico, você determinar que o software do aplicativo AWSTOE está alterado ou corrompido, não execute o arquivo de instalação. Em vez disso, entre em contato Suporte.

Para verificar a validade do binário awstoe baixado em sistemas operacionais Windows, você deve certificar-se de que o thumbprint do certificado do assinante do Amazon Services LLC seja igual a este valor:

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

> **Note**
>
> Durante a janela de lançamento de um novo binário, seu certificado de signatário pode não corresponder à nova impressão digital. Se seu certificado de signatário não corresponder à nova impressão digital, verifique se o valor da impressão digital é:
> F8 83 11 EE F0 4A A2 91 E3 79 21 BA 6B FC AF F8 19 92 12 D7

Para verificar esse valor, execute o procedimento a seguir:

1. Clique com o botão direito do mouse no `awstoe.exe` baixado e abra a janela Propriedades.
2. Escolha a guia Assinaturas digitais.
3. Em Lista de assinaturas, escolha Amazon Web Services LLC e selecione Detalhes.
4. Escolha a guia Geral, se ainda não estiver selecionada, e escolha Visualizar certificado.
5. Selecione a guia Detalhes e Todos na lista suspensa Exibir, se ela ainda não estiver selecionada.

6. Role para baixo até ver o campo Impressão digital e, em seguida, escolha Impressão digital. Isso exibe todo o valor da impressão digital na janela inferior.

   - Se o valor da impressão digital na janela inferior for idêntico a este valor:

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

     então seu AWSTOE binário baixado é autêntico e pode ser instalado com segurança.
   - Se o valor da impressão digital na janela de detalhes inferior não for idêntico ao valor anterior, não execute `awstoe.exe`.

Etapas básicas

- Etapa 1: instalar AWSTOE
- Etapa 2: definir AWS credenciais
- Etapa 3: Desenvolver documentos de componentes localmente
- Etapa 4: Validar componentes AWSTOE
- Etapa 5: Executar AWSTOE componentes

Etapa 1: instalar AWSTOE

Para desenvolver componentes localmente, baixe e instale o AWSTOE aplicativo.

1. Baixe o AWSTOE aplicativo

   Para instalar AWSTOE, escolha o link de download apropriado para sua arquitetura e plataforma. Para obter a lista completa de links para download de aplicativos, consulte AWSTOE downloads

   > **Important**
   >
   > AWS está eliminando gradualmente o suporte para as versões 1.0 e 1.1 do TLS. Para acessar o bucket do S3 para AWSTOE downloads, seu software cliente deve usar o TLS versão 1.2 ou posterior. Para obter mais informações, consulte esta AWS publicação do blog de Segurança.

2. Verifique a assinatura

   As etapas para verificar seu download dependem da plataforma do servidor em que você executa o AWSTOE aplicativo depois de instalá-lo. Para verificar seu download em um servidor Linux, consulteVerificar a assinatura no Linux ou no macOS. Para verificar seu download em um servidor Windows, consulteVerifique a assinatura no Windows.

> **Note**
>
> AWSTOE é invocado diretamente de seu local de download. Não há necessidade de uma etapa de instalação separada. Isso também significa que AWSTOE pode fazer alterações no ambiente local.
> Para garantir que você isole as alterações durante o desenvolvimento do componente, recomendamos que você use uma EC2 instância para desenvolver e testar AWSTOE componentes.

Etapa 2: definir AWS credenciais

AWSTOE requer AWS credenciais para se conectar a outros Serviços da AWS, como Amazon S3 e CloudWatch Amazon, ao executar tarefas, como:

- Baixar AWSTOE documentos de um caminho do Amazon S3 fornecido pelo usuário.
- Módulos de execução `S3Download` ou ação `S3Upload`.
- Streaming de registros para CloudWatch, quando ativado.

Se você estiver executando AWSTOE em uma EC2 instância, a execução AWSTOE usa as mesmas permissões da função do IAM anexada à EC2 instância.

Para obter mais informações sobre as funções do IAM para EC2, consulte Funções do IAM para a Amazon EC2.

Os exemplos a seguir mostram como definir AWS credenciais usando as variáveis de `AWS_SECRET_ACCESS_KEY` ambiente `AWS_ACCESS_KEY_ID` e.

Para definir essas variáveis no Linux, macOS ou Unix, use `export`.

```
export AWS_ACCESS_KEY_ID=your_access_key_id
```

```
export AWS_SECRET_ACCESS_KEY=your_secret_access_key
```

Para definir essas variáveis no Windows usando PowerShell, use`$env`.

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

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

Para definir essas variáveis no Windows o prompt de comando, use `set`.

```
set AWS_ACCESS_KEY_ID=your_access_key_id
```

```
set AWS_SECRET_ACCESS_KEY=your_secret_access_key
```

Etapa 3: Desenvolver documentos de componentes localmente

Os componentes são criados com documentos YAML de texto simples. Para obter mais informações sobre sintaxe de documentos, consulte Use a estrutura de documentos de AWSTOE componentes para componentes personalizados.

Veja a seguir exemplos de documento de componente Hello World para ajudar você a começar.

Linux

Alguns dos exemplos de componente do Linux neste guia se referem a um arquivo de documento de componente chamado `hello-world-linux.yml`. Você pode usar o documento a seguir para começar a usar esses exemplos.

```
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

Alguns dos exemplos de componente do Windows neste guia se referem a um arquivo de documento de componente chamado `hello-world-windows.yml`. Você pode usar o documento a seguir para começar a usar esses exemplos.

```
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

Alguns dos exemplos de componente do macOS neste guia se referem a um arquivo de documento de componente chamado `hello-world-macos.yml`. Você pode usar o documento a seguir para começar a usar esses exemplos.

```
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.'
```

Etapa 4: Validar componentes AWSTOE

Você pode validar a sintaxe dos AWSTOE componentes localmente com o AWSTOE aplicativo. Os exemplos a seguir mostram o `validate` comando do AWSTOE aplicativo para validar a sintaxe de um componente sem executá-lo.

> **Note**
>
> O AWSTOE aplicativo pode validar somente a sintaxe do componente para o sistema operacional atual. Por exemplo, ao executar `awstoe.exe` no Windows, você não pode validar a sintaxe de um documento Linux que usa o módulo de ação `ExecuteBash`.

Linux ou macOS

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

Windows

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

Etapa 5: Executar AWSTOE componentes

O AWSTOE aplicativo pode executar uma ou mais fases de documentos especificados usando o argumento da linha de `--phases` comando. Os valores compatíveis para `--phases` são `build`, `validate` e `test`. Vários valores de fase podem ser inseridos como valores separados por vírgula.

Quando você fornece uma lista de fases, o AWSTOE aplicativo executa sequencialmente as fases especificadas de cada documento. Por exemplo, AWSTOE executa `validate` as fases `build` e de`document1.yaml`, seguidas pelas `validate` fases `build` e de`document2.yaml`.

Para garantir que seus logs sejam armazenados com segurança e retidos para solução de problemas, recomendamos configurar o armazenamento de logs no Amazon S3. No Image Builder, a localização do Amazon S3 para publicação de logs é especificada na configuração da infraestrutura. Para obter mais informações sobre configuração de infraestrutura, consulte Gerenciar a configuração da infraestrutura do Image Builder

Se uma lista de fases não for fornecida, o AWSTOE aplicativo executará todas as fases na ordem listada no documento YAML.

Para executar fases específicas em um ou vários documentos, use os comandos a seguir.

Fase única

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

Várias fases

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

Execução de documentos

Executar todas as fases em um único documento

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

Executar todas as fases em vários documentos

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

Insira as informações do Amazon S3 para fazer upload de AWSTOE registros de um caminho local definido pelo usuário (recomendado)

```
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
```

Executar todas as fases em um único documento e exibir todos os logs no console

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

Exemplo de comando

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

Executar documento com ID exclusivo

```
awstoe run --documents documentName.yaml --execution-id user-provided-id --
phases build,test
```

Obtenha ajuda com AWSTOE

```
awstoe --help
```

Use a estrutura de documentos de AWSTOE componentes para componentes personalizados

Para criar um componente usando a estrutura de componentes AWS Task Orchestrator and Executor(AWSTOE), você deve fornecer um documento baseado em YAML que represente as fases e etapas que se aplicam ao componente criado.Serviços da AWS use seu componente ao criar uma nova Amazon Machine Image (AMI) ou imagem de contêiner.

Tópicos

- Fluxo de trabalho do documento de componente
- Registro em log do componente
- Encadeamento de entrada e saída
- Esquema e definições do documento
- Documentos de exemplo
- Usar variáveis em seu documento de componente personalizado
- Usar constructos condicionais no AWSTOE
- Usar operadores de comparação em documentos de componente do AWSTOE
- Usar operadores lógicos em documentos de componente do AWSTOE
- Use estruturas em loop no AWSTOE

Fluxo de trabalho do documento de componente

O documento do AWSTOE componente usa fases e etapas para agrupar tarefas relacionadas e organizar essas tarefas em um fluxo de trabalho lógico para o componente.

> **Tip**
>
> O serviço que usa seu componente para criar uma imagem pode implementar regras sobre quais fases usar no processo de criação e quando essas fases podem ser executadas. É importante considerar isso ao projetar seu componente.

Fases

As fases representam a progressão do seu fluxo de trabalho por meio do processo de compilação da imagem. Por exemplo, o serviço Image Builder usa as fases `build` e `validate` durante seu estágio

de compilação para as imagens que ele produz. Ele usa as fases `test` e `container-host-test` durante o estágio de teste para garantir que o snapshot da imagem ou a imagem do contêiner produza os resultados esperados antes de criar a AMI final ou distribuir a imagem do contêiner.

Quando o componente é executado, os comandos associados a cada fase são aplicados na ordem em que aparecem no documento do componente.

Regras para as fases

- Cada nome de fase deve ser exclusivo dentro de um documento.
- Você pode definir várias fases em seu documento.
- Você deve incluir pelo menos uma das fases a seguir em seu documento:
  - compilação: para o Image Builder, essa fase geralmente é usada durante o estágio de compilação.
  - validação: para o Image Builder, essa fase geralmente é usada durante o estágio de compilação.
  - teste: para o Image Builder, essa fase geralmente é usada durante o estágio de teste.
- As fases sempre são executadas na ordem em que são definidas no documento. A ordem na qual eles são especificados para AWSTOE os comandos no não AWS CLI tem efeito.

Steps

As etapas são unidades individuais de trabalho que definem o fluxo de trabalho em cada fase. As etapas são executadas em ordem sequencial. No entanto, a entrada ou saída de uma etapa também pode alimentar uma etapa subsequente como entrada. Isso é chamado de “encadeamento”.

Regras para as etapas

- O nome da etapa deve ser exclusivo para a fase.
- A etapa deve usar uma ação compatível (módulo de ação) que retorne um código de saída.

  Para obter uma lista completa dos módulos de ação compatíveis, como eles funcionam, input/output valores e exemplos, consulteMódulos de ação compatíveis com o gerenciador de componentes do AWSTOE.

Registro em log do componente

AWSTOE cria uma nova pasta de log nas EC2 instâncias que são usadas para criar e testar uma nova imagem sempre que seu componente é executado. Para imagens de contêiner, a pasta de log é armazenada no contêiner.

Para ajudar na solução de problemas se algo der errado durante o processo de criação da imagem, o documento de entrada e todos os arquivos de saída AWSTOE criados durante a execução do componente são armazenados na pasta de registro.

O nome da pasta de log é composto pelas seguintes partes:

1. Diretório de log — quando um serviço executa um AWSTOE componente, ele passa para o diretório de log, junto com outras configurações do comando. Nos exemplos a seguir, mostramos o formato de arquivo de log usado pelo Image Builder.
   - Linux e macOS: `/var/lib/amazon/toe/`
   - Windows: `$env:ProgramFiles\Amazon\TaskOrchestratorAndExecutor\`
2. Prefixo do arquivo: esse é um prefixo padrão usado para todos os componentes: “`T0E_`”.
3. Tempo de execução — Este é um timestamp no formato YYYY-MM-DD _HH-MM-SS_UTC-0.
4. ID de execução — Esse é o GUID atribuído ao AWSTOE executar um ou mais componentes.

Exemplo: */var/lib/amazon/toe/*`T0E_`*2021-07-01_12-34-56_UTC-0_a1bcd2e3-45f6-789a-bcde-0fa1b2c3def4*

AWSTOE armazena os seguintes arquivos principais na pasta de log:

Arquivos de entrada

- document.yaml: o documento usado como entrada para o comando. Depois que o componente é executado, esse arquivo é armazenado como um artefato.

Arquivos de saída

- application.log: O log do aplicativo contém informações de nível de depuração com o timestamp do AWSTOE sobre o que está acontecendo enquanto o componente está sendo executado.
- detailedoutput.json: esse arquivo JSON tem informações detalhadas sobre o status de execução, entradas, saídas e falhas de todos os documentos, fases e etapas que se aplicam ao componente enquanto ele é executado.

- console.log — O log do console contém todas as informações de saída padrão (stdout) e erro padrão (stderr) que são AWSTOE gravadas no console enquanto o componente está em execução.
- chaining.json — Esse arquivo JSON representa otimizações aplicadas para resolver expressões de encadeamento.AWSTOE

> **Note**
>
> A pasta de log também pode conter outros arquivos temporários que não são abordados aqui.

Encadeamento de entrada e saída

O aplicativo de gerenciamento de AWSTOE configuração fornece um recurso para encadear entradas e saídas escrevendo referências nos seguintes formatos:

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

or

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

O atributo de encadeamento permite que você recicle o código e melhore a capacidade de manutenção do documento.

Regras para o encadeamento

- Expressões de encadeamento só podem ser usadas na seção de entradas de cada etapa.
- Instruções com expressões encadeadas devem estar entre aspas. Por exemplo:
  - Expressão inválida: `echo {{ phase.step.inputs.variable }}`
  - Expressão válida: `"echo {{ phase.step.inputs.variable }}"`
  - Expressão válida: `'echo {{ phase.step.inputs.variable }}'`
- Expressões de encadeamento podem referenciar variáveis de outras etapas e fases no mesmo documento. No entanto, o serviço de chamada pode ter regras que exijam que as expressões de encadeamento operem somente no contexto de um único estágio. Por exemplo, o Image Builder não é compatível com o encadeamento do estágio de compilação até o estágio de teste, pois ele executa cada estágio de forma independente.

- Os índices em expressões de encadeamento seguem a indexação com base em zero. O índice começa com zero (0) para referenciar o primeiro elemento.

Exemplos

Para se referir à variável de origem na segunda entrada da etapa de exemplo a seguir, o padrão de encadeamento é `{{ 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'
```

Para se referir à variável de saída (igual a "Hello") da etapa de exemplo a seguir, o padrão de encadeamento é `{{ build.SamplePowerShellStep.outputs.stdout }}`.

```
phases:
  - name: 'build'
    steps:
      - name: SamplePowerShellStep
        action: ExecutePowerShell
        timeoutSeconds: 120
        onFailure: Abort
        maxAttempts: 3
        inputs:
          commands:
            - 'Write-Host "Hello"'
```

Esquema e definições do documento

Veja a seguir o esquema YAML para um documento.

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

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

As definições de esquema para um documento são as seguintes.

Campo	Description	Tipo	Obrigatório
name	O nome do documento.	String	Não
descrição	Descrição do documento.	String	Não
schemaVersion	Versão do esquema do documento, atualmente 1.0.	String	Sim
phases	Uma lista de fases com suas etapas.	Lista	Sim

As definições de esquema para uma fase são as seguintes.

Campo	Description	Tipo	Obrigatório
name	Nome da fase.	String	Sim
steps	Lista das etapas da fase.	Lista	Sim

As definições de esquema para uma etapa são as seguintes.

Campo	Description	Tipo	Obrigatório	Valor padrão
name	Nome definido pelo usuário para a etapa.	String		
action	Palavra-chave referente ao módulo que executa a etapa.	String		
timeoutSeconds	Número de segundos em que a etapa é executada antes de falhar ou tentar novamente. Além disso, suporta o valor -1, que indica um tempo limite infinito. 0 e outros valores negativos não são permitidos.	Inteiro	Não	7.200 segundos (120 minutos)
onFailure	Especifica o que a etapa deve fazer em caso de falha. Os valores válidos são os seguintes: •	String	Não	Anular

Campo	Description	Tipo	Obrigatório	Valor padrão
	Abortar: falha na etapa após o número máximo de tentativas e para de ser executada. Define o status da fase e do documento para `Failed`. • Continuar: falha na etapa após o número máximo de tentativas e continua executando as etapas restantes. Define o status da fase e do documento para `Failed`. • Ignorar: define a etapa para `IgnoredFailure` depois do número máximo de tentativas			

Campo	Description	Tipo	Obrigatório	Valor padrão
	malsucedidas e continua executando as etapas restantes. Define o status da fase e do documento para `SuccessWithIgnoredFailure`.			
maxAttempts	Número máximo de tentativas permitidas antes de falhar na etapa.	Inteiro	Não	1
inputs	Contém os parâmetros exigidos pelo módulo de ação para executar a etapa.	Dict	Sim	

Documentos de exemplo

Os exemplos a seguir mostram documentos de AWSTOE componentes que executam tarefas para o sistema operacional de destino.

Linux

Exemplo 1: executar um arquivo binário personalizado

Veja a seguir um exemplo de documento para baixar e executar um arquivo binário personalizado em uma instância do 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

Exemplo 1: instalar atualizações do Windows

Veja a seguir um exemplo de documento para instalar todas as atualizações disponíveis do Windows, executar um script de configuração, validar as alterações antes da criação da AMI e testar as alterações após a criação da AMI.

```
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}}'
```

Exemplo 2: instalar o AWS CLI em uma instância do Windows

Veja a seguir um exemplo de documento que instala o AWS CLI em uma instância do Windows, usando o arquivo de configuração.

```
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 }}'
```

Exemplo 3: Instale o AWS CLI com o instalador MSI

A seguir está um exemplo de documento que instala o AWS CLI com o instalador 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

Exemplo 1: executar um arquivo binário personalizado no macOS

Veja a seguir um exemplo de documento para baixar e executar um arquivo binário personalizado em uma instância do 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 }}'
```

Usar variáveis em seu documento de componente personalizado

As variáveis oferecem uma forma de rotular dados com nomes significativos que podem ser usados em todo o aplicativo. Você pode definir variáveis personalizadas com formatos simples e legíveis para fluxos de trabalho complexos e referenciá-las no documento do componente do aplicativo YAML para um componente.AWSTOE

Esta seção fornece informações para ajudá-lo a definir variáveis para seu AWSTOE componente no documento do componente do aplicativo YAML, incluindo sintaxe, restrições de nome e exemplos.

Constantes

Constantes são variáveis imutáveis que não podem ser modificadas ou substituídas depois de definidas. As constantes podem ser definidas usando valores na `constants` seção de um AWSTOE documento.

Regras para denominação de constantes

- O nome deve ter entre 3 e 128 caracteres.
- O nome pode conter somente os caracteres alfanuméricos (A-Z, a-z, 0-9), hífens (-) ou sublinhados (_).
- O nome deve ser exclusivo dentro do documento.
- O nome deve ser especificado como uma cadeia YAML.

Sintaxe

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

Nome da chave	Obrigatório	Descrição
`name`	Sim	Nome da constante. Deve ser exclusivo para o documento (não deve ser igual a nenhum outro nome de parâmetro ou constante).
`value`	Sim	Valor da constante.
`type`	Sim	Tipo da constante. O tipo compatível é `string`.

Valores de constante de referência em um documento

Você pode referenciar constantes em informações de etapa ou loop dentro do seu documento YAML, da seguinte maneira:

- As referências da constante diferenciam letras maiúsculas de minúsculas e o nome deve corresponder exatamente.
- O nome deve estar entre colchetes duplos. `{{ MyConstant }}`
- Dentro dos parêntesis, os espaços são permitidos e são cortados automaticamente. Por exemplo, todas as referências a seguir são válidas:

  `{{ MyConstant }}`, `{{ MyConstant}}`, `{{MyConstant }}`, `{{MyConstant}}`
- A referência no documento YAML deve ser especificada como uma string (entre aspas simples ou duplas).

  Por exemplo: `- {{ MyConstant }}` não é válido, pois não é identificado como uma string.

  No entanto, as seguintes referências são válidas: `- '{{ MyConstant }}'` e `- "{{ MyConstant }}"`.

Exemplos

Constante referenciada nas informações da etapa

```
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'
```

Constante referenciada nas informações de 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 }}
```

Parâmetros

Os parâmetros são variáveis mutáveis, com configurações que o aplicativo de chamada pode fornecer no runtime. Você pode definir parâmetros na seção `Parameters` do documento YAML.

Regras para denominação de parâmetro

- O nome deve ter entre 3 e 128 caracteres.

- O nome pode conter somente os caracteres alfanuméricos (A-Z, a-z, 0-9), hífens (-) ou sublinhados (_).
- O nome deve ser exclusivo dentro do documento.
- O nome deve ser especificado como uma cadeia YAML.

Sintaxe

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

Nome da chave	Obrigatório	Descrição
`name`	Sim	O nome do parâmetro. Deve ser exclusivo para o documento (não deve ser igual a nenhum outro nome de parâmetro ou constante).
`type`	Sim	O tipo de dado válido do parâmetro. Os tipos compatíveis incluem: `string`.
`default`	Não	O valor padrão do parâmetro.
`description`	Não	Descreve o parâmetro.

Valores de parâmetros de referência em um documento

Você pode referenciar parâmetros em etapa ou loop dentro do seu documento YAML, da seguinte maneira:

- As referências de parâmetro diferenciam letras maiúsculas de minúsculas e o nome deve corresponder exatamente.
- O nome deve estar entre colchetes duplos. `{{ MyParameter }}`

- Dentro dos parêntesis, os espaços são permitidos e são cortados automaticamente. Por exemplo, todas as referências a seguir são válidas:

  `{{ MyParameter }}`, `{{ MyParameter}}`, `{{MyParameter }}`, `{{MyParameter}}`
- A referência no documento YAML deve ser especificada como uma string (entre aspas simples ou duplas).

  Por exemplo: `- {{ MyParameter }}` não é válido, pois não é identificado como uma string.

  No entanto, as seguintes referências são válidas: `- '{{ MyParameter }}'` e `- "{{ MyParameter }}"`.

Exemplos

Os exemplos a seguir mostram como usar parâmetros no seu documento YAML:

- Consulte um parâmetro nas informações da etapa:

```
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'
```

- Consulte um parâmetro nas informações de 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 }}
```

Substituir parâmetros no runtime

Você pode usar a `--parameters` opção AWS CLI com um par de valores-chave para definir um valor de parâmetro em tempo de execução.

- Especifique o par chave-valor do parâmetro como nome e valor, separados por um sinal de igual (<name>=<value>).
- Vários parâmetros devem ser separados por uma vírgula.
- Os nomes dos parâmetros que não são encontrados no documento do componente YAML são ignorados.
- O nome e o valor do parâmetro são obrigatórios.

Important

Os parâmetros do componente são valores de texto simples e estão logados em AWS CloudTrail. Recomendamos que você use AWS Secrets Manager ou o AWS Systems Manager Parameter Store para armazenar seus segredos. Para obter mais informações sobre o Secrets Manager, consulte O que é o Secrets Manager? no AWS Secrets Manager Guia do usuário. Para obter mais informações sobre o AWS Systems Manager Parameter Store, consulte AWS Systems Manager Parameter Store no AWS Systems Manager Guia do usuário.

Sintaxe

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

Opção da CLI:	Obrigatório	Description
--parâmetros *name* =*value*,...	Não	Essa opção usa uma lista de pares chave-valor, com o nome do parâmetro como chave.

Exemplos

Os exemplos a seguir mostram como usar parâmetros no seu documento YAML:

- O par valor-chave do parâmetro especificado nessa opção do `--parameter` não é válido:

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

- Defina um par valor-chave de parâmetro com a opção do `--parameter` no AWS CLI:

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

- Defina vários pares valor-chave de parâmetros com a opção `--parameter` no AWS CLI:

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

Use os parâmetros do Systems Manager Parameter Store

Você pode referenciar os parâmetros do AWS Systems Manager Parameter Store (parâmetros SSM) em seus documentos de componentes prefixando variáveis com. `aws:ssm` Por exemplo,

`{{ aws:ssm:/my/param }}`resolve o valor do parâmetro SSM. `/my/param`

Esse recurso oferece suporte aos seguintes tipos de parâmetros SSM:

- String — Mapeia para o tipo de AWSTOE string.
- StringList — Mapas para o AWSTOE`StringList` tipo.

- SecureString — Mapeia para o tipo de AWSTOE string.

Para obter mais informações sobre o Parameter Store, consulte AWS Systems Manager Parameter Store no Guia AWS Systems Manager do usuário.

Você também pode referenciar AWS Secrets Manager segredos usando um parâmetro `SecureString` SSM. Por exemplo: `{{ aws:ssm:/aws/reference/secretsmanager/test/test-secret }}`. Para obter mais informações, consulte Como referenciar AWS Secrets Manager segredos dos parâmetros do Parameter Store.

> **Important**
>
> O Image Builder exclui a resolução de `SecureString` parâmetros de seus registros. No entanto, você também é responsável por garantir que informações confidenciais não sejam registradas por meio de comandos emitidos no documento do componente. Por exemplo, se você usar o `echo` comando com uma string segura, o comando grava um valor em texto simples no log.

Permissões obrigatórias do IAM

Para usar os parâmetros do Systems Manager em seus componentes, sua função de instância deve ter a `ssm:GetParameter` permissão para o recurso de parâmetros ARN. Por exemplo:

JSON

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

Para acessar valores criptografados, você também precisará das seguintes permissões:

- Adicione `kms:Decrypt` `SecureString` parâmetros ou AWS Secrets Manager valores que são criptografados com um cliente gerenciado AWS KMS key.
- Adicione `secretsmanager:GetSecretValue` se você fizer referência a um segredo do Secrets Manager.

Fazer referência a um parâmetro SSM em um documento componente

O exemplo a seguir mostra como referenciar um parâmetro do Systems Manager Parameter Store dos parâmetros do Systems Manager em um componente:

```
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 }}."
```

Resolução variável de tempo de execução dinâmico para parâmetros SSM

AWSTOE fornece a seguinte função integrada que você pode usar em referências de variáveis para manipular ou transformar valores em tempo de execução.

função de resolução

A `resolve` função resolve uma referência de variável dentro de outra referência de variável, permitindo a referência dinâmica de nomes de variáveis. Isso é útil ao trabalhar com parâmetros SSM em que parte do caminho do parâmetro pode ser variável e transmitida como um parâmetro do documento.

A `resolve` função suporta somente a resolução dinâmica da parte do nome de um parâmetro SSM.

Sintaxe

O `dynamic_variable` exemplo a seguir representa o nome de um parâmetro SSM e deve ser um dos seguintes:

- Uma referência de parâmetro SSM (por exemplo,`aws:ssm:`*`/my/param`*)
- Uma referência de parâmetro do documento do componente (por exemplo,*`parameter-name`*)

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

Exemplo: Resolver um parâmetro SSM em tempo de execução

O exemplo a seguir mostra como usar a `resolve` função em um documento do componente YAML:

```
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) }}"
```

Usar constructos condicionais no AWSTOE

Os constructos condicionais realizam diferentes ações no documento do componente dependendo de a expressão condicional especificada ser avaliada como `true` ou `false`. Você pode usar o constructo `if` para controlar o fluxo de execução no documento do componente.

Constructo “if”

Você pode usar o constructo `if` para avaliar se uma etapa deve ser executada ou não. Por padrão, quando a expressão condicional `if` é avaliada como `true`, o AWSTOE executa a etapa e, quando a condição é avaliada como `false`, o AWSTOE pula a etapa. Se uma etapa for ignorada, ela poderá ser tratada como bem-sucedida quando o AWSTOE avaliar se a fase e o documento foram executados com êxito.

> **Note**
>
> Uma instrução `if` só é avaliada uma vez, mesmo que a etapa acione uma reinicialização. Se uma etapa for reiniciada, ela reconhecerá que a instrução `if` já foi avaliada e continuará de onde parou.

Sintaxe

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

Nome da chave	Obrigatório	Descrição
Expressão condicional	Sim	A expressão condicional pode conter exatamente um dos seguintes tipos de operador no nível superior. • Operador de comparação — Para obter uma lista de operadores de comparação e informações sobre como eles funcionam em documentos de AWSTOE component

Nome da chave	Obrigatório	Descrição
		es, consulteOperadores de comparação. • Operador lógico: os operadores lógicos incluem `and`, `or` e `not` e operam em um ou mais operadores de comparação. Para obter mais informações sobre como os operadores lógicos funcionam em documentos do componentes do AWSTOE , consulte Operadores lógicos. Se sua expressão precisar satisfazer várias condições, use um operador lógico para especificar suas condições.
then	Não	Define a ação a ser executada se a expressão condicional for avaliada como `true`.
else	Não	Define a ação a ser executada se a expressão condicional for avaliada como `false`.

Nome da chave	Obrigatório	Descrição
ações de etapa	Condicional	Ao usar `then` ou `else`, você deve especificar uma das seguintes ações de etapa: • Anular: o AWSTOE marca a etapa como com falha. • Executar: o AWSTOE executa a etapa. • Ignorar: o AWSTOE pula a etapa.

Exemplo 1: instalar um pacote

As etapas de exemplo a seguir de um documento de AWSTOE componente usam operadores lógicos para testar um valor de parâmetro e executar os comandos apropriados do gerenciador de pacotes para instalar um aplicativo se o pacote for descompactado.

```
- 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
```

Exemplo 2: ignorar uma etapa

O exemplo a seguir mostra duas maneiras de pular uma etapa. Uma usa um operador lógico e a outra usa um operador de comparação com a ação de etapa `Skip`.

```
# 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'
```

Usar operadores de comparação em documentos de componente do AWSTOE

Você pode usar os operadores de comparação a seguir com o módulo de ação Assert e com expressões condicionais que usam o Constructo “if”. Um operador de comparação pode operar em um único valor (por exemplo, `stringIsEmpty`) ou comparar um valor de referência com um segundo valor (valor variável) para determinar se a expressão condicional é avaliada como `true` ou `false`.

Se a comparação operar em dois valores, o segundo valor poderá ser uma variável de encadeamento.

Ao comparar valores de um tipo diferente, as seguintes conversões de valor podem ocorrer antes da comparação:

- Para comparações numéricas, se o valor da variável for uma string, AWSTOE converte a string em um número antes da avaliação. Se a conversão não for possível, a comparação responderá com `false`. Por exemplo, se o valor da variável for `"1.0"`, a conversão funcionará, mas, se o valor da variável for `"a10"`, a conversão falhará.
- Para comparações de strings, se o valor da variável for um número, AWSTOE converte-o em uma string antes da avaliação.

Comparar strings

Os operadores de comparação a seguir funcionam com strings de caracteres para comparar valores, testar espaços ou uma string vazia ou comparar um valor de entrada com um padrão regex. As comparações de strings não diferenciam maiúsculas de minúsculas e não removem os espaços do início ou do final das entradas de string.

Operadores de comparação de strings

- stringIsEmpty
- stringIsWhitespace
- stringEquals
- stringLessThan
- stringLessThanEquals
- stringGreaterThan
- stringGreaterThanEquals

- patternMatches

stringIsEmpty

O operador `stringIsEmpty` responderá com `true` se a string especificada não contiver nenhum caractere. Por exemplo:

```
# Evaluates to true
stringIsEmpty: ""

# Evaluates to false
stringIsEmpty: " "

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

stringIsWhitespace

Testa se a string especificada para `stringIsWhitespace` contém somente espaços. Por exemplo:

```
# Evaluates to true
stringIsWhitespace: "   "

# Evaluates to false
stringIsWhitespace: ""

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

stringEquals

Testa se a string especificada para `stringEquals` é uma correspondência exata para a string especificada no parâmetro `value`. Por exemplo:

```
# 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

Testa se a string especificada para `stringLessThan` é menor do que a string especificada no parâmetro `value`. Por exemplo:

```
# 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'
```

stringLessThanIgual

Testa se a string especificada para `stringLessThanEquals` é menor ou igual à string especificada no parâmetro `value`. Por exemplo:

```
# 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

Testa se a string especificada para `stringGreaterThan` é maior do que a string especificada no parâmetro `value`. Por exemplo:

```
# 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'
```

stringGreaterThanIgual

Testa se a string especificada para `stringGreaterThanEquals` é maior ou igual à string especificada no parâmetro `value`. Por exemplo:

```
# 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'
```

patternMatches

Testa se a string especificada para `value` corresponde ao padrão regex especificado para `patternMatches`. A comparação usa o pacote Golang regexp, que está em conformidade com a sintaxe. RE2 Para obter mais informações sobre RE2 regras, consulte o repositório google/ re2 em. GitHub

O exemplo a seguir mostra uma correspondência de padrões que responde com `true`:

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

Comparar números

Os operadores de comparação a seguir funcionam com números. Os valores fornecidos para esses operadores devem ser de um dos tipos a seguir, de acordo com a especificação YAML. O suporte a comparações numéricas usa o operador de comparação do pacote Golang Big, por exemplo: func (*Float) Cmp.

- Inteiro

- Float (com base no float64, compatível com números de -1.7e+308 a +1.7e+308)
- Uma string que corresponde ao seguinte padrão de regex: `^[-+]?([0-9]+[.])?[0-9]+$`

Operadores de comparação de números

- numberEquals
- numberLessThan
- numberLessThanEquals
- numberGreaterThan
- numberGreaterThanEquals

numberEquals

Testa se o número especificado para `numberEquals` é igual ao número especificado no parâmetro `value`. Todos os seguintes exemplos de comparação respondem com `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

Testa se o número especificado para `numberLessThan` é menor do que o número especificado no parâmetro `value`. Por exemplo:

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

# Evaluates to true
numberLessThan: 2
value: 1.9

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

numberLessThanIgual

Testa se o número especificado para `numberLessThanEquals` é menor ou igual ao número especificado no parâmetro `value`. Por exemplo:

```
# 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

Testa se o número especificado para `numberGreaterThan` é maior do que o número especificado no parâmetro `value`. Por exemplo:

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

# Evaluates to true
numberGreaterThan: 1
value: 1.1

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

numberGreaterThanIgual

Testa se o número especificado para `numberGreaterThanEquals` é maior ou igual ao número especificado no parâmetro `value`. Por exemplo:

```
# 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
```

Verificar arquivos

Os operadores de comparação a seguir verificam o hash do arquivo ou se um arquivo ou pasta existe.

Operadores de arquivos e pastas

- binaryExists

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

binaryExists

Testa se uma aplicação está disponível no caminho atual. Por exemplo:

```
binaryExists: 'foo'
```

> **Note**
>
> Em sistemas Linux e macOS, para um aplicativo chamado*foo*, isso funciona da mesma forma que o seguinte comando bash:type ***foo*** >/dev/null 2>&1, onde $? == 0 indica uma comparação bem-sucedida.
> Em sistemas Windows, para um aplicativo chamado*foo*, isso funciona da mesma forma que o PowerShell comando & C:\Windows\System32\where.exe /Q ***foo*** que $LASTEXITCODE = 0 indica uma comparação bem-sucedida.

fileExists

Testa se um arquivo existe no caminho especificado. Você pode fornecer um caminho absoluto ou relativo. Se o local especificado existir e for um arquivo, a comparação será avaliada como `true`. Por exemplo:

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

> **Note**
>
> Nos sistemas Linux e macOS, isso funciona da mesma forma que o seguinte comando bash: -d ***/path/to/file***, em que $? == 0 indica uma comparação bem-sucedida.

Nos sistemas Windows, isso funciona da mesma forma que o PowerShell comandoTest-Path -Path '***C:\path\to\file***' -PathType 'Leaf'.

folderExists

Testa se uma pasta existe no caminho especificado. Você pode fornecer um caminho absoluto ou relativo. Se o local especificado existir e for uma pasta, a comparação será avaliada como `true`. Por exemplo:

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

> **Note**
>
> Nos sistemas Linux e macOS, isso funciona da mesma forma que o seguinte comando bash: -d ***/path/to/folder***, em que $? == 0 indica uma comparação bem-sucedida. Nos sistemas Windows, isso funciona da mesma forma que o PowerShell comandoTest-Path -Path '***C:\path\to\folder***' -PathType 'Container'.

arquivo é igual MD5 a

Testa se o MD5 hash de um arquivo é igual a um valor especificado. Por exemplo:

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

arquivo é igual SHA1 a

Testa se o SHA1 hash de um arquivo é igual a um valor especificado. Por exemplo:

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

arquivo é igual SHA256 a

Testa se o SHA256 hash de um arquivo é igual a um valor especificado. Por exemplo:

```
fileSHA256Equals: '<SHA256Hash>'
```

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

arquivo é igual SHA512 a

Testa se o SHA512 hash de um arquivo é igual a um valor especificado. Por exemplo:

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

Usar operadores lógicos em documentos de componente do AWSTOE

Você pode usar os operadores lógicos a seguir para adicionar ou modificar expressões condicionais no documento do componente. AWSTOE avalia expressões condicionais na ordem em que as condições são especificadas. Para obter mais informações sobre operadores de comparação para documentos de componente, consulte Usar operadores de comparação em documentos de componente do AWSTOE.

and

Com o operador `and`, você pode avaliar duas ou mais comparações como uma única expressão. A expressão é avaliada como `true` quando todas as condições na lista são verdadeiras. Caso contrário, a expressão será avaliada como `false`.

Exemplos:

O exemplo a seguir faz duas comparações: uma string e um número. Ambas as comparações são verdadeiras, então a expressão é avaliada como verdadeira.

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

O exemplo a seguir também realiza duas comparações. A primeira comparação é falsa e, em dado momento, a avaliação é interrompida e a segunda comparação é ignorada. A expressão é avaliada como `false`.

```
and:
```

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

or

Com o operador `or`, você pode avaliar duas ou mais comparações como uma única expressão. A expressão é avaliada como `true` quando uma das comparações especificadas é verdadeira. Se nenhuma das comparações especificadas for avaliada como `true`, a expressão será avaliada como `false`.

Exemplos:

O exemplo a seguir faz duas comparações: uma string e um número. A primeira comparação é verdadeira, então a expressão é avaliada como `true` e a segunda comparação é ignorada.

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

O exemplo a seguir também realiza duas comparações. A primeira comparação é falsa e a avaliação continua. A segunda comparação é verdadeira, então a expressão é avaliada como `true`.

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

No exemplo final, ambas as comparações são falsas, então a expressão é avaliada como `false`.

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

not

Com o operador `not`, você pode negar uma única comparação. A expressão é avaliada como `true` se a comparação for falsa. Se a comparação for verdadeira, a expressão será avaliada como `false`.

Exemplos:

O exemplo a seguir executa uma comparação de strings. A comparação é falsa, então a expressão é avaliada como `true`.

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

O exemplo a seguir também executa uma comparação de strings. A comparação é verdadeira, então a expressão é avaliada como `false`.

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

Use estruturas em loop no AWSTOE

Esta seção fornece informações para ajudar você a criar estruturas em loop no AWSTOE. Estruturas em loop definem uma sequência repetida de instruções. Você pode usar os seguintes tipos de estruturas em loop no AWSTOE:

- `for` estruturas: itere sobre uma sequência limitada de inteiros.
- `forEach` estruturas
  - `forEach` loop com lista de entrada: itera sobre uma coleção finita de strings.
  - `forEach` loop com lista delimitada: itera sobre uma coleção finita de strings unidas por um delimitador.

> **Note**
>
> As estruturas em loop suportam somente tipos de dados de string.

Tópicos de estruturas em loop

- Variáveis de iteração de referência
- Tipos de estruturas em loop
- Campos de etapa
- Saídas de etapa e iteração

Variáveis de iteração de referência

Para se referir ao índice e ao valor da variável de iteração atual, a expressão de referência `{{ loop.* }}` deve ser usada no corpo de entrada de uma etapa que contém uma estrutura em loop. Essa expressão não pode ser usada para se referir às variáveis de iteração da estrutura em loop de outra etapa.

A expressão de referência consiste nos seguintes membros:

- `{{ loop.index }}`: a posição ordinal da iteração atual, indexada em `0`.
- `{{ loop.value }}`: o valor associado à variável de iteração atual.

Nomes de loop

Todas as estruturas em loop têm um campo de nome opcional para identificação. Se um nome de loop for fornecido, ele poderá ser usado para se referir às variáveis de iteração no corpo de entrada da etapa. Para se referir aos índices e valores de iteração de um loop nomeado, use `{{ <loop_name>.* }}` com `{{ loop.* }}` no corpo de entrada da etapa. Essa expressão não pode ser usada para se referir à estrutura em loop nomeada de outra etapa.

A expressão de referência consiste nos seguintes membros:

- `{{ <loop_name>.index }}`: a posição ordinal da iteração atual do loop nomeado, indexada em `0`.
- `{{ <loop_name>.value }}`: o valor associado à variável de iteração atual do loop nomeado.

Resolver expressões de referência

AWSTOE Resolve as expressões de referência da seguinte forma:

- `{{ <loop_name>.* }}`— AWSTOE resolve essa expressão usando a seguinte lógica:

  - Se o loop da etapa em execução no momento corresponder ao valor do `<loop_name>`, a expressão de referência será resolvida para a estrutura em loop da etapa em execução no momento.
  - `<loop_name>` resolve para a estrutura em loop nomeada se ela aparecer na etapa em execução no momento.
- `{{ loop.* }}`— AWSTOE resolve a expressão usando a construção de loop definida na etapa atualmente em execução.

Se expressões de referência forem usadas em uma etapa que não contém um loop, elas AWSTOE não resolverão as expressões e elas aparecerão na etapa sem substituição.

> **Note**
>
> As expressões de referência devem estar entre aspas duplas para serem interpretadas corretamente pelo compilador YAML.

Tipos de estruturas em loop

Esta seção fornece informações e exemplos sobre os tipos de estrutura em loop que podem ser usados no AWSTOE.

Tipos de estrutura em loop

- loop for
- loop forEach com lista de entrada
- loop forEach com lista delimitada

loop **`for`**

O loop `for` itera em um intervalo de inteiros especificado dentro de um limite delineado pelo início e pelo fim das variáveis. Os valores de iteração estão no conjunto `[start, end]` e incluem valores limite.

AWSTOE verifica os `updateBy` valores `startend`, e para garantir que a combinação não resulte em um loop infinito.

esquema de loop `for`

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

entrada de loop **`for`**

Campo	Descrição	Tipo	Obrigatório	Padrão
`name`	Nome exclusivo do loop. Ele deve ser exclusivo em comparação com outros nomes de loop na mesma fase.	String	Não	""
`start`	Valor inicial da iteração. Não aceita expressões de encadeamento.	Inteiro	Sim	n/a
`end`	Valor final da iteração. Não aceita expressões de encadeamento.	Inteiro	Sim	n/a
`updateBy`	Diferença pela qual um valor iterativo é	Inteiro	Sim	n/a

Campo	Descrição	Tipo	Obrigatório	Padrão
	atualizado por meio da adição. Deve ser um valor negativo ou positivo diferente de zero. Não aceita expressões de encadeamento.			

exemplo de entrada de loop `for`

```
  - 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
```

loop **`forEach`** com lista de entrada

O loop `forEach` itera em uma lista explícita de valores, que podem ser cadeias de caracteres e expressões encadeadas.

loop `forEach` com esquema de lista de entrada

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

loop **`forEach`** com entrada de lista de entrada

Campo	Descrição	Tipo	Obrigatório	Padrão
`name`	Nome exclusivo do loop. Ele deve ser exclusivo em comparação com outros nomes de loop na mesma fase.	String	Não	""
Lista de string de loop `forEach`	Lista de strings para iteração. Aceita expressões encadeadas como cadeias de strings na lista. As expressões encadeadas devem estar entre aspas duplas para serem interpretadas corretamente	Lista de strings	Sim	n/a

Campo	Descrição	Tipo	Obrigatório	Padrão
	pelo compilador YAML.			

loop `forEach` com lista de entrada (exemplo 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
```

loop `forEach` com lista de entrada (exemplo 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 }}"
```

loop `forEach` com lista de entrada (exemplo 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 }}"
```

loop **`forEach`** com lista delimitada

O loop itera sobre uma string contendo valores separados por um delimitador. Para iterar sobre os constituintes da string, AWSTOE usa o delimitador para dividir a string em uma matriz adequada para iteração.

loop `forEach` com esquema de lista delimitada

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

loop **`forEach`** com entrada de lista delimitada

Campo	Descrição	Tipo	Obrigatório	Padrão
`name`	Nome exclusivo dado ao loop. Deve ser	String	Não	""

Campo	Descrição	Tipo	Obrigatório	Padrão
	exclusivo quando comparado com outros nomes de loop na mesma fase.			
`list`	Uma string composta por strings constituintes unidas por um caractere delimitador comum. Também aceita expressões encadeadas. No caso de expressões encadeadas, certifique-se de que elas estejam entre aspas duplas para uma interpretação correta pelo compilador YAML.	String	Sim	n/a

Campo	Descrição	Tipo	Obrigatório	Padrão
`delimiter`	Caractere usado para separar as strings dentro de um bloco. O padrão é o caractere de vírgula. Somente um caractere delimitador é permitido na lista fornecida: • Ponto: `"."` • Vírgula: `","` • Ponto e vírgula: `";"` • Dois pontos: `":"` • Nova linha: `"\n"` • Tab: `"\t"` • Espaço: `" "` • Hífen: `"-"` • Sublinhado: `"_"`	String	Não	Vírgula: `","`

Campo	Descrição	Tipo	Obrigatório	Padrão
	Expressões encadeadas não podem ser usadas.			

> **Note**
>
> O valor de `list` é tratado como uma string imutável. Se a fonte de `list` for alterada durante o runtime, ela não será refletida durante a execução.

loop `forEach` com lista delimitada (exemplo 1)

Este exemplo usa o seguinte padrão de expressão de encadeamento para se referir à saída de outra etapa: `<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 }}"
```

loop `forEach` com lista delimitada (exemplo 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 }}"
```

Campos de etapa

Os loops fazem parte de uma etapa. Qualquer campo relacionado à execução de uma etapa não é aplicado a iterações individuais. Os campos de etapa se aplicam somente no nível da etapa, da seguinte forma:

- TimeoutSeconds: todas as iterações do loop devem ser executadas dentro do período especificado por esse campo. Se a execução do loop atingir o tempo limite, AWSTOE execute a política de repetição da etapa e redefina o parâmetro de tempo limite para cada nova tentativa. Se a execução do loop exceder o valor do tempo limite após atingir o número máximo de novas tentativas, a mensagem de falha da etapa indicará que a execução do loop atingiu o tempo limite.
- onFailure: o tratamento de falhas é aplicado à etapa da seguinte forma:
  - Se OnFailure estiver definido como`Abort`, AWSTOE sai do loop e repete a etapa de acordo com a política de repetição. Após o número máximo de tentativas de repetição, AWSTOE marca a etapa atual como falhada e interrompe a execução do processo.

    AWSTOE define o código de status da fase principal e do documento como`Failed`.

    > **Note**
    >
    > Nenhuma outra etapa é executada após a etapa malsucedida.

  - Se onFailure estiver definido como `Continue`, AWSTOE sai do loop e repete a etapa de acordo com a política de repetição. Após o número máximo de tentativas de repetição, AWSTOE marca a etapa atual como falhada e continua executando a próxima etapa.

    AWSTOE define o código de status da fase principal e do documento como`Failed`.
  - Se onFailure estiver definido como `Ignore`, AWSTOE sai do loop e repete a etapa de acordo com a política de repetição. Após o número máximo de tentativas de repetição, AWSTOE marca a etapa atual como `IgnoredFailure` e continua executando a próxima etapa.

    AWSTOE define o código de status da fase principal e do documento como`SuccessWithIgnoredFailure`.

    > **Note**
    >
    > Isso ainda é considerado uma execução bem-sucedida, mas inclui informações para que você saiba que uma ou mais etapas falharam e foram ignoradas.

- maxAttempts: para cada nova tentativa, toda a etapa e todas as iterações são executadas desde o início.
- status: o status geral da execução de uma etapa. `status` não representa o status de iterações individuais. O status de uma etapa com loops é determinado da seguinte forma:
  - Se uma única iteração falhar na execução, o status de uma etapa aponta para uma falha.
  - Se todas as iterações forem bem-sucedidas, o status de uma etapa indicará um sucesso.
- startTime: a hora geral de início da execução de uma etapa. Não representa a hora de início das iterações individuais.
- endTime: a hora geral de término da execução de uma etapa. Não representa a hora de término das iterações individuais.
- failureMessage: inclui os índices de iteração que falharam em caso de erros sem tempo limite. Em caso de erros de tempo limite, a mensagem indica que a execução do loop falhou. Mensagens de erro individuais para cada iteração não são fornecidas para minimizar o tamanho das mensagens de falha.

Saídas de etapa e iteração

Cada iteração contém uma saída. No final de uma execução de loop, AWSTOE consolida todas as saídas de iteração bem-sucedidas em. `detailedOutput.json` As saídas consolidadas são um agrupamento de valores que pertencem às chaves de saída correspondentes, conforme definido no esquema de saída do módulo de ação. O exemplo a seguir mostra como as saídas são consolidadas:

Saída de **`ExecuteBash`** para a Iteração 1

```
{
 "stdout":"Hello"
}
```

Saída de **`ExecuteBash`** para a Iteração 2

```
{
 "stdout":"World"
}
```

Saída de **`ExecuteBash`** para a Etapa

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

Por exemplo, `ExecuteBash`, `ExecutePowerShell` e `ExecuteBinary` são módulos de ação que retornam `STDOUT` como saída do módulo de ação. As mensagens `STDOUT` são unidas ao novo caractere de linha para produzir a saída geral da etapa em `detailedOutput.json`.

AWSTOE não consolidará os resultados de iterações malsucedidas.

Módulos de ação compatíveis com o gerenciador de componentes do AWSTOE

Os serviços de criação de imagens, como o EC2 Image Builder, usam módulos de AWSTOE ação para ajudar a configurar as EC2 instâncias usadas para criar e testar imagens de máquina personalizadas. Esta seção descreve os recursos dos módulos de AWSTOE ação mais usados e como configurá-los, incluindo exemplos.

Os componentes são criados com documentos YAML de texto simples. Para obter mais informações sobre sintaxe de documentos, consulte Use a estrutura de documentos de AWSTOE componentes para componentes personalizados.

> **Note**
>
> Todos os módulos de ação usam a mesma conta do atendente do Systems Manager quando são executados, os quais são `root`, no Linux, e `NT Authority\SYSTEM`, no Windows.

A referência cruzada a seguir categoriza os módulos de ação pelo tipo de ação que eles executam.

Execução geral

- Assert (Linux, Windows, macOS)
- ExecuteBash (Linux, macOS)
- ExecuteBinary (Linux, Windows, macOS)

- ExecuteDocument (Linux, Windows, macOS)
- ExecutePowerShell (Windows)

Download e upload de arquivos

- S3Download (Linux, Windows, macOS)
- S3Upload (Linux, Windows, macOS)
- WebDownload (Linux, Windows, macOS)

Operações do sistema de arquivos

- 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)

Ações de instalação de software

- InstallMSI (Windows)
- UninstallMSI (Windows)

Ações do sistema

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

Módulos de execução geral

A seção a seguir contém detalhes dos módulos de ação que executam comandos e controlam fluxos de trabalho de execução.

Módulos de ação de execução geral

- Assert (Linux, Windows, macOS)
- ExecuteBash (Linux, macOS)
- ExecuteBinary (Linux, Windows, macOS)
- ExecuteDocument (Linux, Windows, macOS)
- ExecutePowerShell (Windows)

Assert (Linux, Windows, macOS)

O módulo de ação Assert realiza comparações de valores usando Operadores de comparação ou Operadores lógicos como entrada. O resultado da expressão do operador (verdadeiro ou falso) indica o status geral de êxito ou falha da etapa.

Se a comparação ou a expressão do operador lógico for avaliada como `true`, a etapa será marcada como `Success`. Caso contrário, a etapa será marcada como `Failed`. Se a etapa falhar, o parâmetro on`Failure` determinará o resultado da etapa.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório
`input`	Contém um único operador lógico ou de comparação. Observe que os operadores lógicos podem conter mais de um operador de comparação.	Isso é variável, dependendo do operador.	Sim

Exemplo de entrada: uma comparação simples usando o operador de comparação **`stringEquals`**

Este exemplo é avaliado como `true`.

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

Exemplo de entrada: comparações regex usando o operador de comparação **`patternMatches`**

Todos estes exemplos são avaliados como `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'
```

Exemplo de entrada: comparações aninhadas com operadores lógicos e variáveis encadeadas

O exemplo a seguir demonstra comparações aninhadas com operadores lógicos que usam comparações com variáveis encadeadas. O `Assert` avalia como `true` se alguma das seguintes afirmações for verdadeira:

- A `ApplicationVersion` é superior a `2.0` e a `CPUArchitecture` é igual a `arm64`.
- A `CPUArchitecture` é igual a `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 }}'
```

Saída:

O resultado do `Assert` é êxito ou falha da etapa.

ExecuteBash (Linux, macOS)

O módulo de ExecuteBashação permite que você execute scripts bash com códigos/comandos de shell embutidos. Este módulo é compatível com Linux.

Todos os comandos e instruções que você especifica no bloco de comandos são convertidos em um arquivo (por exemplo, `input.sh`) e executados com o bash shell. O resultado da execução do arquivo shell é o código de saída da etapa.

O ExecuteBashmódulo manipula as reinicializações do sistema se o script sair com um código de saída de. 194 Quando iniciado, o aplicativo executa uma das seguintes ações:

- O aplicativo entrega o código de saída ao chamador se ele for executado pelo Systems Manager Agent. O Systems Manager Agent controla a reinicialização do sistema e executa a mesma etapa que iniciou a reinicialização, conforme descrito em Como reinicializar a instância gerenciada a partir de scripts.
- O aplicativo salva o `executionstate` atual, configura um gatilho de reinicialização para executar o aplicativo novamente e reinicia o sistema.

Após a reinicialização do sistema, o aplicativo executa a mesma etapa que iniciou a reinicialização. Se você precisar dessa funcionalidade, deverá escrever scripts idempotentes que possam lidar com várias invocações do mesmo comando shell.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório
`commands`	Contém uma lista de instruções ou comandos a serem executados de acordo com a sintaxe do bash. O YAML de várias linhas é permitido.	Lista	Sim

Exemplo de entrada: antes e depois de uma reinicialização

```
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
```

Saída

Campo	Descrição	Tipo
`stdout`	Saída padrão da execução do comando.	string

Se você iniciar uma reinicialização e retornar o código de saída 194 como parte do módulo de ação, a compilação será retomada na mesma etapa do módulo de ação que iniciou a reinicialização. Se você iniciar uma reinicialização sem o código de saída, o processo de compilação poderá falhar.

Exemplo de saída: antes da reinicialização (primeira vez no documento)

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

Exemplo de saída: após a reinicialização, (segunda vez no documento)

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

ExecuteBinary (Linux, Windows, macOS)

O módulo de ExecuteBinaryação permite que você execute arquivos binários com uma lista de argumentos de linha de comando.

O ExecuteBinarymódulo manipula as reinicializações do sistema se o arquivo binário sair com um código de saída 194 (Linux) ou `3010` (Windows). Quando iniciado, o aplicativo executa uma das seguintes ações:

- O aplicativo entrega o código de saída ao chamador se ele for executado pelo Systems Manager Agent. O Systems Manager Agent controla a reinicialização do sistema e executa a mesma etapa que começou a reinicialização, conforme descrito em Como reinicializar a instância gerenciada a partir de scripts.
- O aplicativo salva o `executionstate` atual, configura um gatilho de reinicialização para executar o aplicativo novamente e reinicia o sistema.

Após a reinicialização do sistema, o aplicativo executa a mesma etapa que iniciou a reinicialização. Se você precisar dessa funcionalidade, deverá escrever scripts idempotentes que possam lidar com várias invocações do mesmo comando shell.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório
`path`	O caminho para o arquivo binário para execução.	String	Sim
`arguments`	Contém uma lista de argumentos de linha de comando a serem usados ao executar o binário.	Lista de strings	Não

Exemplo de entrada: install.NET

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

Saída

Campo	Descrição	Tipo
`stdout`	Saída padrão da execução do comando.	string

Exemplo de saída

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

ExecuteDocument (Linux, Windows, macOS)

O módulo de ExecuteDocumentação adiciona suporte para documentos de componentes aninhados, executando vários documentos de componentes a partir de um documento. AWSTOE valida o documento que é passado no parâmetro de entrada em tempo de execução.

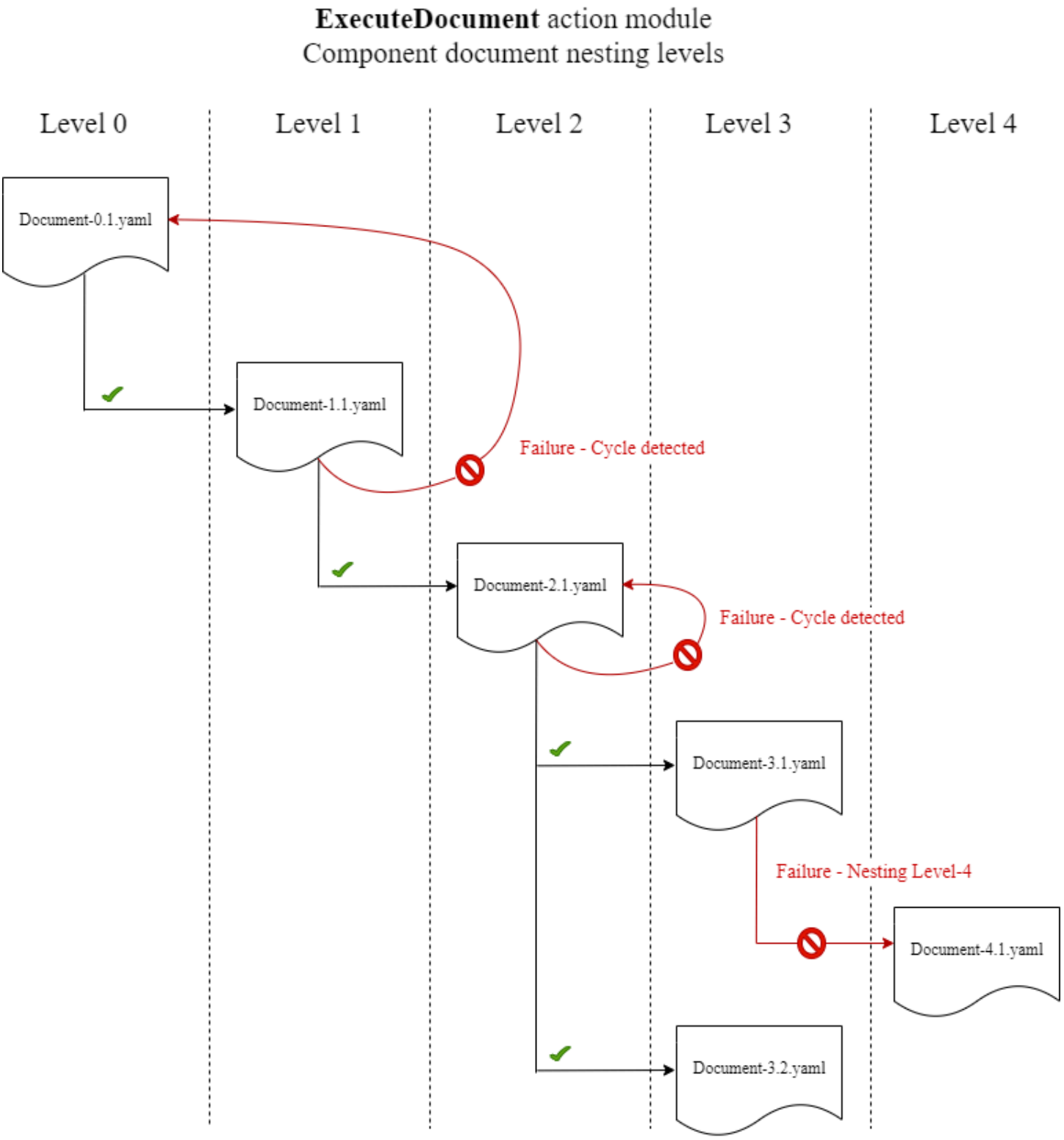
Restrições

- Esse módulo de ação é executado uma vez, sem a permissão de novas tentativas e sem a opção de definir limites de tempo limite. ExecuteDocumentdefine os seguintes valores padrão e retorna um erro se você tentar alterá-los.
  - `timeoutSeconds`: -1
  - `maxAttempts`: 1

  > **Note**
  >
  > Você pode deixar esses valores em branco e AWSTOE usar os valores padrão.

- O aninhamento de documentos é permitido, com até três níveis de profundidade, mas não mais do que isso. Três níveis de aninhamento se traduzem em quatro níveis de documentos, já que o nível superior não está aninhado. Nesse cenário, o documento de nível mais baixo não deve chamar nenhum outro documento.
- A execução cíclica de documentos de componentes não é permitida. Qualquer documento que se chame fora de uma estrutura em loop, ou que chame outro documento mais alto na cadeia de

execução atual, inicia um ciclo que pode resultar em um loop infinito. Quando o AWSTOE detecta uma execução cíclica, ele interrompe a execução e registra a falha.



Se um documento do componente tentar se executar sozinho ou executar qualquer um dos documentos do componente que estão mais acima na cadeia de execução atual, a execução falhará.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório
`document`	Caminho do documento do componente. Entre as opções válidas estão: • Caminhos de arquivo locais • S3 URIs • EC2 Versão de compilação do componente Image Builder ARNs	String	Sim
`document-s3-bucket-owner`	O ID da conta do proprietário do bucket do S3 onde são armazenados os documentos do componente. (Recomendado se você estiver usando o S3 URIs no documento do componente.)	String	Não
`phases`	Fases a serem executadas no	String	Não

Nome da chave	Descrição	Tipo	Obrigatório
	documento do componente, expressas como uma lista separada por vírgulas. Se nenhuma fase for especificada, todas as fases serão executadas.		
`parameters`	Parâmetros de entrada que são passados para o documento do componente no runtime como pares valores-chave.	Lista de mapas de parâmetros	Não

Entrada do mapa de parâmetros

Nome da chave	Descrição	Tipo	Obrigatório
`name`	O nome do parâmetro de entrada a ser passado para o documento do componente que o módulo de ExecuteDo cumentação está executando.	String	Sim
`value`	O valor do parâmetro de entrada.	String	Sim

Exemplos de entrada

Os exemplos a seguir mostram variações das entradas do documento do componente, dependendo do caminho de instalação.

Exemplo de entrada: caminho do documento local

```
# 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
```

Exemplo de entrada: URI do S3 como caminho do documento

```
# 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
```

Exemplo de entrada: ARN do componente EC2 Image Builder como um caminho de documento

```
# 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
```

Usando um ForEach loop para executar documentos

```
# 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
```

Usando um loop For para executar documentos

```
# 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
```

Saída

AWSTOE cria um arquivo de saída chamado `detailedoutput.json` toda vez que é executado. O arquivo contém detalhes sobre cada fase e etapa de cada documento do componente que é invocado durante a execução. Para o módulo de ExecuteDocumentação, você pode encontrar um breve resumo do tempo de execução no `outputs` campo e detalhes sobre as fases, etapas e documentos que ele é executado no`detailedOutput`.

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

O objeto de resumo de saída de cada documento do componente contém os seguintes detalhes, conforme mostrado aqui, com valores de amostra:

- executedStepCount“:1
- "executionId":"12345a67-89bc-01de-2f34-abcd56789012"
- “failedStepCount“:0
- "failureMessage":""
- “ignoredFailedStepContagem” :0
- "logUrl":""
- “status” :"sucesso”

Exemplo de saída

O exemplo a seguir mostra a saída do módulo de ExecuteDocumentação quando ocorre uma execução aninhada. Neste exemplo, o documento do componente `main.yaml` executa com êxito o documento do componente `Sample-1.yaml`.

```
{
    "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 (Windows)

O módulo de ExecutePowerShellação permite que você execute PowerShell scripts com códigos/comandos de shell embutidos. Este módulo oferece suporte à plataforma Windows e ao Windows PowerShell.

Todos os itens commands/instructions especificados no bloco de comandos são convertidos em um arquivo de script (por exemplo,`input.ps1`) e executados usando o WindowsPowerShell. O resultado da execução do arquivo shell é o código de saída.

O ExecutePowerShellmódulo manipula as reinicializações do sistema se o comando shell sair com um código de saída de. `3010` Quando iniciado, o aplicativo executa uma das seguintes ações:

- O aplicativo entrega o código de saída ao chamador se ele for executado pelo Systems Manager Agent. O Systems Manager Agent controla a reinicialização do sistema e executa a mesma etapa que iniciou a reinicialização, conforme descrito em Como reinicializar a instância gerenciada a partir de scripts.

- Salva o `executionstate` atual, configura um gatilho de reinicialização para executar novamente o aplicativo e reinicializa o sistema.

Após a reinicialização do sistema, o aplicativo executa a mesma etapa que iniciou a reinicialização. Se você precisar dessa funcionalidade, deverá escrever scripts idempotentes que possam lidar com várias invocações do mesmo comando shell.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório
`commands`	Contém uma lista de instruções ou comandos a serem executados de acordo com a PowerShel l sintaxe. O YAML de várias linhas é permitido.	Lista de strings	Sim. Deve especificar `commands` ou `file`, não ambos.
`file`	Contém o caminho para um arquivo de PowerShell script. PowerShell será executado nesse arquivo usando o argumento da linha de `-file` comando. O caminho deve apontar para um arquivo do `.ps1`.	String	Sim. Deve especificar `commands` ou `file`, não ambos.

Exemplo de entrada: antes e depois de uma reinicialização

```
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)
```

Saída

Campo	Descrição	Tipo
`stdout`	Saída padrão da execução do comando.	string

Se você executar uma reinicialização e retornar o código de saída `3010` como parte do módulo de ação, a compilação será retomada na mesma etapa do módulo de ação que iniciou a reinicialização. Se você executar uma reinicialização sem o código de saída, o processo de compilação poderá falhar.

Exemplo de saída: antes da reinicialização (primeira vez no documento)

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

Exemplo de saída: após a reinicialização, (segunda vez no documento)

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

Módulos de download e upload de arquivos

A seção a seguir contém detalhes dos módulos de ação que fazem upload ou download de arquivos.

Baixe e faça upload de módulos de ação

- S3Download (Linux, Windows, macOS)
- S3Upload (Linux, Windows, macOS)
- WebDownload (Linux, Windows, macOS)

S3Download (Linux, Windows, macOS)

Com o módulo de ação `S3Download`, você pode baixar um objeto do Amazon S3, ou um conjunto de objetos, para um arquivo ou pasta local que você especifica com o caminho `destination`. Se algum arquivo já existir no local especificado e o sinalizador do `overwrite` estiver definido como verdadeiro, o `S3Download` substituirá o arquivo.

Sua localização `source` pode apontar para um objeto específico no Amazon S3, ou você pode usar um prefixo de chave com um caractere curinga de asterisco (`*`) para baixar um conjunto de objetos que correspondam ao caminho do prefixo da chave. Quando você especifica um prefixo de chave na sua localização `source`, o módulo de ação `S3Download` baixa tudo o que corresponde ao prefixo (arquivos e pastas incluídos). Certifique-se de que o prefixo da chave termine com uma barra, seguida por um asterisco (`/*`), para que você baixe tudo que corresponda ao prefixo. Por exemplo: *`s3://my-bucket/my-folder/*`*.

Se a ação `S3Download` de um prefixo de chave especificado falhar durante um download, o conteúdo da pasta não voltará ao estado anterior à falha. A pasta de destino permanece como estava no momento da falha.

Casos de uso compatíveis

O módulo de ação `S3Download` é compatível com os seguintes casos de uso:

- O objeto Amazon S3 é baixado para uma pasta local, conforme especificado no caminho de download.

- Os objetos do Amazon S3 (com um prefixo de chave no caminho do arquivo do Amazon S3) são baixados para a pasta local especificada, que copia recursivamente todos os objetos do Amazon S3 que correspondem ao prefixo da chave na pasta local.

Requisitos do IAM

O perfil do IAM que você associa ao seu perfil de instância precisa ter permissões para executar o módulo de ação `S3Download`. As seguintes políticas do IAM devem ser anexadas ao perfil do IAM associada ao perfil de instância:

- Arquivo único: `s3:GetObject` contra o bucket/object (por exemplo,`arn:aws:s3:::BucketName/*`).
- Vários arquivos: `s3:ListBucket` contra o bucket/object (por exemplo,`arn:aws:s3:::BucketName`) e `s3:GetObject` contra o bucket/object (por exemplo,`arn:aws:s3:::BucketName/*`).

Entrada

Chave	Descrição	Tipo	Obrigatório	Padrão
`source`	O bucket do Amazon S3 que é a fonte para seu download. Você pode especificar um caminho para um objeto específico ou usar um prefixo de chave que termine com uma barra, seguida por um caractere curinga de	String	Sim	N/D

Chave	Descrição	Tipo	Obrigatório	Padrão
	asterisco (`/*`), para baixar um conjunto de objetos que correspondam ao prefixo da chave.			
`destination`	O caminho local em que os objetos do Amazon S3 são baixados. Para baixar um único arquivo, especifique o nome do arquivo como parte do caminho. Por exemplo, *`./myfolder/package.zip`*	String	Sim	N/D

Chave	Descrição	Tipo	Obrigatório	Padrão
`expectedBucketOwner`	ID da conta do proprietário esperada do bucket fornecido no caminho `source`. Recomendamos que você verifique a propriedade do bucket Amazon S3 especificado na fonte.	String	Não	N/D

Chave	Descrição	Tipo	Obrigatório	Padrão
`overwrite`	Quando definido como verdadeiro, se um arquivo com o mesmo nome já existir na pasta de destino no caminho local especificado, o arquivo de download substituirá o arquivo local. Quando definido como false, o arquivo existente no sistema local é protegido contra a substituição e o módulo de ação falha com um erro de download. Por exemplo, `Error: S3Download: File already exists and "overwrite" property for "destinat`	Booleano	Não	true

Chave	Descrição	Tipo	Obrigatório	Padrão
	ion" file is set to false. Cannot download. .			

> **Note**
>
> Nos exemplos a seguir, o caminho da pasta do Windows pode ser substituído por um caminho do Linux. Por exemplo, *C:\myfolder\package.zip* pode ser substituído por */myfolder/package.zip*.

Exemplo de entrada: copiar um objeto do Amazon S3 para um arquivo local

O exemplo a seguir mostra como copiar um objeto do Amazon S3 para um arquivo local.

```
- 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
```

Exemplo de entrada: copie todos os objetos do Amazon S3 para um bucket do Amazon S3 com prefixo de chave em uma pasta local

O exemplo a seguir mostra como copiar todos os objetos do Amazon S3 para um bucket do Amazon S3 para uma pasta local. O Amazon S3 não tem o conceito de pasta, portanto, todos os objetos

que correspondem ao prefixo da chave são copiados. O número máximo de objetos que podem ser baixados é 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
```

Saída

Nenhum.

S3Upload (Linux, Windows, macOS)

Com o módulo de ação S3Upload, você pode carregar um arquivo de um arquivo ou pasta de origem para um local do Amazon S3. Você pode usar um curinga (*) no caminho especificado para o local de origem para carregar todos os arquivos cujo caminho corresponda ao padrão curinga.

Se a ação recursiva do S3Upload falhar, todos os arquivos que já foram carregados permanecerão no bucket Amazon S3 de destino.

Casos de uso compatíveis

- Arquivo local para o objeto Amazon S3.
- Arquivos locais na pasta (com curinga) para o prefixo de chave do Amazon S3.
- Copie a pasta local (`recurse` deve ter sido definido como `true`) para o prefixo de chave do Amazon S3.

Requisitos do IAM

O perfil do IAM que você associa ao seu perfil de instância precisa ter permissões para executar o módulo de ação `S3Upload`. As seguintes políticas do IAM devem ser anexadas ao perfil do IAM associada ao perfil de instância. A política deve conceder permissões de `s3:PutObject` ao bucket de destino do Amazon S3. Por exemplo, `arn:aws:s3:::BucketName/*`.

Entrada

Chave	Descrição	Tipo	Obrigatório	Padrão
`source`	O caminho local de origem da fonte files/ folders . O `source` suporta um caractere curinga de asterisco (*).	String	Sim	N/D
`destination`	O caminho para o bucket de destino do Amazon S3 onde os arquivos/ pastas de origem são carregados.	String	Sim	N/D
`recurse`	Quando definido como `true`, executa o S3Upload recursivamente.	String	Não	`false`
`expectedBucketOwner`	O ID esperado da conta do proprietário	String	Não	N/D

Chave	Descrição	Tipo	Obrigatório	Padrão
	para o bucket do Amazon S3 especificado no caminho de destino. Recomenda mos que você verifique a propriedade do bucket Amazon S3 especificado no destino.			

Exemplo de entrada: copiar um arquivo local para um objeto do Amazon S3

O exemplo a seguir mostra como copiar um arquivo local para um objeto do 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
```

Exemplo de entrada: copie todos os arquivos em uma pasta local para um bucket do Amazon S3 com prefixo de chave

O exemplo a seguir mostra como copiar todos os arquivos na pasta local para um bucket do Amazon S3 com prefixo de chave. Este exemplo não copia subpastas ou seu conteúdo porque o `recurse` não está especificado e o padrão é `false`.

```
- name: MyS3UploadMultipleFiles
  action: S3Upload
  onFailure: Abort
```

```
    maxAttempts: 3
    inputs:
      - source: C:\myfolder\*
        destination: s3://amzn-s3-demo-destination-bucket/path/to/
        expectedBucketOwner: 123456789022
```

Exemplo de entrada: copiar recursivamente todos os arquivos e pastas de uma pasta local para um bucket do Amazon S3

O exemplo a seguir mostra como copiar recursivamente todos os arquivos e pastas de uma pasta local para um bucket do Amazon S3 com o prefixo de chaves.

```
- 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
```

Saída

Nenhum.

WebDownload (Linux, Windows, macOS)

O módulo de WebDownloadação permite que você baixe arquivos e recursos de um local remoto pelo HTTP/HTTPS protocolo (HTTPS é recomendado). Não há limites para o número ou o tamanho dos downloads. Este módulo lida com a lógica de repetição e recuo exponencial.

Cada operação de download recebe no máximo 5 tentativas de sucesso, de acordo com as entradas do usuário. Essas tentativas diferem das especificadas no campo `maxAttempts` do documento `steps`, que estão relacionadas às falhas do módulo de ação.

Esse módulo de ação manipula implicitamente os redirecionamentos. Todos os códigos de status HTTP, exceto `200`, resultam em um erro.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório	Padrão
`source`	O HTTP/HTTPS URL válido (HTTPS é recomendado), que segue o padrão RFC 3986. Expressões de encadeamento são permitidas.	String	Sim	N/D
`destination`	Um caminho absoluto ou relativo de arquivo ou pasta no sistema local. Os caminhos das pastas devem terminar com `/`. Se não terminarem com `/`, serão tratados como caminhos de arquivo. O módulo cria qualquer arquivo ou pasta necessário para downloads bem-sucedidos. Expressões de encadeamento são permitidas.	String	Sim	N/D

Nome da chave	Descrição	Tipo	Obrigatório	Padrão
`overwrite`	Quando ativado, substitui todos os arquivos existentes no sistema local pelo arquivo ou recurso baixado. Quando não ativado, os arquivos existentes no sistema local não são sobrescritos e o módulo de ação falha com um erro. Quando a substituição está ativada e a soma de verificação e o algoritmo são especific ados, o módulo de ação baixa o arquivo somente se a soma de verificação e o hash de qualquer arquivo preexistente não corresponderem.	Booleano	Não	`true`

Nome da chave	Descrição	Tipo	Obrigatório	Padrão
`checksum`	Quando você especifica a soma de verificação, ela é comparada com o hash do arquivo baixado que é gerado com o algoritmo fornecido. Para que a verificação do arquivo seja ativada, a soma de verificação e o algoritmo devem ser fornecidos. Expressões de encadeamento são permitidas.	String	Não	N/D

Nome da chave	Descrição	Tipo	Obrigatório	Padrão
`algorithm`	O algoritmo usado para calcular a soma de verificação. As opções são MD5 SHA1 SHA256,, SHA512 e. Para que a verificação do arquivo seja ativada, a soma de verificação e o algoritmo devem ser fornecidos. Expressões de encadeamento são permitidas.	String	Não	N/D
`ignoreCertificateErrors`	A validação do certificado SSL é ignorada quando ativada.	Booleano	Não	`false`

Saída

Nome da chave	Descrição	Tipo
`destination`	String delimitada por caracteres de nova linha que	String

Nome da chave	Descrição	Tipo				
	especifica o caminho de destino em que os arquivos ou recursos baixados são armazenados.					

Exemplo de entrada: baixar arquivo remoto no destino local

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

Saída:

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

Exemplo de entrada: baixe mais de um arquivo remoto em mais de um destino local

```
  - 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/
```

Saída:

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

Exemplo de entrada: baixe um arquivo remoto sem sobrescrever o destino local e baixe outro arquivo remoto com verificação de arquivo

```
  - 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
```

Saída:

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

Exemplo de entrada: baixe o arquivo remoto e ignore a validação da certificação SSL

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

Saída:

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

Módulos de operações do sistema de arquivos

A seção a seguir contém detalhes dos módulos de ação que executam operações do sistema de arquivos.

Módulos de ação de operação do sistema de arquivos

- 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)

O módulo de AppendFileação adiciona conteúdo especificado ao conteúdo preexistente de um arquivo.

Se o valor da codificação do arquivo for diferente do valor padrão de codificação (`utf-8`), você poderá especificar o valor da codificação do arquivo usando a opção `encoding`. Por padrão, presume-se que `utf-16` e `utf-32` usam a codificação little-endian.

O módulo de ação retorna um erro quando ocorre o seguinte:

- O arquivo especificado não existe no runtime.
- Você não tem permissões de gravação para modificar o conteúdo do arquivo.
- O módulo encontra um erro durante a operação do arquivo.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
`path`	O caminho do arquivo.	String	Sim	N/D	N/D	Sim
`content`	O conteúdo a ser anexado ao arquivo.	String	Não	String vazia	N/D	Sim
`encoding`	O padrão de codificação.	String	Não	`utf8`	`utf8`, `utf-8`, `utf16`,`utf-` `utf16-LE`, `utf-16-LE` `utf16-BE`, `utf-16-BE` , `utf32`, `utf-32`,	Sim

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
					`utf32-LE`,`utf-32-LE` , `utf32-BE` e `utf-32-BE` . O valor da opção de codificação não diferencia maiúsculas de minúsculas.	

Exemplo de entrada: anexar arquivo sem codificação (Linux)

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

Exemplo de entrada: anexar arquivo sem codificação (Windows)

```
- name: AppendingFileWithOutEncodingWindows
  action: AppendFile
  inputs:
    - path: C:\MyFolder\MyFile.txt
```

```
        content: "The string to be appended to the file"
```

Exemplo de entrada: anexar arquivo com codificação (Linux)

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

Exemplo de entrada: anexar arquivo com codificação (Windows)

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

Exemplo de entrada: anexar arquivo com string vazio (Linux)

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

Exemplo de entrada: anexar arquivo com string vazio (Windows)

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

Saída

Nenhum.

CopyFile (Linux, Windows, macOS)

O módulo de CopyFileação copia arquivos da fonte especificada para o destino especificado. Por padrão, o módulo cria recursivamente a pasta de destino se ela não existir no runtime.

Se um arquivo com o nome especificado já existir na pasta especificada, o módulo de ação, por padrão, substituirá o arquivo existente. Você pode substituir esse comportamento padrão configurando a opção de substituição como `false`. Quando a opção de substituição estiver definida como `false`, e um arquivo já estiver no local especificado com o nome especificado, o módulo de ação retornará um erro. Essa opção funciona da mesma forma que o comando `cp` no Linux, que substitui por padrão.

O nome do arquivo de origem pode incluir um curinga (`*`). Caracteres curinga são aceitos somente após o último separador de caminho de arquivo (`/` ou `\`). Se caracteres curinga forem incluídos no nome do arquivo de origem, todos os arquivos que corresponderem ao curinga serão copiados para a pasta de destino. Se você quiser mover mais de um arquivo usando um caractere curinga, a entrada para a opção `destination` deverá terminar com um separador de caminho de arquivo (`/` ou `\`), que indica que a entrada de destino é uma pasta.

Se o nome do arquivo de destino for diferente do nome do arquivo de origem, você poderá especificar o nome do arquivo de destino usando a opção `destination`. Se você não especificar um nome de arquivo de destino, o nome do arquivo de origem será usado para criar o arquivo de destino. Qualquer texto que siga o último separador de caminho de arquivo (`/` ou `\`) será tratado como o nome do arquivo. Se você quiser usar o mesmo nome de arquivo do arquivo de origem, a entrada da opção `destination` deverá terminar com um separador de caminho de arquivo (`/` ou `\`).

O módulo de ação retorna um erro quando ocorre o seguinte:

- Você não tem a permissão para criar um arquivo na pasta especificada.
- Os arquivos de origem não existem no runtime.
- Já existe uma pasta com o nome de arquivo especificado e a `overwrite`opção está definida como `false`.
- O módulo de ação encontra um erro ao executar a operação.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
`source`	O caminho do arquivo de origem.	String	Sim	N/D	N/D	Sim
`destination`	O caminho do arquivo de destino.	String	Sim	N/D	N/D	Sim
`overwrite`	Quando definido como false, os arquivos de destino não serão substituídos quando já houver um arquivo no local especificado com o nome especificado.	Booleano	Não	`true`	N/D	Sim

Exemplo de entrada: copiar um arquivo (Linux)

```
- name: CopyingAFileLinux
  action: CopyFile
```

```
  inputs:
    - source: /Sample/MyFolder/Sample.txt
      destination: /MyFolder/destinationFile.txt
```

Exemplo de entrada: copiar um arquivo (Windows)

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

Exemplo de entrada: copiar um arquivo usando o nome do arquivo de origem (Linux)

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

Exemplo de entrada: copiar um arquivo usando o nome do arquivo de origem (Windows)

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

Exemplo de entrada: copiar um arquivo usando o caractere curinga (Linux)

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

Exemplo de entrada: copiar um arquivo usando o caractere curinga (Windows)

```
- name: CopyingFilesWithWildCardWindows
  action: CopyFile
  inputs:
```

```
    - source: C:\Sample\MyFolder\Sample*
      destination: C:\MyFolder\
```

Exemplo de entrada: copiar um arquivo sem substituir (Linux)

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

Exemplo de entrada: copiar um arquivo sem substituir (Windows)

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

Saída

Nenhum.

CopyFolder (Linux, Windows, macOS)

O módulo de CopyFolderação copia uma pasta da origem especificada para o destino especificado. A entrada para a opção `source` é a pasta a ser copiada, e a entrada para a opção `destination` é a pasta em que o conteúdo da pasta de origem é copiado. Por padrão, o módulo cria recursivamente a pasta de destino se ela não existir no runtime.

Se uma pasta com o nome especificado já existir na pasta especificada, o módulo de ação, por padrão, substituirá a pasta existente. Você pode substituir esse comportamento padrão configurando a opção de substituição como `false`. Quando a opção de substituição estiver definida como `false` e já houver uma pasta no local especificado com o nome especificado, o módulo de ação retornará um erro.

O nome da pasta de origem pode incluir um curinga (*). Caracteres curinga são aceitos somente após o último separador de caminho de arquivo (/ ou \). Se caracteres curinga forem incluídos no nome da pasta de origem, todas as pastas que corresponderem ao curinga serão copiados para a

pasta de destino. Se você quiser copiar mais de uma pasta usando um caractere curinga, a entrada para a opção `destination` deverá terminar com um separador de caminho de arquivo (/ ou \), que indica que a entrada de destino é uma pasta.

Se o nome da pasta de destino for diferente do nome da pasta de origem, você poderá especificar o nome do arquivo da pasta usando a opção `destination`. Se você não especificar um nome da pasta de destino, o nome da pasta de origem será usado para criar a pasta de destino. Qualquer texto que siga o último separador de caminho de arquivo (/ ou \) será tratado como o nome da pasta. Se você quiser usar o mesmo nome de pasta da pasta de origem, a entrada da opção `destination` deverá terminar com um separador de caminho de arquivo (/ ou \).

O módulo de ação retorna um erro quando ocorre o seguinte:

- Você não tem a permissão para criar uma pasta na pasta especificada.
- As pastas de origem não existem no runtime.
- Já existe uma pasta com o nome de pasta especificado e a `overwrite`opção está definida como `false`.
- O módulo de ação encontra um erro ao executar a operação.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
`source`	O caminho da pasta de origem.	String	Sim	N/D	N/D	Sim
`destination`	O caminho da pasta de destino.	String	Sim	N/D	N/D	Sim
`overwrite`	Quando definido como false,	Booleano	Não	`true`	N/D	Sim

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
	as pastas de destino não serão substituídas quando já houver uma pasta no local especificado com o nome especificado.					

Exemplo de entrada: copiar uma pasta (Linux)

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

Exemplo de entrada: copiar uma pasta (Windows)

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

Exemplo de entrada: copiar uma pasta usando o nome da pasta de origem (Linux)

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

Exemplo de entrada: copiar uma pasta usando o nome da pasta de origem (Windows)

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

Exemplo de entrada: copiar uma pasta usando o caractere curinga (Linux)

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

Exemplo de entrada: copiar uma pasta usando o caractere curinga (Windows)

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

Exemplo de entrada: copiar uma pasta sem substituir (Linux)

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

Exemplo de entrada: copiar uma pasta sem substituir (Windows)

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

Saída

Nenhum.

CreateFile (Linux, Windows, macOS)

O módulo de CreateFileação cria um arquivo em um local especificado. Por padrão, se necessário, o módulo também cria as pastas principais recursivamente.

Se um arquivo já existir na pasta especificada, o módulo de ação, por padrão, truncará ou substituirá o arquivo existente. Você pode substituir esse comportamento padrão configurando a opção de substituição como `false`. Quando a opção de substituição estiver definida como `false`, e um arquivo já estiver no local especificado com o nome especificado, o módulo de ação retornará um erro.

Se o valor da codificação do arquivo for diferente do valor padrão de codificação (`utf-8`), você poderá especificar o valor da codificação do arquivo usando a opção `encoding`. Por padrão, presume-se que `utf-16` e `utf-32` usam a codificação little-endian.

`owner`,`group` e `permissions` são entradas opcionais. A entrada para `permissions` deve ser um valor de string. Os arquivos são criados com valores padrão quando não são fornecidos. Essas opções não são suportadas nas plataformas Windows. Esse módulo de ação valida e retorna um erro se as opções `owner`, `group` e `permissions` forem usadas em plataformas Windows.

Esse módulo de ação pode criar um arquivo com permissões definidas pelo valor padrão `umask` do sistema operacional. Você deve definir o valor `umask` se quiser substituir o valor padrão.

O módulo de ação retorna um erro quando ocorre o seguinte:

- Você não tem a permissão para criar um arquivo ou uma pasta na pasta parent especificada.
- O módulo de ação encontra um erro ao executar a operação.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
`path`	O caminho do arquivo.	String	Sim	N/D	N/D	Sim
`content`	O conteúdo de texto do arquivo.	String	Não	N/D	N/D	Sim
`encoding`	O padrão de codificação.	String	Não	`utf8`	`utf8`, `utf-8`, `utf16`,`utf-16`, `utf16-LE`, `utf-16-LE` `utf16-BE`, `utf-16-BE` , `utf32`, `utf-32`, `utf32-LE`,`utf-32-LE` , `utf32-BE` e `utf-32-BE` . O valor da opção de codificação não	Sim

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
					diferencia maiúsculas de minúsculas.	
`owner`	O nome do usuário ou ID.	String	Não	N/D	N/D	Não compatível com Windows.
`group`	O nome ou ID do grupo.	String	Não	O usuário atual.	N/D	Não compatível com Windows.
`permissions`	As permissões de arquivo.	String	Não	`0666`	N/D	Não compatível com Windows.

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
`overwrite`	Se o nome do arquivo especificado já existir, definir esse valor como `false` evita que o arquivo seja truncado ou substituído por padrão.	Booleano	Não	`true`	N/D	Sim

Exemplo de entrada: criar um arquivo sem substituir (Linux)

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

Exemplo de entrada: criar um arquivo sem substituir (Windows)

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

```
        overwrite: false
```

Exemplo de entrada: criar um arquivo com propriedades do arquivo

```
- 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
```

Exemplo de entrada: criar um arquivo sem propriedades do arquivo

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

Exemplo de entrada: criar um arquivo vazio para ignorar uma seção no script de limpeza do Linux

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

Para obter mais informações, consulte Substitua o script de limpeza do Linux.

Saída

Nenhum.

CreateFolder (Linux, Windows, macOS)

O módulo de CreateFolderação cria uma pasta em um local especificado. Por padrão, se necessário, o módulo também cria as pastas principais recursivamente.

Se a pasta já existir na pasta especificada, o módulo de ação, por padrão, truncará ou substituirá a pasta existente. Você pode substituir esse comportamento padrão configurando a opção de substituição como `false`. Quando a opção de substituição estiver definida como `false` e já houver uma pasta no local especificado com o nome especificado, o módulo de ação retornará um erro.

`owner`,`group` e `permissions` são entradas opcionais. A entrada para `permissions` deve ser um valor de string. Essas opções não são suportadas nas plataformas Windows. Esse módulo de ação valida e retorna um erro se as opções `owner`, `group` e `permissions` forem usadas em plataformas Windows.

Esse módulo de ação pode criar uma pasta com permissões definidas pelo valor padrão `umask` do sistema operacional. Você deve definir o valor `umask` se quiser substituir o valor padrão.

O módulo de ação retorna um erro quando ocorre o seguinte:

- Você não tem permissão para criar uma pasta no local especificado.
- O módulo de ação encontra um erro ao executar a operação.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
`path`	O caminho da pasta.	String	Sim	N/D	N/D	Sim
`owner`	O nome do usuário ou ID.	String	Não	O usuário atual.	N/D	Não compatível com Windows.

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
`group`	O nome ou ID do grupo.	String	Não	O grupo do usuário atual.	N/D	Não compatível com Windows.
`permissions`	As permissões da pasta.	String	Não	`0777`	N/D	Não compatível com Windows.
`overwrite`	Se o nome do arquivo especificado já existir, definir esse valor como `false` evita que o arquivo seja truncado ou substituído por padrão.	Booleano	Não	`true`	N/D	Sim

Exemplo de entrada: criar uma pasta (Linux)

```
- name: CreatingFolderLinux
  action: CreateFolder
  inputs:
```

```
      - path: /Sample/MyFolder/
```

Exemplo de entrada: criar uma pasta (Windows)

```
- name: CreatingFolderWindows
  action: CreateFolder
  inputs:
    - path: C:\MyFolder
```

Exemplo de entrada: criar uma pasta especificando as propriedades da pasta

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

Exemplo de entrada: criar uma pasta que substitua a pasta existente, se houver uma.

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

Saída

Nenhum.

CreateSymlink (Linux, Windows, macOS)

O módulo de CreateSymlinkação cria links simbólicos ou arquivos que contêm uma referência a outro arquivo. Este módulo não é compatível com plataformas Windows.

A entrada para as opções `path` e `target` pode ser um caminho absoluto ou relativo. Se a entrada da opção `path` for um caminho relativo, ela será substituída pelo caminho absoluto quando o link for criado.

Por padrão, quando um link com o nome especificado já existe na pasta especificada, o módulo de ação retorna um erro. Você pode substituir esse comportamento padrão configurando a opção `force` como `true`. Quando a opção `force` estiver definida como `true`, o módulo substituirá o link existente.

Se uma pasta principal não existir, o módulo de ação cria a pasta recursivamente, por padrão.

O módulo de ação retorna um erro quando ocorre o seguinte:

- O arquivo de destino não existe no runtime.
- Já existe um arquivo de link não simbólico com o nome especificado.
- O módulo de ação encontra um erro ao executar a operação.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
`path`	O caminho do arquivo.	String	Sim	N/D	N/D	Não compatível com Windows.
`target`	O caminho do arquivo de destino para o qual o link simbólico aponta.	String	Sim	N/D	N/D	Não compatível com Windows.
`force`	Força a criação de um link quando já	Booleano	Não	`false`	N/D	Não compatível com Windows.

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
	existe um link com o mesmo nome.					

Exemplo de entrada: criar um link simbólico que força a criação de um link

```
- name: CreatingSymbolicLinkWithForce
  action: CreateSymlink
  inputs:
    - path: /Folder2/Symboliclink.txt
      target: /Folder/Sample.txt
      force: true
```

Exemplo de entrada: criar um link simbólico que não força a criação de um link

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

Saída

Nenhum.

DeleteFile (Linux, Windows, macOS)

O módulo de DeleteFileação exclui um arquivo ou arquivos em um local especificado.

A entrada de `path` deve ser um caminho de arquivo válido ou um caminho de arquivo com um caractere curinga (*) no nome do arquivo. Quando caracteres curinga são especificados no nome do arquivo, todos os arquivos na mesma pasta que correspondam ao curinga são excluídos.

O módulo de ação retorna um erro quando ocorre o seguinte:

- Você não tem a permissão para realizar a operação solicitada.
- O módulo de ação encontra um erro ao executar a operação.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
`path`	O caminho do arquivo.	String	Sim	N/D	N/D	Sim

Exemplo de entrada: excluir um único arquivo (Linux)

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

Exemplo de entrada: excluir um único arquivo (Windows)

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

Exemplo de entrada: excluir um arquivo que termina com “log” (Linux)

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

Exemplo de entrada: excluir um arquivo que termina com “log” (Windows)

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

Exemplo de entrada: excluir todos os arquivos em uma pasta especificada (Linux)

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

Exemplo de entrada: excluir todos os arquivos em uma pasta especificada (Windows)

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

Saída

Nenhum.

DeleteFolder (Linux, Windows, macOS)

O módulo de DeleteFolderação exclui pastas.

Se a pasta não estiver vazia, você deverá definir a `force` opção `true` para remover a pasta e seu conteúdo. Se você não definir a opção `force` como `true` e a pasta que você está tentando excluir não estiver vazia, o módulo de ação retornará um erro. O valor padrão da opção `force` é `false`.

O módulo de ação retorna um erro quando ocorre o seguinte:

- Você não tem a permissão para realizar a operação solicitada.
- O módulo de ação encontra um erro ao executar a operação.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
`path`	O caminho da pasta.	String	Sim	N/D	N/D	Sim
`force`	Remove a pasta, esteja ela vazia ou não.	Booleano	Não	`false`	N/D	Sim

Exemplo de entrada: excluir uma pasta que não esteja vazia usando a **`force`** opção (Linux)

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

Exemplo de entrada: excluir uma pasta que não esteja vazia usando a **`force`** opção (Windows)

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

Exemplo de entrada: excluir uma pasta (Linux)

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

Exemplo de entrada: excluir uma pasta (Windows)

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

Saída

Nenhum.

ListFiles (Linux, Windows, macOS)

O módulo de ListFilesação lista os arquivos em uma pasta especificada. Quando a opção recursiva está definida como `true`, ela lista os arquivos nas subpastas. Por padrão, este módulo não lista arquivos em subpastas.

Para listar todos os arquivos com nomes que correspondam a um padrão especificado, use a opção `fileNamePattern` para fornecer o padrão. A opção `fileNamePattern` aceita o valor curinga (*). Quando o `fileNamePattern` é fornecido, todos os arquivos que correspondem ao formato de nome de arquivo especificado são retornados.

O módulo de ação retorna um erro quando ocorre o seguinte:

- A pasta especificada não existe no runtime.
- Você não tem a permissão para criar um arquivo ou uma pasta na pasta parent especificada.
- O módulo de ação encontra um erro ao executar a operação.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
`path`	O caminho da pasta.	String	Sim	N/D	N/D	Sim

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
`fileNamePattern`	O padrão a ser correspondido ao listar todos os arquivos com nomes que correspondam ao padrão.	String	Não	N/D	N/D	Sim
`recursive`	Lista os arquivos na pasta recursivamente.	Booleano	Não	`false`	N/D	Sim

Exemplo de entrada: listar todos os arquivos em uma pasta especificada (Linux)

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

Exemplo de entrada: listar todos os arquivos em uma pasta especificada (Windows)

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

Exemplo de entrada: listar arquivos que terminam com “log” (Linux)

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

Exemplo de entrada: listar arquivos que terminam com “log” (Windows)

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

Exemplo de entrada: listar arquivos recursivamente

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

Saída

Nome da chave	Descrição	Tipo
`files`	A lista de arquivos.	String

Exemplo de saída

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

MoveFile (Linux, Windows, macOS)

O módulo de MoveFileação move arquivos da fonte especificada para o destino especificado.

Se um arquivo já existir na pasta especificada, o módulo de ação, por padrão, substituirá o arquivo existente. Você pode substituir esse comportamento padrão configurando a opção de substituição como `false`. Quando a opção de substituição estiver definida como `false`, e um arquivo já estiver no local especificado com o nome especificado, o módulo de ação retornará um erro. Essa opção funciona da mesma forma que o comando `mv` no Linux, que substitui por padrão.

O nome do arquivo de origem pode incluir um curinga (`*`). Caracteres curinga são aceitos somente após o último separador de caminho de arquivo (`/` ou `\`). Se caracteres curinga forem incluídos no nome do arquivo de origem, todos os arquivos que corresponderem ao curinga serão copiados para a pasta de destino. Se você quiser mover mais de um arquivo usando um caractere curinga, a entrada para a opção `destination` deverá terminar com um separador de caminho de arquivo (`/` ou `\`), que indica que a entrada de destino é uma pasta.

Se o nome do arquivo de destino for diferente do nome do arquivo de origem, você poderá especificar o nome do arquivo de destino usando a opção `destination`. Se você não especificar um nome de arquivo de destino, o nome do arquivo de origem será usado para criar o arquivo de destino. Qualquer texto que siga o último separador de caminho de arquivo (`/` ou `\`) será tratado como o nome do arquivo. Se você quiser usar o mesmo nome de arquivo do arquivo de origem, a entrada da opção `destination` deverá terminar com um separador de caminho de arquivo (`/` ou `\`).

O módulo de ação retorna um erro quando ocorre o seguinte:

- Você não tem a permissão para criar um arquivo na pasta especificada.
- Os arquivos de origem não existem no runtime.
- Já existe uma pasta com o nome de arquivo especificado e a `overwrite`opção está definida como `false`.
- O módulo de ação encontra um erro ao executar a operação.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
`source`	O caminho do arquivo de origem.	String	Sim	N/D	N/D	Sim
`destination`	O caminho do arquivo de destino.	String	Sim	N/D	N/D	Sim
`overwrite`	Quando definido como false, os arquivos de destino não serão substituídos quando já houver um arquivo no local especificado com o nome especificado.	Booleano	Não	`true`	N/D	Sim

Exemplo de entrada: mover um arquivo (Linux)

```
- name: MovingAFileLinux
  action: MoveFile
```

```
  inputs:
    - source: /Sample/MyFolder/Sample.txt
      destination: /MyFolder/destinationFile.txt
```

Exemplo de entrada: mover um arquivo (Windows)

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

Exemplo de entrada: mover um arquivo usando o nome do arquivo de origem (Linux)

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

Exemplo de entrada: mover um arquivo usando o nome do arquivo de origem (Windows)

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

Exemplo de entrada: mover um arquivo usando o caractere curinga (Linux)

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

Exemplo de entrada: mover um arquivo usando o caractere curinga (Windows)

```
- name: MovingFilesWithWildCardWindows
  action: MoveFile
  inputs:
```

```
      - source: C:\Sample\MyFolder\Sample*
        destination: C:\MyFolder
```

Exemplo de entrada: mover um arquivo sem substituir (Linux)

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

Exemplo de entrada: mover um arquivo sem substituir (Windows)

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

Saída

Nenhum.

MoveFolder (Linux, Windows, macOS)

O módulo de MoveFolderação move as pastas da origem especificada para o destino especificado. A entrada para a opção `source` é a pasta a ser movida, e a entrada para a opção `destination` é a pasta para a qual o conteúdo das pastas de origem é movido.

Se a pasta parent de destino ou a entrada para a opção `destination` não existirem no runtime, o comportamento padrão do módulo é criar a pasta no destino especificado recursivamente.

Se uma pasta igual à pasta de origem já existir na pasta de destino, o módulo de ação, por padrão, substituirá a pasta existente. Você pode substituir esse comportamento padrão configurando a opção de substituição como `false`. Quando a opção de substituição estiver definida como `false` e já houver uma pasta no local especificado com o nome especificado, o módulo de ação retornará um erro.

O nome da pasta de origem pode incluir um curinga (*). Caracteres curinga são aceitos somente após o último separador de caminho de arquivo (/ ou \). Se caracteres curinga forem incluídos no

nome da pasta de origem, todas as pastas que corresponderem ao curinga serão copiados para a pasta de destino. Se você quiser mover mais de uma pasta usando um caractere curinga, a entrada para a opção `destination` deverá terminar com um separador de caminho de arquivo (/ ou \), que indica que a entrada de destino é uma pasta.

Se o nome da pasta de destino for diferente do nome da pasta de origem, você poderá especificar o nome do arquivo da pasta usando a opção `destination`. Se você não especificar um nome da pasta de destino, o nome da pasta de origem será usado para criar a pasta de destino. Qualquer texto que siga o último separador de caminho de arquivo (/ ou \) será tratado como o nome da pasta. Se você quiser usar o mesmo nome de pasta da pasta de origem, a entrada da opção `destination` deverá terminar com um separador de caminho de arquivo (/ ou \).

O módulo de ação retorna um erro quando ocorre o seguinte:

- Você não tem a permissão para criar uma pasta na pasta de destino.
- As pastas de origem não existem no runtime.
- Já existe uma pasta com o nome especificado e a opção `overwrite` está definida como `false`.
- O módulo de ação encontra um erro ao executar a operação.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
`source`	O caminho da pasta de origem.	String	Sim	N/D	N/D	Sim
`destination`	O caminho da pasta de destino.	String	Sim	N/D	N/D	Sim
`overwrite`	Quando definido como false,	Booleano	Não	`true`	N/D	Sim

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
	as pastas de destino não serão substituídas quando já houver uma pasta no local especificado com o nome especificado.					

Exemplo de entrada: mover uma pasta (Linux)

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

Exemplo de entrada: mover uma pasta (Windows)

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

Exemplo de entrada: mover uma pasta usando o nome da pasta de origem (Linux)

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

Exemplo de entrada: mover uma pasta usando o nome da pasta de origem (Windows)

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

Exemplo de entrada: mover uma pasta usando o caractere curinga (Linux)

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

Exemplo de entrada: mover uma pasta usando o caractere curinga (Windows)

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

Exemplo de entrada: mover uma pasta sem substituir (Linux)

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

Exemplo de entrada: mover uma pasta sem substituir (Windows)

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

Saída

Nenhum.

ReadFile (Linux, Windows, macOS)

O módulo de ReadFileação lê o conteúdo de um arquivo de texto do tipo string. Esse módulo pode ser usado para ler o conteúdo de um arquivo para uso em etapas subsequentes por meio de encadeamento ou para ler dados no arquivo do `console.log`. Se o caminho especificado for um link simbólico, este módulo retornará o conteúdo do arquivo de destino. Este módulo só oferece suporte a arquivos de texto.

Se o valor da codificação do arquivo for diferente do valor padrão de codificação (`utf-8`), você poderá especificar o valor da codificação do arquivo usando a opção `encoding`. Por padrão, presume-se que `utf-16` e `utf-32` usam a codificação little-endian.

Por padrão, esse módulo não pode imprimir o conteúdo do arquivo no arquivo do `console.log`. Você pode substituir essa configuração definindo a propriedade `printFileContent` como `true`.

Esse módulo pode retornar somente o conteúdo de um arquivo. Ele não pode analisar arquivos, como arquivos Excel ou JSON.

O módulo de ação retorna um erro quando ocorre o seguinte:

- O arquivo não existe no runtime.
- O módulo de ação encontra um erro ao executar a operação.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
`path`	O caminho do arquivo.	String	Sim	N/D	N/D	Sim
`encoding`	O padrão de codificação.	String	Não	`utf8`	`utf8`, `utf-8`, `utf16`,`utf-16`, `utf16-LE`, `utf-16-LE` `utf16-BE`, `utf-16-BE` , `utf32`, `utf-32`, `utf32-LE`,`utf-32-LE` , `utf32-BE` e `utf-32-BE` . O valor da opção de codificação não diferencia maiúsculas de	Sim

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
					minúsculas.	
`printFileContent`	Imprime o conteúdo do arquivo no arquivo `console.log` .	Booleano	Não	false	N/D	Sim.

Exemplo de entrada: ler um arquivo (Linux)

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

Exemplo de entrada: ler um arquivo (Windows)

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

Exemplo de entrada: ler um arquivo e especificar o padrão de codificação

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

Exemplo de entrada: ler um arquivo e imprimir no **`console.log`** arquivo do

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

Saída

Campo	Descrição	Tipo
`content`	O conteúdo do arquivo.	string

Exemplo de saída

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

SetFileEncoding (Linux, Windows, macOS)

O módulo de SetFileEncodingação modifica a propriedade de codificação de um arquivo existente. Este módulo pode converter a codificação do arquivo do `utf-8` em um padrão de codificação especificado. Por padrão, presume-se que `utf-16` e `utf-32` usam a codificação little-endian.

O módulo de ação retorna um erro quando ocorre o seguinte:

- Você não tem a permissão para realizar a modificação especificada.
- O arquivo não existe no runtime.
- O módulo de ação encontra um erro ao executar a operação.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
`path`	O caminho do arquivo.	String	Sim	N/D	N/D	Sim
`encoding`	O padrão de codificação.	String	Não	`utf8`	`utf8`, `utf-8`, `utf16`,`utf-16`, `utf16-LE`, `utf-16-LE` `utf16-BE`, `utf-16-BE` , `utf32`, `utf-32`, `utf32-LE`,`utf-32-LE` , `utf32-BE` e `utf-32-BE` . O valor da opção de codificação não diferencia maiúsculas de	Sim

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
					minúsculas.	

Exemplo de entrada: definir propriedade de codificação do arquivo

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

Saída

Nenhum.

SetFileOwner (Linux, Windows, macOS)

O módulo de SetFileOwneração modifica as propriedades `owner` e do `group` proprietário de um arquivo existente. Se o arquivo especificado for um link simbólico, o módulo modifica a propriedade `owner` do arquivo de origem. Este módulo não é compatível com plataformas Windows.

Este módulo aceita nomes de usuários e grupos como entradas. Se o nome do grupo não for fornecido, o módulo atribuirá o proprietário do grupo do arquivo ao grupo ao qual o usuário pertence.

O módulo de ação retorna um erro quando ocorre o seguinte:

- Você não tem a permissão para realizar a modificação especificada.
- O nome do usuário ou grupo especificado não existe no runtime.
- O arquivo não existe no runtime.
- O módulo de ação encontra um erro ao executar a operação.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
`path`	O caminho do arquivo.	String	Sim	N/D	N/D	Não compatível com Windows.
`owner`	O nome do usuário.	string	Sim	N/D	N/D	Não compatível com Windows.
`group`	O nome do grupo de usuários.	String	Não	O nome do grupo ao qual o usuário pertence.	N/D	Não compatível com Windows.

Exemplo de entrada: definir a propriedade do proprietário do arquivo sem especificar o nome do grupo de usuários

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

Exemplo de entrada: definir a propriedade do proprietário do arquivo especificando o nome do grupo de usuários

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

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

Saída

Nenhum.

SetFolderOwner (Linux, Windows, macOS)

O módulo de SetFolderOwneração modifica recursivamente as propriedades `owner` e o `group` proprietário de uma pasta existente. Por padrão, o módulo pode modificar a propriedade de todo o conteúdo de uma pasta. Você pode configurar a opção `recursive` para `false` para substituir esse comportamento. Este módulo não é compatível com plataformas Windows.

Este módulo aceita nomes de usuários e grupos como entradas. Se o nome do grupo não for fornecido, o módulo atribuirá o proprietário do grupo do arquivo ao grupo ao qual o usuário pertence.

O módulo de ação retorna um erro quando ocorre o seguinte:

- Você não tem a permissão para realizar a modificação especificada.
- O nome do usuário ou grupo especificado não existe no runtime.
- A pasta não existe no runtime.
- O módulo de ação encontra um erro ao executar a operação.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
`path`	O caminho da pasta.	String	Sim	N/D	N/D	Não compatível com Windows.

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
`owner`	O nome do usuário.	string	Sim	N/D	N/D	Não compatível com Windows.
`group`	O nome do grupo de usuários.	String	Não	O nome do grupo ao qual o usuário pertence.	N/D	Não compatível com Windows.
`recursive`	Substitui o comportamento padrão de modificar a propriedade de todo o conteúdo de uma pasta quando definido como `false`.	Booleano	Não	`true`	N/D	Não compatível com Windows.

Exemplo de entrada: definir a propriedade do proprietário da pasta sem especificar o nome do grupo de usuários

```
- name: SettingFolderPropertyWithOutGroup
  action: SetFolderOwner
```

```
    inputs:
      - path: /SampleFolder/
        owner: LinuxUser
```

Exemplo de entrada: definir a propriedade do proprietário da pasta sem substituir a propriedade de todo o conteúdo em uma pasta

```
- name: SettingFolderPropertyWithOutRecursively
  action: SetFolderOwner
  inputs:
    - path: /SampleFolder/
      owner: LinuxUser
      recursive: false
```

Exemplo de entrada: definir a propriedade do arquivo especificando o nome do grupo de usuários

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

Saída

Nenhum.

SetFilePermissions (Linux, Windows, macOS)

O módulo de SetFilePermissionsação modifica o `permissions` de um arquivo existente. Este módulo não é compatível com plataformas Windows.

A entrada para `permissions` deve ser um valor de string.

Esse módulo de ação pode criar um arquivo com permissões definidas pelo valor unmask padrão do sistema operacional. Você deve definir o valor `umask` se quiser substituir o valor padrão.

O módulo de ação retorna um erro quando ocorre o seguinte:

- Você não tem a permissão para realizar a modificação especificada.
- O arquivo não existe no runtime.
- O módulo de ação encontra um erro ao executar a operação.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
`path`	O caminho do arquivo.	String	Sim	N/D	N/D	Não compatível com Windows.
`permissions`	As permissões de arquivo.	String	Sim	N/D	N/D	Não compatível com Windows.

Exemplo de entrada: modificar permissões de arquivo

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

Saída

Nenhum.

SetFolderPermissions (Linux, Windows, macOS)

O módulo de SetFolderPermissionsação modifica recursivamente a `permissions` de uma pasta existente e de todos os seus subarquivos e subpastas. Por padrão, esse módulo pode modificar as permissões para todo o conteúdo da pasta especificada. Você pode configurar a `recursive` opção para `false` para substituir esse comportamento. Este módulo não é compatível com plataformas Windows.

A entrada para `permissions` deve ser um valor de string.

Esse módulo de ação pode modificar as permissões de acordo com o valor umask padrão do sistema operacional. Você deve definir o valor `umask` se quiser substituir o valor padrão.

O módulo de ação retorna um erro quando ocorre o seguinte:

- Você não tem a permissão para realizar a modificação especificada.
- A pasta não existe no runtime.
- O módulo de ação encontra um erro ao executar a operação.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
`path`	O caminho da pasta.	String	Sim	N/D	N/D	Não compatível com Windows.
`permissions`	As permissões da pasta.	String	Sim	N/D	N/D	Não compatível com Windows.
`recursive`	Substitui o comportamento padrão de modificar as permissões de todo o conteúdo de uma pasta	Booleano	Não	`true`	N/D	Não compatível com Windows.

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
	quando definido como `false`.					

Exemplo de entrada: definir permissões de pasta

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

Exemplo de entrada: definir permissões de pasta sem modificar as permissões para todo o conteúdo de uma pasta

```
- name: SettingFolderPermissionsNoRecursive
  action: SetFolderPermissions
  inputs:
    - path: /home/UserName/SampleFolder/
      permissions: 777
      recursive: false
```

Saída

Nenhum.

Ações de instalação de software

A seção a seguir descreve os módulos de ação que instalam ou desinstalam software.

Requisitos do IAM

Se o caminho de download da instalação for um URI do S3, o perfil do IAM associado ao perfil de instância deverá ter permissão para executar o módulo de ação `S3Download`. Para conceder a permissão necessária, anexe a `S3:GetObject` política do IAM ao perfil do IAM associada ao seu perfil de instância e especifique o caminho para seu bucket. Por exemplo, *`arn:aws:s3:::BucketName/*`*.

Entradas MSI complexas

Se suas cadeias de caracteres de entrada contiverem aspas duplas (''), você deverá usar um dos seguintes métodos para garantir que elas sejam interpretadas corretamente:

- Você pode usar aspas simples (') na parte externa da string, para contê-la, e aspas duplas (“) dentro da string, conforme mostrado no exemplo a seguir.

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

  Nesse caso, se você precisar usar um apóstrofo dentro de sua string, deverá contorná-la. Isso significa usar outra aspa simples (') antes do apóstrofo.
- Você pode usar aspas duplas (“) na parte externa da string para contê-la. E você pode contornar qualquer aspa dupla dentro de sua string, usando o caractere de barra invertida (\), conforme mostrado no exemplo a seguir.

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

Ambos os métodos passam o valor `COMPANYNAME="Acme ""Widgets"" and ""Gizmos."""` para o comando msiexec.

Módulos de ações de instalação de software

- InstallMSI (Windows)
- UninstallMSI (Windows)

InstallMSI (Windows)

O módulo de ação `InstallMSI` instala um aplicativo do Windows usando um arquivo MSI. Você pode especificar o arquivo MSI usando um caminho local, um URI de objeto do S3 ou um URL da web. A opção de reinicialização configura o comportamento de reinicialização do sistema.

AWSTOE gera o msiexec comando com base nos parâmetros de entrada para o módulo de ação. Os valores dos parâmetros de entrada `path` (localização do arquivo MSI) e `logFile` (localização do arquivo de log) devem estar entre aspas (“).

Os seguintes códigos de saída MSI são considerados bem-sucedidos:

- 0 (Sucesso)
- 1614 (ERROR_PRODUCT_UNINSTALLED)
- 1641 (reinicialização iniciada)
- 3010 (reinicialização necessária)

Entrada

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
`path`	Especifique o local do arquivo MSI usando uma das opções a seguir: • O caminho do arquivo local. O caminho pode ser absoluto ou relativo •	String	Sim	N/D	N/D

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
	Um URI de objeto S3 válido. • Um HTTP/HTTPS URL da web válido (HTTPS é recomendado) que segue o padrão RFC 3986. Expressões de encadeamento são permitidas.				

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
`reboot`	Configure o comportamento de reinicialização do sistema após a execução bem-sucedida do módulo de ação. Configurações: • `Force`: inicia a reinicialização do sistema após a execução bem-sucedida do comando msiexec. • `Allow`: inicia uma reinicialização do sistema se o comando	String	Não	`Allow`	`Allow`, `Force`, `Skip`

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
	msiexec retornar um código de saída que indique que uma reinicialização é necessária. • `Skip`: registra o log de uma mensagem informativa no arquivo `console.log` indicando que a reinicialização foi ignorada. Essa opção impede a reinicialização, mesmo que o comando msiexec retorne um código				

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
	de saída indicando que uma reinicial ização é necessária.				

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
`logOptions`	Especifique as opções a serem usadas para o registro de instalação do MSI. Os sinalizadores especificados são passados para o instalador do MSI, junto com o parâmetro da linha do comando `/L` para ativar o registro. Se nenhum sinalizador for especificado, AWSTOE usa o valor padrão. Para obter mais informações sobre opções de log do MSI, consulte	String	Não	`*VX`	`i,w,e,a,r,u,c,m,o,p,v,x,+,!,*`

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
	Opções de linha de comando na documentação de produto do Microsoft Windows Installer.				
`logFile`	Um caminho absoluto ou relativo para a localização do arquivo de log. Se o caminho de log do arquivo não existir, será criado. Se o caminho do arquivo de log não for fornecido, AWSTOE não armazenará o log de instalação do MSI.	String	Não	N/D	N/D

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
`properties`	Pares valores-chave de propriedades de registro MSI, por exemplo: `TARGETDIR: "C:\target\location"` Observação: a modificação das seguintes propriedades não é permitida: • `REBOOT="ReallySupress"` • `REINSTALLMODE="ecmus"` • `REINSTALL="ALL"`	Map[String]String	Não	N/D	N/D

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
`ignoreAuthenticodeSignatureErrors`	Sinalize para ignorar erros de validação de assinatura authenticode para o instalador especificado no caminho. O comando Get-AuthenticodeSignature é usado para validar os instaladores. Configurações: • `true`: os erros de validação são ignorados e o instalador é executado. • `false`: os erros de validação não são	Booleano	Não	`false`	`true`, `false`

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
	ignorados. O instalador é executado somente quando a validação é bem-sucedida. Esse é o comportamento padrão.				

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
`allowUnsignedInstaller`	Sinalize para permitir a execução do instalador não assinado especificado no caminho. O comando Get-AuthenticodeSignature é usado para validar os instaladores. Configurações: • `true`: ignora o status de `NotSigned` retornado pelo comando Get-AuthenticodeSignature e executa o instalador. • `false`: requer que	Booleano	Não	`false`	`true`, `false`

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
	o instalador seja assinado. Instaladores não assinados não serão executados. Esse é o comportamento padrão.				

Exemplos

Os exemplos a seguir mostram variações da seção de entrada do documento do componente, dependendo do caminho de instalação.

Exemplo de entrada: caminho de instalação do documento local

```
- 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
```

Exemplo de entrada: instalação do caminho do 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
```

Exemplo de entrada: instalação do caminho da 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
```

Saída

Este é um exemplo da saída do módulo de ação `InstallMSI`.

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

UninstallMSI (Windows)

O módulo de ação `UninstallMSI` permite que você remova um aplicativo do Windows usando um arquivo MSI. Você pode especificar o local do arquivo MSI usando um caminho local, um URI de objeto do S3 ou um URL da web. A opção de reinicialização configura o comportamento de reinicialização do sistema.

AWSTOE gera o msiexec comando com base nos parâmetros de entrada para o módulo de ação. A localização do arquivo MSI (`path`) e a localização do arquivo de log (`logFile`) são explicitamente colocadas entre aspas duplas (“) ao gerar o comando. msiexec

Os seguintes códigos de saída MSI são considerados bem-sucedidos:

- 0 (Sucesso)
- 1605 (ERROR_UNKNOWN_PRODUCT)
- 1614 (ERROR_PRODUCT_UNINSTALLED)
- 1641 (reinicialização iniciada)
- 3010 (reinicialização necessária)

Entrada

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
`path`	Especifique o local do arquivo MSI usando uma das opções a seguir: • O caminho do arquivo local. O caminho pode ser absoluto ou relativo. • Um URI de objeto S3 válido. •	String	Sim	N/D	N/D

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
	Um HTTP/HTTPS URL da web válido (HTTPS é recomendado) que segue o padrão RFC 3986. Expressões de encadeamento são permitidas.				

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
`reboot`	Configure o comportamento de reinicialização do sistema após a execução bem-sucedida do módulo de ação. Configurações: • `Force`: inicia a reinicialização do sistema após a execução bem-sucedida do comando msiexec. • `Allow`: inicia uma reinicialização do sistema se o comando	String	Não	`Allow`	`Allow`, `Force`, `Skip`

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
	msiexec retornar um código de saída que indique que uma reinicialização é necessária. • `Skip`: registra o log de uma mensagem informativa no arquivo `console.log` indicando que a reinicialização foi ignorada. Essa opção impede a reinicialização, mesmo que o comando msiexec retorne um código				

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
	de saída indicando que uma reinicial ização é necessária.				

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
`logOptions`	Especifique as opções a serem usadas para o registro de instalação do MSI. Os sinalizadores especificados são passados para o instalador do MSI, junto com o parâmetro da linha do comando `/L` para ativar o registro. Se nenhum sinalizador for especificado, AWSTOE usa o valor padrão. Para obter mais informações sobre opções de log do MSI, consulte	String	Não	`*VX`	`i,w,e,a,r,u,c,m,o,p,v,x,+,!,*`

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
	Opções de linha de comando na documentação de produto do Microsoft Windows Installer.				
`logFile`	Um caminho absoluto ou relativo para a localização do arquivo de log. Se o caminho de log do arquivo não existir, será criado. Se o caminho do arquivo de log não for fornecido, AWSTOE não armazenará o log de instalação do MSI.	String	Não	N/D	N/D

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
`properties`	Pares valores-chave de propriedades de registro MSI, por exemplo: `TARGETDIR: "C:\target\location"` Observação: a modificação das seguintes propriedades não é permitida: • `REBOOT="ReallySupress"` • `REINSTALLMODE="ecmus"` • `REINSTALL="ALL"`	Map[String]String	Não	N/D	N/D

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
`ignoreAuthenticodeSignatureErrors`	Sinalize para ignorar erros de validação de assinatura authenticode para o instalador especificado no caminho. O comando Get-AuthenticodeSignature é usado para validar os instaladores. Configurações: • `true`: os erros de validação são ignorados e o instalador é executado. • `false`: os erros de validação não são	Booleano	Não	`false`	`true`, `false`

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
	ignorados. O instalador é executado somente quando a validação é bem-sucedida. Esse é o comportamento padrão.				

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
`allowUnsignedInstaller`	Sinalize para permitir a execução do instalador não assinado especificado no caminho. O comando Get-AuthenticodeSignature é usado para validar os instaladores. Configurações: • `true`: ignora o status de `NotSigned` retornado pelo comando Get-AuthenticodeSignature e executa o instalador. • `false`: requer que	Booleano	Não	`false`	`true`, `false`

Nome da chave	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
	o instalador seja assinado. Instaladores não assinados não serão executados. Esse é o comportamento padrão.				

Exemplos

Os exemplos a seguir mostram variações da seção de entrada do documento do componente, dependendo do caminho de instalação.

Exemplo de entrada: remove a insta;ação do caminho do documento local

```
- 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
```

Exemplo de entrada: remove a instalação do caminho do 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
```

Exemplo de entrada: remove a instalação do caminho da 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
```

Saída

Este é um exemplo da saída do módulo de ação `UninstallMSI`.

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

Módulos de ação do sistema

A seção a seguir descreve os módulos de ação que executam ações do sistema ou atualizam as configurações do sistema.

Módulos de ação do sistema

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

- UpdateOS (Linux, Windows)

Reboot (Linux, Windows)

O módulo de ação de Reboot reinicia a instância. Ele tem uma opção configurável para atrasar o início da reinicialização. Por padrão, `delaySeconds` está definido como `0`, o que significa que não há atraso. O tempo limite da etapa não é compatível com o módulo de ação de reinicialização, pois não se aplica quando a instância é reinicializada.

Se o aplicativo for chamado pelo Systems Manager Agent, ele entregará o código de saída (`3010` para Windows, `194` para Linux) para o Systems Manager Agent. O Systems Manager Agent lida com a reinicialização do sistema conforme descrito em Como reinicializar a instância gerenciada a partir de scripts.

Se o aplicativo é invocado no host como um processo independente, ele salva o estado de execução atual, configura um gatilho de execução automática pós-reinicialização para executar novamente o aplicativo após a reinicialização e, em seguida, reinicializa o sistema.

Acionador de execução automática após a reinicialização:

- Windows. AWSTOE cria uma entrada do Windows Task Scheduler com um gatilho que é executado automaticamente em `SystemStartup`
- Linux. AWSTOE adiciona um trabalho no crontab que é executado automaticamente após a reinicialização do sistema.

```
@reboot /download/path/awstoe run --document s3://bucket/key/doc.yaml
```

Esse gatilho é limpo quando o aplicativo é iniciado.

> **Novas tentativas**
>
> Por padrão, o número máximo de novas tentativas é definido para o `CommandRetryLimit` do Systems Manager. Se o número de reinicializações exceder o limite de novas tentativas, a automação falhará. É possível alterar o limite editando o arquivo de configuração do agente do Systems Manager (`Mds.CommandRetryLimit`). Consulte Configuração do runtime no código aberto do agente do Systems Manager.

Para usar o módulo de ação de Reboot, para etapas que contêm reinicialização `exitcode` (por exemplo, `3010`), você deve executar o binário do aplicativo como `sudo user`.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório	Padrão
`delaySeconds`	Atrasa um determinado período de tempo antes de iniciar uma reinicialização.	Inteiro	Não	`0`

Exemplo de entrada: etapa de reinicialização

```
- name: RebootStep
  action: Reboot
  onFailure: Abort
  maxAttempts: 2
  inputs:
    delaySeconds: 60
```

Saída

Nenhum.

Quando o módulo Reboot é concluído, o Image Builder continua na próxima etapa da compilação.

SetRegistry (Windows)

O módulo de SetRegistryação aceita uma lista de entradas e permite que você defina o valor da chave de registro especificada. Se não existir uma chave de registro, ela será criada no caminho definido. Esse atributo se aplica somente ao Windows.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório
`path`	Caminho da chave de registro.	String	Sim

Nome da chave	Descrição	Tipo	Obrigatório
`name`	Nome da chave de registro.	String	Sim
`value`	Valor da chave de registro.	String/Number/Array	Sim
`type`	Tipo de valor da chave de registro.	String	Sim

Prefixos de caminho compatíveis

- `HKEY_CLASSES_ROOT / HKCR:`
- `HKEY_USERS / HKU:`
- `HKEY_LOCAL_MACHINE / HKLM:`
- `HKEY_CURRENT_CONFIG / HKCC:`
- `HKEY_CURRENT_USER / HKCU:`

Tipos compatíveis

- `BINARY`
- `DWORD`
- `QWORD`
- `SZ`
- `EXPAND_SZ`
- `MULTI_SZ`

Exemplo de entrada: definir valores de chave de registro

```
- 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
```

Saída

Nenhum.

UpdateOS (Linux, Windows)

O módulo de ação UpdateOS adiciona suporte para a instalação de atualizações do Windows e do Linux. Ele instala todas as atualizações disponíveis por padrão. Como alternativa, você pode configurar uma lista de uma ou mais atualizações específicas para instalar o módulo de ação. Você também pode especificar atualizações a serem excluídas da instalação.

Se as listas “incluir” e “excluir” forem fornecidas, a lista de atualizações resultante poderá incluir somente aquelas listadas na lista “incluir” que não estejam listadas na lista “excluir”.

> **Note**
>
> O UpdateOS não é compatível com o Amazon Linux 2023 (AL2023). Recomendamos que você atualize sua AMI básica para a nova versão que vem com cada lançamento. Para outras alternativas, consulte Controlar as atualizações recebidas de versões principais e secundárias no Guia do usuário do Amazon Linux 2023.

- Windows. As atualizações são instaladas a partir da fonte de atualização configurada na máquina de destino.
- Linux. O aplicativo verifica o gerenciador de pacotes compatível na plataforma Linux e usa o gerenciador de pacotes `yum` ou `apt-get`. Se nenhum dos dois for compatível, será retornado um erro. Você deve ter permissões de `sudo` para executar o módulo de ação UpdateOS. Se você não tiver permissões de `sudo`, um `error.Input` será retornado.

Entrada

Nome da chave	Descrição	Tipo	Obrigatório
`include`	Para Windows, você pode especificar o seguinte: • Um ou mais artigos da Base de Conhecimento da Microsoft (KB) IDs para incluir na lista de atualizaç ões que podem ser instaladas. Os formatos válidos são `KB1234567` ou `1234567`. • Um nome de atualização usando um valor curinga (`*`). Os formatos válidos são `Security*` ou `*Security*` . Para Linux, você pode especificar um ou mais pacotes a serem incluídos na lista de atualizações para instalação.	Lista de strings	Não
`exclude`		Lista de strings	Não

Nome da chave	Descrição	Tipo	Obrigatório
	Para Windows, você pode especificar o seguinte: • Um ou mais artigos IDs da Base de Conhecimento da Microsoft (KB) a serem incluídos na lista de atualizações a serem excluídas da instalação. Os formatos válidos são `KB1234567` ou `1234567`. • Um nome de atualização usando um valor curinga (`*`). Os formatos válidos são `Security*` ou `*Security*` . Para Linux, você pode especificar um ou mais pacotes a serem excluídos na lista de atualizações para instalação.		

Exemplo de entrada: adicionar suporte para instalação de atualizações do Linux

```
- name: UpdateMyLinux
  action: UpdateOS
  onFailure: Abort
  maxAttempts: 3
  inputs:
    exclude:
      - ec2-hibinit-agent
```

Exemplo de entrada: adicionar suporte para instalação de atualizações do Windows

```
- name: UpdateWindowsOperatingSystem
  action: UpdateOS
  onFailure: Abort
  maxAttempts: 3
  inputs:
    include:
      - KB1234567
      - '*Security*'
```

Saída

Nenhum.

Configurar a entrada para o comando de AWSTOE execução

Para simplificar a entrada da linha de comando para seu AWSTOE run comando, você pode incluir configurações para parâmetros e opções de comando em um arquivo de configuração de entrada no formato JSON com uma extensão de `.json` arquivo. AWSTOE pode ler seu arquivo em um dos seguintes locais:

- Um caminho de arquivo local (*`./config.json`*).
- Um bucket S3 (*`s3://<bucket-path>/<bucket-name>/config.json`*).

Ao inserir o comando run, você pode especificar o arquivo de configuração de entrada usando o parâmetro --config. Por exemplo:

```
awstoe run --config <file-path>/config.json
```

Um arquivo de configuração de entrada

O arquivo JSON de configuração de entrada inclui pares de valores-chave para todas as configurações que você pode fornecer diretamente por meio de parâmetros e opções de comando run. Se você especificar uma configuração no arquivo de configuração de entrada e no comando run, como parâmetro ou opção, as seguintes regras de precedência se aplicam:

Regras de precedência

1. Uma configuração fornecida diretamente ao run comando no AWS CLI, por meio de um parâmetro ou opção, substitui qualquer valor definido no arquivo de configuração de entrada para a mesma configuração.
2. Uma configuração no arquivo de configuração de entrada substitui o valor padrão de um componente.
3. Se nenhuma outra configuração for passada para o documento do componente, ele poderá aplicar um valor padrão, se houver.

Há duas exceções a essa regra: documentos e parâmetros. Estas configurações funcionam de forma diferente na configuração de entrada e como parâmetros de comando. Se você usar o arquivo de configuração de entrada, não deverá especificar esses parâmetros diretamente no comando run. Fazer isso gerará um erro.

Configurações do componente

O arquivo de configuração de entrada contém as seguintes configurações. Para simplificar seu arquivo, você pode omitir as configurações opcionais que não são necessárias. Todas as configurações são opcionais, salvo indicação em contrário.

- cwIgnoreFailures(Boolean) — Ignore as falhas de registro dos CloudWatch registros.
- cwLogGroup(String) — O `LogGroup` nome dos CloudWatch registros.
- cwLogRegion(String) — A AWS região que se aplica aos CloudWatch registros.
- cwLogStream(String) — O `LogStream` nome dos CloudWatch registros, que direciona para AWSTOE onde transmitir o `console.log` arquivo.
- DocumentS3 BucketOwner (String) — O ID da conta do proprietário do bucket para documentos baseados em URI do S3.
- documents (matriz de objetos, obrigatório) — Uma matriz de objetos JSON representando os documentos do componente YAML que o AWSTOE run comando está executando. Pelo menos um documento de componente deve ser especificado.

Cada objeto consiste dos seguinte campos:

- path (String, obrigatório): o local do arquivo do documento do componente YAML. Este valor deve ser um dos seguintes:
  - Um caminho de arquivo local (`./component-doc-example.yaml`).
  - Um URI do S3 (`s3://bucket/key`).
  - Uma versão ARN de compilação do componente Image Builder (arn:aws:imagebuilder:us-west-/ /2021.12.02/1). `2:123456789012 my-example-component`
- parâmetros (matriz de objetos): uma matriz de objetos de pares de valores-chave, cada um representando um parâmetro específico do componente que o comando runtransmite ao executar o documento do componente. Os parâmetros são opcionais para componentes. O documento do componente pode ou não ter parâmetros definidos.

  Cada objeto consiste dos seguinte campos:

  - nome (string, obrigatório): o nome do parâmetro de componente.
  - valor (String, obrigatório): o valor a ser passado para o documento do componente para o parâmetro nomeado.

  Para saber mais sobre os parâmetros do componente, consulte a seção Parâmetros na página Usar variáveis em seu documento de componente personalizado.

- ExecutonId (String): esse é o ID exclusivo que se aplica à execução do comando atual run. Este ID é incluído nos nomes dos arquivos de saída e log, para identificar esses arquivos de forma exclusiva e vinculá-los à execução atual do comando. Se essa configuração for omitida, AWSTOE gera um GUID.
- LogDirectory (String) — O diretório de destino em que AWSTOE armazena todos os arquivos de log da execução desse comando. Este arquivo está localizado no diretório a seguir, por padrão: `TOE_<DATETIME>_<EXECUTIONID>`. Se você não especificar o diretório de log, AWSTOE usa o diretório de trabalho atual (`.`).
- LogS3 BucketName (String) — Se os registros do componente estiverem armazenados no Amazon S3 (recomendado) AWSTOE , carrega os registros do aplicativo do componente no bucket do S3 nomeado nesse parâmetro.
- LogS3 BucketOwner (String) — Se os logs do componente estiverem armazenados no Amazon S3 (recomendado), esse é o ID da conta do proprietário do bucket em AWSTOE que grava os arquivos de log.

- LogS3 KeyPrefix (String) — Se os logs do componente estiverem armazenados no Amazon S3 (recomendado), esse é o prefixo da chave de objeto do S3 para a localização do log no bucket.
- parameters (matriz de objetos): uma matriz de objetos de pares de valores-chave que representam parâmetros que se aplicam globalmente a todos os componentes incluídos na execução atual do comando run.
  - nome (string, obrigatório): o nome do parâmetro global.
  - valor (String, obrigatório): o valor a ser passado para todos os documentos do componente do parâmetro nomeado.
- fases (String): uma lista separada por vírgulas que especifica quais fases devem ser executadas a partir dos documentos do componente YAML. Se um documento do componente incluir fases adicionais, elas não serão executadas.
- StateDirectory (String): o caminho do arquivo em que os arquivos de rastreamento de estado são armazenados.
- traço (Boolean): habilita o log detalhado no console.

Exemplos

O exemplo a seguir mostra um arquivo de configuração de entrada que executa as fases `build` e `test` para dois documentos de componentes: `sampledoc.yaml` e `conversation-intro.yaml`. Cada documento de componente tem um parâmetro que se aplica somente a si mesmo, e ambos usam um parâmetro compartilhado. O parâmetro `project` se aplica aos dois documentos de componentes.

```
{
   "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>"
}
```

Recursos de imagens de saída do Image Builder

Depois de criar recursos de imagem para AMI ou imagens de contêiner com o Image Builder, você pode gerenciá-los usando o console do Image Builder, por meio da API do Image Builder ou com comandos imagebuilder em AWS CLI.

> **Tip**
>
> Quando você tem vários recursos do mesmo tipo, a marcação com tags ajuda a identificar um recurso específico com base nas tags que você atribuiu a ele. Para obter mais informações sobre como marcar seus recursos usando os comandos do Image Builder no AWS CLI, consulte a Marcar recursos seção deste guia.

Esta seção aborda como listar, visualizar e criar imagens. Para obter informações sobre fluxos de trabalho de imagem e como gerenciá-los, consulte Gerenciar fluxos de trabalho de compilação e teste para imagens do Image Builder.

Conteúdo

- Listar versões de compilação e imagens
- Visualizar os detalhes do recurso de imagem
- Criar imagens personalizadas com o Image Builder
- Importar e exportar imagens de máquina virtual com o Image Builder
- Importe imagens de disco ISO verificadas do Windows com o Image Builder
- Gerenciar descobertas de segurança para imagens do Image Builder
- Limpar recursos do Image Builder

Listar versões de compilação e imagens

Na página Imagens no console do Image Builder, você pode ver listas de todos os recursos de imagem do Image Builder que você possui, que estão compartilhados com você e aos quais você tem acesso. Os resultados da lista incluem alguns detalhes importantes sobre esses recursos.

Você também pode ver todas as imagens em sua conta que têm ações pendentes de fluxo de trabalho.

Conteúdo

- Listar imagens
- Listar imagens que estão aguardando ação
- Listar versões da compilação de imagem

Listar imagens

Esta seção descreve as diferentes maneiras pelas quais você pode listar informações sobre suas imagens.

Você pode usar um dos métodos a seguir para listar os recursos de imagem do Image Builder aos quais você tem acesso. Para a ação da API, consulte ListImagesna Referência da API do EC2 Image Builder. Para ver a solicitação de SDK associada, consulte o link Consulte também na mesma página.

Conteúdo

- Listar imagens no console
- Listar imagens com AWS CLI comandos

Listar imagens no console

Siga estas etapas para abrir a página da lista de imagens no console:

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Escolha Imagens no painel de navegação.

A página Imagens no console é dividida em guias, com base no proprietário da imagem ou nas ações pendentes de fluxo de trabalho. Esta seção aborda as três primeiras guias que mostram as imagens de sua propriedade ou às quais você tem acesso.

Guia do console: De minha propriedade

Na guia De minha propriedade, você pode usar os filtros a seguir para otimizar os resultados da lista de imagens.

- É possível pesquisar pelo nome inteiro ou por parte do nome na barra de pesquisas.

- É possível filtrar imagens com base na plataforma do sistema operacional (Windows, Linux ou macOS).
- É possível filtrar imagens com base no tipo de saída que elas produzem (AMI ou imagem de contêiner).
- Você pode usar a fonte de filtro para encontrar imagens que foram importadas de uma máquina virtual (VMIE) ou de uma imagem de disco ISO.

Além dos controles de filtro, a guia De minha propriedade mostra uma lista das imagens do Image Builder que você criou, com os seguintes detalhes dos recursos listados:

Nome/versão

Os nomes dos recursos de imagem do Image Builder começam com o nome e a versão da fórmula que serviu de base para a compilação deles. Selecione o link para ver todas as versões relacionadas de compilação de imagens.

Tipo

O tipo de imagem de saída que o Image Builder cria para esse recurso de imagem (uma AMI ou uma imagem de contêiner).

Plataforma

A plataforma do sistema operacional do recurso de imagem; por exemplo, “Linux”, “Windows” ou “macOS”.

Fonte da imagem

A origem da imagem de base que o Image Builder usou para compilar esse recurso de imagem. Isso é usado principalmente para filtrar resultados de imagens que foram importadas de uma máquina virtual (VMIE).

Hora de criação

A data e a hora em que o Image Builder criou a versão atual do recurso de imagem.

ARN

O nome do recurso da Amazon (ARN) da versão atual do recurso de imagem.

Guia do console: Compartilhado comigo

Na guia Compartilhado comigo, você pode usar os filtros a seguir para otimizar os resultados da lista de imagens.

- É possível pesquisar pelo nome inteiro ou por parte do nome na barra de pesquisas.
- É possível filtrar imagens com base na plataforma do sistema operacional (Windows, Linux ou macOS).
- É possível filtrar imagens com base no tipo de saída que elas produzem (AMI ou imagem de contêiner).
- Você pode usar a fonte de filtro para encontrar imagens que foram importadas de uma máquina virtual (VMIE) ou de uma imagem de disco ISO.

Além dos controles de filtro, a guia Compartilhado comigo mostra uma lista das imagens do Image Builder que foram compartilhadas com você, com os seguintes detalhes dos recursos listados:

Nome da imagem

O nome do recurso de imagem que foi compartilhado com você. Para usar uma imagem compartilhada em uma fórmula, selecione a opção Selecionar imagens gerenciadas e altere a Origem da imagem para Imagens compartilhadas comigo.

Tipo

O tipo de imagem de saída que o Image Builder cria para esse recurso de imagem (uma AMI ou uma imagem de contêiner).

Versão

A versão da plataforma do sistema operacional para o recurso de imagem, geralmente um campo numérico no seguinte formato: <major>.<minor>.<patch>.

Fonte da imagem

A origem da imagem de base que o Image Builder usou para compilar esse recurso de imagem, se for o caso. Isso é usado principalmente para filtrar resultados de imagens que foram importadas de uma máquina virtual (VMIE).

Plataforma

A plataforma do sistema operacional do recurso de imagem; por exemplo, “Linux”, “Windows” ou “macOS”.

Hora de criação

A data e a hora em que o Image Builder criou a versão do recurso de imagem que foi compartilhada com você.

Proprietário

O proprietário do recurso de imagem compartilhada.

ARN

O nome do recurso da Amazon (ARN) da versão do recurso de imagem que foi compartilhada com você.

Guia do console: Gerenciado pela Amazon

Na guia Gerenciado pela Amazon, você pode usar os filtros a seguir para otimizar os resultados da lista de imagens.

- É possível pesquisar pelo nome inteiro ou por parte do nome na barra de pesquisas.
- É possível filtrar imagens com base na plataforma do sistema operacional (Windows, Linux ou macOS).
- É possível filtrar imagens com base no tipo de saída que elas produzem (AMI ou imagem de contêiner).
- Você pode usar a fonte de filtro para encontrar imagens que foram importadas de uma máquina virtual (VMIE) ou de uma imagem de disco ISO.

Seguindo os controles de filtro, a guia Gerenciado pela Amazon mostra uma lista de imagens do Image Builder gerenciadas pela Amazon que você pode usar como imagens de base para suas fórmulas. O Image Builder exibe os seguintes detalhes dos recursos listados:

Nome da imagem

O nome da imagem gerenciada. Quando você cria uma fórmula, o padrão para sua imagem de base é Início rápido (gerenciado pela Amazon). As imagens listadas nessa guia preenchem a lista Nomes de imagem associada à plataforma do sistema operacional que você escolhe para sua imagem de base ao criar uma fórmula.

Tipo

O tipo de imagem de saída que o Image Builder cria para esse recurso de imagem (uma AMI ou uma imagem de contêiner).

Versão

A versão da plataforma do sistema operacional para o recurso de imagem, geralmente um campo numérico no seguinte formato: <major>.<minor>.<patch>.

Plataforma

A plataforma do sistema operacional do recurso de imagem; por exemplo, “Linux”, “Windows” ou “macOS”.

Hora de criação

A data e a hora em que o Image Builder criou a versão do recurso de imagem que foi compartilhada com você.

Proprietário

A Amazon é proprietária das imagens gerenciadas.

ARN

O nome do recurso da Amazon (ARN) da versão do recurso de imagem que foi compartilhada com você.

Listar imagens com AWS CLI comandos

Ao executar o list-images comando no AWS CLI, você pode obter uma lista de imagens que você possui ou às quais tem acesso.

O exemplo de comando a seguir mostra como usar o comando list-images sem filtros para listar todos os recursos de imagem do Image Builder de sua propriedade.

Exemplo: listar todas as imagens

```
aws imagebuilder list-images
```

Saída:

```
{
 "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"
  }
 ]
}
```

Ao executar o comando list-images, você pode aplicar filtros para otimizar os resultados, como mostra o exemplo a seguir. Para obter mais informações sobre como filtrar seus resultados, consulte o comando list-images na AWS CLI Referência de comandos.

Exemplo: filtrar para imagens do Linux

```
aws imagebuilder list-images --filters name="platform",values="Linux"
```

Saída:

```
{
```

```
 "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"
  }
 ]
}
```

Listar imagens que estão aguardando ação

Quando você usa a ação de etapa `WaitForAction` em seu fluxo de trabalho de imagem, ela pausa o fluxo de trabalho até que você envie um sinal para retomar o processamento ou falhar no fluxo de trabalho. É possível usar essa ação de etapa se houver um processo externo que precise ser executado antes de você continuar. Em seguida, você poderá usar o `SendWorkflowStepAction` para enviar um sinal para a etapa pausada para `RESUME` ou `STOP`. Também é possível interromper ou retomar seu fluxo de trabalho diretamente do console.

As guias a seguir mostram como obter uma lista de todos os recursos de imagem em sua conta com etapas do fluxo de trabalho que estejam pausadas aguardando um sinal para retomar ou parar. As guias abrangem as etapas do console e o AWS CLI comando.

Você também pode usar a API ou um SDK para obter uma lista das etapas do fluxo de trabalho que estão aguardando ação. Para a ação da API, consulte ListWaitingWorkflowStepsna Referência da API do EC2 Image Builder. Para ver a solicitação de SDK associada, consulte o link Consulte também na mesma página.

Console

Para acessar a guia Aguardando ação no console, siga estas etapas:

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Escolha Imagens no painel de navegação. Isso abrirá a página da lista Imagens.
3. Selecione a guia Aguardando ação na página da lista.

4. (opcional) Para interromper ou retomar uma etapa, marque a caixa de seleção ao lado do nome e escolha Parar etapa ou Retomar etapa. Você pode marcar mais de uma caixa de seleção para realizar a mesma ação em todas as etapas selecionadas.

Detalhes da etapa pendente do fluxo de trabalho

Os detalhes do fluxo de trabalho da etapa pendente incluem o seguinte:

- Nome da imagem: o nome do recurso de imagem que está com a etapa pendente. Você pode selecionar o link do nome para exibir a página de detalhes dessa imagem.
- Nome da etapa pendente: o nome da etapa do fluxo de trabalho que está aguardando ação.
- ID de execução da etapa: identifica de maneira exclusiva a instância de runtime da etapa do fluxo de trabalho. Você pode selecionar a ID vinculada para exibir os detalhes do runtime da etapa.
- Início da etapa: o carimbo de data e hora em que a instância de runtime da etapa de fluxo de trabalho foi iniciada.
- ARN de fluxo de trabalho: o nome do recurso da Amazon (ARN) do fluxo de trabalho com a etapa pendente.
- Ações: a ação escalonada que está em estado de espera.

AWS CLI

Ao executar o list-waiting-workflow-steps comando no AWS CLI, você receberá uma lista de todas as imagens em sua conta que têm etapas de fluxo de trabalho que aguardam ação antes de concluir o processo de criação da imagem.

O exemplo de comando a seguir mostra como usar o comando list-waiting-workflow-steps para listar todas as imagens em sua conta com etapas do fluxo de trabalho que estão aguardando ação.

Exemplo: listar imagens em sua conta com etapas de fluxo de trabalho aguardando

```
aws imagebuilder list-waiting-workflow-steps
```

Saída:

A saída deste exemplo mostra uma imagem na conta com uma etapa aguardando ação.

```
{
    "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"
        }
    ]
}
```

Listar versões da compilação de imagem

Na página Versões da compilação de imagem do console do Image Builder, você pode ver uma lista de versões da compilação e detalhes adicionais de um recurso de imagem que você possui. Você também pode usar comandos ou ações com a API Image Builder ou listar AWS CLI as versões de criação de imagens. SDKs

Você pode usar um dos métodos a seguir para listar as versões de compilação de imagem dos recursos de imagem que você possui. Para a ação da API, consulte ListImageBuildVersionsna Referência da API do EC2 Image Builder. Para ver a solicitação de SDK associada, consulte o link Consulte também na mesma página.

Console

Detalhes da versão

Os detalhes na página de Versões do Image Builder no console do Image Builder incluem o seguinte:

- Versão – A versão da compilação do recurso de imagem. No console do Image Builder, a versão é vinculada a uma página de detalhes da imagem.
- Tipo – O tipo de saída que o Image Builder distribuiu ao criar esse recurso de imagem (uma AMI ou uma imagem de contêiner).

- Data de criação – A data e a hora em que o Image Builder criou a versão de compilação da imagem.
- Status da imagem – O status atual da versão de compilação da imagem. O status pode estar relacionado à compilação ou disposição da imagem. Por exemplo, durante o processo de compilação, você pode ver um status de `Building` ou `Distributing`. Para a disposição da imagem, você pode ver um status de `Deprecated` ou `Deleted`.
- Motivo da falha – O motivo do status da imagem. O console do Image Builder somente exibe o motivo quando a compilação falha (o Status da imagem é igual a `Failed`).
- Descobertas de segurança – As descobertas agregadas da varredura de imagens para a versão de compilação da imagem referenciada.
- ARN – O nome do recurso da Amazon (ARN) para a versão do recurso de imagem referenciada.
- Fluxo de logs – Um link para os detalhes do fluxo de logs para a versão de compilação da imagem referenciada.

Listar versões

Para listar as versões de compilação de imagem no console do Image Builder, realize as seguintes etapas:

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Escolha Imagens no painel de navegação. Por padrão, a lista de imagens mostra a versão atual de cada uma das imagens que você possui.
3. Para ver uma lista de todas as versões de uma imagem, escolha o link da versão atual. O link abre a página Versões de compilação da imagem que lista todas as versões de compilação de uma imagem específica.

AWS CLI

Ao executar o list-image-build-versions comando no AWS CLI, você receberá uma lista completa das versões de compilação do recurso de imagem especificado. Você precisa ser o proprietário da imagem para executar esse comando.

O exemplo de comando a seguir mostra como usar o comando list-image-build-versions para listar todas as versões da compilação da imagem especificada.

Exemplo: listar versões da compilação para uma imagem específica

```
aws imagebuilder list-image-build-versions --image-version-arn
 arn:aws:imagebuilder:us-west-2:123456789012:image/image-recipe-name/1.0.0
```

Saída:

A saída para esse exemplo inclui duas versões de compilação para a fórmula de imagem especificada.

```
{
  "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**
>
> No momento, a saída do comando list-image-build-versions não inclui descobertas de segurança ou fluxos de logs.

Visualizar os detalhes do recurso de imagem

Na página de detalhes da imagem no console do Image Builder, você pode ver os detalhes de um recurso de imagem específico que você possui. Você também pode usar comandos ou ações com a API Image Builder ou AWS CLI para obter detalhes da imagem. SDKs

Para obter mais informações sobre recursos que outra pessoa Conta da AWS compartilhou com você por meio de um compartilhamento de recursos AWS Resource Access Manager (AWS RAM), consulte Acessar AWS recursos compartilhados com você no Guia AWS RAM do Usuário.

Conteúdo

- Ver os detalhes da imagem no console do Image Builder

- Obtenha detalhes da política de imagem no AWS CLI

Ver os detalhes da imagem no console do Image Builder

A página de detalhes da imagem no console do Image Builder inclui uma seção de resumo, com informações adicionais agrupadas em guias. O título da página é o nome e a versão de compilação da fórmula que criou a imagem. Se uma guia não se aplicar à sua imagem, ela ficará inativa e não exibirá dados.

Seções e guias de detalhes do console

- Seção de resumo
- Guia de recursos de saída
- Guia de configuração de infraestrutura
- Guia de configurações de distribuição
- Guia de fluxo de trabalho
- Guia de descobertas de segurança
- Guia de tags

Seção de resumo

A seção de resumo abrange a largura da página e inclui os detalhes a seguir. Esses detalhes são sempre exibidos.

Fórmula

O nome da fórmula e a versão que não inclui a versão de compilação. Por exemplo, se a versão de compilação for `sample-linux-recipe | 1.0.1/2`, a fórmula é `sample-linux-recipe | 1.0.1`, e a versão de compilação é 2.

Data da criação

A data e a hora em que o Image Builder criou a versão de compilação da imagem.

Status da imagem

O status atual da versão de compilação da imagem. O status pode estar relacionado à compilação ou disposição da imagem. Por exemplo, durante o processo de compilação, você

pode ver um status de `Building` ou `Distributing`. Para a disposição da imagem, você pode ver um status de `Deprecated` ou `Deleted`.

Motivo da falha

O motivo do status da imagem. O console do Image Builder somente exibe o motivo quando a compilação falha (o Status da imagem é igual a `Failed`).

Guia de recursos de saída

A guia Recursos de saída lista os detalhes de saída e distribuição do recurso de imagem exibido atualmente. As informações exibidas pelo Image Builder dependem do tipo de fórmula que o pipeline usou para criar a imagem, como segue.

Fórmula da imagem

- Região – A região de distribuição para a imagem de máquina da Amazon (AMI) de saída que é especificada na coluna Imagem.
- Imagem – O ID da AMI que o Image Builder distribuiu para o destino. Esse ID está vinculado à página Amazon Machine Images (AMIs) no EC2 console da Amazon.

> **Note**
>
> O Image Builder cria a AMI depois de criar o recurso de imagem de saída e antes de distribuir a AMI para o destino.

- Nome – O nome da AMI que o Image Builder distribuiu para o destino.
- Descrição – A descrição opcional da fórmula da imagem que o pipeline usou para criar o recurso de imagem de saída.
- Conta — A Conta da AWS que possui o recurso de imagem do Image Builder exibido atualmente.

Fórmula do contêiner

O Image Builder exibe os seguintes detalhes da saída criada a partir de uma fórmula de contêiner.

- Região – A região de distribuição para a imagem do contêiner que é especificada na coluna URI da imagem.

- URI da imagem – O URI da imagem do contêiner de saída que o Image Builder distribuiu para o repositório ECR na região de destino.

> **Note**
>
> O Image Builder exibe uma linha por destino. A imagem de saída sempre tem pelo menos uma entrada para distribuição para a conta que criou a imagem. Destinos adicionais podem incluir distribuições entre regiões, Contas da AWS, ou AWS Organizations. Para obter mais informações, consulte Gerenciar as configurações de distribuição do Image Builder.

Guia de configuração de infraestrutura

A guia Configuração de infraestrutura exibe as configurações de EC2 infraestrutura da Amazon que o Image Builder usou para criar e testar a imagem exibida atualmente. O Image Builder sempre exibe o nome do recurso de configuração de infraestrutura (Nome da configuração) e seu nome do recurso da Amazon (ARN). Se a configuração da sua infraestrutura definir os valores, detalhes adicionais da infraestrutura podem incluir o seguinte

- Tipos de instância
- Um perfil de instância
- Infraestrutura de rede
- Configurações do grupo de segurança
- Um local do Amazon S3 onde o Image Builder armazena os logs do aplicativo
- Um par de EC2 chaves da Amazon para solução de problemas
- Um tópico do Amazon SNS para notificações de evento

Para obter mais informações, consulte Gerenciar a configuração da infraestrutura do Image Builder.

Guia de configurações de distribuição

A guia Configurações de distribuição exibe as configurações que o Image Builder usou para distribuir suas imagens de saída. O Image Builder sempre exibe o nome do recurso de configuração de distribuição (Nome da configuração) e seu nome do recurso da Amazon (ARN). Detalhes adicionais da distribuição dependem do tipo de fórmula que o pipeline do Image Builder usou para criar a imagem, da seguinte forma:

Fórmula da imagem

Se seu recurso de configuração de distribuição definir os valores, os detalhes adicionais da distribuição podem incluir o seguinte:

- Região – A região de distribuição para a imagem de máquina da Amazon (AMI) de saída.
- Nome da AMI de saída – O nome da AMI que o Image Builder distribuiu para o destino.
- Criptografia (chave KMS) – Se configurada, a AWS KMS key que o Image Builder usa para criptografar a imagem para distribuição para a região de destino.
- Contas de destino para distribuição — Se você configurou a distribuição entre contas, essa coluna exibirá uma lista separada por vírgulas das Contas da AWS quais compartilhar a imagem de saída na região de destino.
- Diretores com permissão compartilhada — Uma lista separada por vírgulas dos AWS diretores que têm permissão para lançar sua imagem, por exemplo, Contas da AWS ou grupos ou unidades organizacionais (). AWS Organizations OUs

  > **Note**
  >
  > Quando você concede permissão para outros diretores lançarem sua imagem, você ainda é o proprietário da imagem. AWS cobra da sua conta todas as instâncias que a Amazon EC2 lança a partir da sua imagem.

- Contas de destino para uma configuração de lançamento mais rápida — Contas da AWS onde o EC2 Fast Launch distribui instantâneos pré-provisionados para o lançamento.
- Configurações de licença associadas — A configuração de licença do ARNs License Manager associada à AMI na região especificada.
- Configuração do modelo de lançamento — Identifica um modelo de EC2 lançamento da Amazon para usar em uma conta específica.
- Definir versão padrão do modelo de lançamento — Define o modelo de EC2 lançamento da Amazon especificado como o modelo de lançamento padrão para o especificado Conta da AWS.

Fórmula do contêiner

As distribuições do contêiner sempre incluem os seguintes detalhes:

- Região – A região de distribuição para a imagem do contêiner especificada na coluna URI da imagem.

- URI da imagem – O URI da imagem do contêiner de saída que o Image Builder distribuiu para o repositório Amazon ECR na região de destino.

> **Note**
>
> O Image Builder exibe uma linha por destino. A imagem de saída sempre tem pelo menos uma entrada para distribuição para a conta que criou a imagem. Destinos adicionais podem incluir distribuições entre regiões, Contas da AWS, ou AWS Organizations. Para obter mais informações, consulte Gerenciar as configurações de distribuição do Image Builder.

Guia de fluxo de trabalho

Os fluxos de trabalho definem a sequência de etapas que o Image Builder realiza ao criar uma nova imagem. Todas as imagens têm fluxos de trabalho de criação, teste e distribuição. A guia Fluxo de trabalho exibe os fluxos de trabalho aplicáveis que o Image Builder executou para sua imagem.

Filtrar tipos de fluxo de trabalho

O Image Builder exibe inicialmente o resumo do fluxo de trabalho de criação ou importação e as etapas do fluxo de trabalho por padrão. No entanto, o filtro Fluxo de trabalho mostra todos os fluxos de trabalho que estão em andamento ou concluídos para sua imagem. Para ver um fluxo de trabalho diferente, selecione na lista.

Os fluxos de trabalho de imagem que produzem saída de AMI podem ter fluxos de trabalho de criação, importação ou teste. Os fluxos de trabalho de contêineres que produzem saída de contêiner podem ter fluxos de trabalho de criação, teste ou distribuição.

> **Note**
>
> Se o fluxo de trabalho ainda não tiver sido iniciado, ele não aparecerá na lista. Por exemplo, se sua criação de imagem com fluxos de trabalho de criação e teste configurados acabou de começar, o fluxo de trabalho de criação é o único tipo de fluxo de trabalho que aparece na lista. Quando o fluxo de trabalho de teste começa, o Image Builder o adiciona à lista.

Após o filtro Fluxo de trabalho, o fluxo de trabalho selecionado mostra um resumo do runtime que inclui os seguintes detalhes para cada tipo de fluxo de trabalho:

Status do fluxo de trabalho

O status do runtime atual desse fluxo de trabalho. Os valores podem incluir os seguintes:

- Pendente
- Ignorado
- Em execução
- Concluído
- Falha
- R ollback-in-progress
- Reversão concluída

ID de execução

Um identificador exclusivo que o Image Builder atribui para rastrear os recursos de runtime cada vez que ele que executa um fluxo de trabalho.

Início

O carimbo de data e hora em que a instância de runtime desse fluxo de trabalho foi iniciada.

Fim

O carimbo de data e hora em que essa instância de runtime do fluxo de trabalho foi concluída.

Total de etapas

O número total de etapas no fluxo de trabalho. Isso deve ser igual à soma das contagens de etapas que tiveram êxito, que foram ignoradas e que falharam.

Etapas que tiveram êxito

Uma contagem do runtime para o número de etapas no fluxo de trabalho que foram executadas com êxito.

Etapas que falharam

Uma contagem do runtime para o número de etapas no fluxo de trabalho que falharam.

Etapas que foram ignoradas

Uma contagem do runtime para o número de etapas no fluxo de trabalho que foram ignoradas.

Os detalhes na lista a seguir relatam o status atual de todas as etapas nessa instância de runtime do fluxo de trabalho. O Image Builder exibe os mesmos detalhes para todos os tipos de imagem.

N.º da etapa

Um número que representa a ordem na qual o Image Builder executa as etapas do fluxo de trabalho.

ID da etapa

Um identificador exclusivo para a etapa do fluxo de trabalho, atribuída no runtime.

Status da etapa

O status do runtime atual da etapa do fluxo de trabalho especificada.

Status de reversão

O status de reversão atual se essa instância de runtime do fluxo de trabalho falhar.

Nome da etapa

O nome da etapa do fluxo de trabalho especificado.

Início

O carimbo de data e hora em que a etapa especificada para essa instância de runtime do fluxo de trabalho foi iniciada.

Fim

O carimbo de data e hora em que a etapa especificada para essa instância de runtime do fluxo de trabalho foi concluída.

Guia de descobertas de segurança

Se você ativou a verificação, a guia Descobertas de segurança exibirá as descobertas de Vulnerabilidades e Exposições Comuns (CVE). O Amazon Inspector identificou essas descobertas na instância de teste que o Image Builder iniciou para criar sua nova imagem. Para garantir que o Image Builder capture as descobertas para a sua imagem, você deve configurar a verificação da seguinte forma:

1. Ative os escaneamentos do Amazon Inspector para sua conta. Para obter mais informações, consulte Conceitos básicos do Amazon Inspector no Guia do usuário do Amazon Inspector.
2. Ative as descobertas de segurança para o pipeline que cria essa imagem. Quando você ativa as descobertas de segurança para seu pipeline, o Image Builder salva um snapshot das

descobertas antes de encerrar a instância de teste. Para obter mais informações, consulte Configure escaneamentos de segurança para imagens do Image Builder no Console de gerenciamento da AWS.

A guia Descobertas de segurança inclui os seguintes detalhes para cada vulnerabilidade que o Amazon Inspector identificou para sua imagem.

Gravidade

O nível de gravidade da descoberta de CVE. Os valores são os seguintes:

- Não triado
- Informativo
- Baixo
- Médio
- Alto
- Crítico

ID da descoberta

O identificador exclusivo da descoberta do CVE que o Amazon Inspector detectou para sua imagem ao verificar a instância de teste. O ID está vinculado à página Descobertas de segurança > Por vulnerabilidade. Para obter mais informações, consulte Gerencie descobertas de segurança para imagens do Image Builder no Console de gerenciamento da AWS.

Origem

A fonte das informações de vulnerabilidade para a descoberta do CVE.

Idade

O número de dias desde que a descoberta foi observada pela primeira vez em sua imagem.

Pontuação do Inspector

A pontuação que o Amazon Inspector atribuiu para a descoberta do CVE.

Guia de tags

A guia Tags exibe todas as tags que você definiu para sua imagem.

Obtenha detalhes da política de imagem no AWS CLI

O exemplo a seguir mostra como obter os detalhes de uma política de imagens com seu nome do recurso da Amazon (ARN).

```
aws imagebuilder get-image-policy --image-arn arn:aws:imagebuilder:us-
west-2:123456789012:image/example-image/2019.12.02
```

Criar imagens personalizadas com o Image Builder

Existem várias maneiras diferentes de criar uma nova imagem do Image Builder. Por exemplo, você pode usar um dos métodos a seguir para criar uma imagem com o Console de gerenciamento da AWS ou AWS CLI. Você também pode usar a ação da CreateImageAPI ou executar um pipeline de construção para criar a imagem. Para a solicitação do SDK associada à ação da API, você pode consultar o link Veja também desse comando na Referência da API do EC2 Image Builder.

Console de gerenciamento da AWS

Para criar uma nova imagem com base em um pipeline existente, você pode executar manualmente o pipeline como segue. Você também pode usar o assistente de pipeline para criar uma nova imagem a partir do zero. Veja Assistente de pipeline: criar AMI ou Assistente de pipeline: criar imagem de contêiner, dependendo do tipo de imagem que deseja criar.

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. No painel de navegação, selecione Pipelines de imagem.
3. Marque a caixa de seleção ao lado do nome do pipeline que você deseja executar.
4. Para criar a imagem, selecione Executar pipeline no menu Ações. Isto inicia o pipeline.

Você também pode especificar um cronograma para executar seu pipeline ou usar EventBridge a Amazon para executar seu pipeline com base nas regras que você configura.

AWS CLI

Antes de executar o create-image comando no AWS CLI, você deve criar os seguintes recursos, caso eles ainda não existam:

Recursos necessários

- Fórmula: você deve especificar exatamente uma fórmula para sua imagem, da seguinte forma:

Fórmula da imagem

Especifica o nome do recurso da Amazon (ARN) para seu recurso de fórmula de imagem com o parâmetro `--image-recipe-arn`.

Fórmula do contêiner

Especifique o ARN do seu recurso de fórmula de contêiner com o parâmetro `--container-recipe-arn`.

- Configuração de infraestrutura: especifique o ARN para seu recurso de configuração de infraestrutura com o parâmetro `--infrastructure-configuration-arn`.

Você também pode especificar qualquer um dos seguintes recursos que sua imagem requer:

Recursos e configuração opcionais

- Configuração de distribuição: por padrão, o Image Builder distribui o recurso de imagem de saída para sua conta na região em que você executa o comando create-image. Para fornecer destinos ou configurações adicionais para sua distribuição, especifique o ARN do seu recurso de configuração de distribuição com o parâmetro `--distribution-configuration-arn`.
- Verificação de imagens: para configurar instantâneos das descobertas do Amazon Inspector em sua instância de teste de imagem ou contêiner, use o parâmetro `--image-scanning-configuration`. Para imagens de contêineres, você também especifica o repositório ECR que o Amazon Inspector usa para suas verificações.
- Testes de imagem: para suprimir o estágio de teste do Image Builder, use o parâmetro `--image-tests-configuration`. Como alternativa, você pode definir um tempo limite para execução.
- Tags de imagem: use o parâmetro `--tags` para adicionar tags ao recurso de imagem de saída.
- Fluxos de trabalho de imagem: se você não especificar nenhum fluxo de trabalho de compilação ou teste, o Image Builder criará sua imagem com seu fluxo de trabalho de imagem padrão. Para especificar fluxos de trabalho que você criou, use o parâmetro `--workflows`.

> **Note**
>
> Se você especificar fluxos de trabalho de imagem, também deverá fornecer o nome ou o ARN do perfil do IAM que o Image Builder usa para executar suas ações de fluxo de trabalho no parâmetro `--execution-role`.

O exemplo a seguir mostra como criar uma imagem com o comando create-image da AWS CLI . Para obter mais informações, consulte Referência de comandos da AWS CLI .

Exemplo: criar uma imagem básica com distribuição padrão

```
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
```

Saída:

```
{
"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",
        ...
    }
]
}
```

Cancelar a criação da imagem a partir do AWS CLI

Para cancelar uma compilação de imagem em andamento, use o comando cancel-image-creation da seguinte forma:

```
aws imagebuilder cancel-image-creation --image-build-version-arn
 arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-recipe/2019.12.03/1
```

Importar e exportar imagens de máquina virtual com o Image Builder

Quando você exporta uma VM de seu ambiente de virtualização, esse processo cria um conjunto de um ou mais arquivos de contêiner de disco que atuam como instantâneos do ambiente, das configurações e dos dados da sua VM. Você pode usar esses arquivos para importar sua VM e usá-la como imagem base para suas fórmulas de imagens. Para exportar, você pode criar arquivos de disco VM como saída da sua criação de imagem personalizada e distribuir os arquivos.

O Image Builder é compatível com os seguintes formatos de arquivo para seus contêineres de disco de VM:

- Arquivo de virtualização aberto (OVA)
- Disco de máquina virtual (VMDK)
- Disco rígido virtual (VHD/VHDX)
- Bruto

A importação usa os discos para criar uma imagem de máquina da Amazon (AMI) e um recurso de imagem Image Builder, ambos os quais podem servir como imagem base para sua fórmula de imagem personalizada. Os discos da VM devem ser armazenados em buckets do S3 para a importação. Você também pode importar de um snapshot de EBS existente.

No console do Image Builder, você pode importar a imagem diretamente e, em seguida, usar a imagem de saída ou a AMI em suas fórmulas, ou você pode especificar parâmetros de importação ao criar sua fórmula ou versão da fórmula. Para obter mais informações sobre a importação como parte de sua fórmula de imagem, consulte Configuração de importação de VM.

Importar uma VM para o Image Builder

O Image Builder se integra à Import/Export API da Amazon EC2 VM para permitir que o processo de importação seja executado de forma assíncrona em segundo plano. O Image Builder faz referência ao ID da tarefa de importação da VM para monitorar seu progresso e cria um recurso de imagem do Image Builder como saída. Isso permite que você faça referência ao recurso de imagem do Image Builder em suas fórmulas antes que a importação da VM termine.

Console

Para importar uma VM com o console do Image Builder, siga estas etapas:

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Escolha Imagens no painel de navegação.
3. Para abrir a caixa de diálogo de importação, escolha Importar imagem.
4. Insira as seguintes informações gerais:
   - Especifique um nome exclusivo para sua imagem.
   - Especifique uma Versão para a imagem de base. Use o formato a seguir: *major*.*minor*.*patch*.
5. Escolha o tipo de importação: importação de VM.
6. Forneça detalhes para cada uma das seções a seguir na página Importar imagem. Em seguida, selecione Import image ao concluir.

Sistema operacional da imagem de base

1. Selecione a opção Image Operating System (OS) que corresponda à sua plataforma VM OS.
2. Selecione a OS version (versão do sistema operacional) que corresponde à versão da sua VM na lista.

Configuração de importação de VM

1. Quando você exporta sua VM do ambiente de virtualização, esse processo cria um conjunto de um ou mais arquivos de contêiner de disco. Eles atuam como instantâneos do ambiente, das configurações e dos dados da sua VM. Você pode usar esses arquivos para importar sua VM como imagem de base para sua fórmula de imagem. Para obter mais informações sobre importação VMs no Image Builder, consulteImportar e exportar imagens de VM.

   Para especificar a localização da sua fonte de importação, siga estas etapas:

   Importar fonte

   Especifique a fonte do primeiro contêiner de disco ou snapshot de imagem da VM a ser importado na seção Contêiner de disco 1.

   a. Fonte: essa pode ser um bucket do S3 ou um snapshot do EBS.

b. Selecione a localização do disco do S3: insira a localização no Amazon S3 onde suas imagens de disco estão armazenadas. Para procurar o local, escolha Browse S3.

c. Para adicionar um contêiner de disco, escolha Adicionar contêiner de disco.

2. Perfil do IAM

   Para associar um Perfil do IAM à sua configuração de importação de VM, selecione o perfil na lista suspensa de Perfil do IAM ou escolha Criar novo perfil para criar um novo. Se você criar um novo perfil, a página do console de Perfis do IAM será aberta em uma guia separada.

3. Configurações avançadas: opcionais

   As seguintes configurações são opcionais. Com essas configurações, você pode configurar criptografia, licenciamento, tags e muito mais para a imagem de base criada pela importação.

   Arquitetura da imagem de base

   Para especificar a arquitetura da sua fonte de importação de VM, selecione um valor na lista Arquitetura.

   Criptografia

   Se as imagens de disco da VM estiverem criptografadas, você deverá fornecer uma chave para usar no processo de importação. Para especificar um para a importação, selecione um valor na lista Criptografia (chave KMS). A lista contém chaves KMS às quais sua conta tem acesso na região atual.

   Gerenciamento de licença

   Quando você importa uma VM, o processo de importação detecta automaticamente o OS da VM e aplica a licença apropriada à imagem de base. Dependendo da plataforma do seu OS, os tipos de licença são os seguintes:

   - Licença incluída: Uma licença AWS apropriada para sua plataforma é aplicada à sua imagem de base.

- Traga a sua própria licença (BYOL): Retém a licença da sua VM, se aplicável.

Para anexar as configurações de licença criadas com AWS License Manager à sua imagem base, selecione na lista Nome da configuração da licença. Para obter mais informações sobre o License Manager, consulte Trabalhando com AWS License Manager

> **Note**
>
> - As configurações de licença contêm regras de licenciamento com base nos termos de seus contratos empresariais.
> - O Linux só oferece suporte a licenças BYOL.

Tags (imagem de base)

As tags usam pares de chave–valor para atribuir texto pesquisável ao seu recurso do Image Builder. Para especificar tags para a imagem de base importada, insira pares de chave–valor com as caixas Chave e Valor.

Para adicionar uma tag, escolha Adicionar tag. Para remover uma tag, selecione Remover tag.

AWS CLI

Para importar uma VM de discos para uma AMI e criar um recurso de imagem do Image Builder que você possa referir imediatamente, siga estas etapas na AWS CLI:

1. Inicie uma importação de VM, com o Import/Export import-image comando Amazon EC2 VM no. AWS CLI Anote a ID de tarefa que for retornada na resposta de comando. Você precisará dela para a próxima etapa. Para obter mais informações, consulte Importando uma VM como imagem usando o VM Import/Export no Guia do usuário da VM. Import/Export

2. Criar um arquivo JSON de entrada da CLI

Para simplificar o import-vm-image comando Image Builder usado no AWS CLI, criamos um arquivo JSON que contém toda a configuração de importação que queremos passar para o comando.

> **Note**
>
> A convenção de nomenclatura para os valores de dados no arquivo JSON segue o padrão especificado para os parâmetros de solicitação de operação da API Image Builder. Para revisar os parâmetros da solicitação de operação da API, consulte a ImportVmImageoperação na Referência da API EC2 Image Builder.
> Para fornecer os valores dos dados como parâmetros da linha de comando, consulte os nomes dos parâmetros especificados na Referência de comando do AWS CLI para o comando do import-vm-image como opções.

Aqui está um resumo dos parâmetros que especificamos nestes exemplos:

- name (string, obrigatório): o nome do recurso de imagem do Image Builder a ser criado como saída da importação.
- SemanticVersion (string, obrigatório): a versão semântica da imagem de saída que especifica a versão no formato a seguir, com valores numéricos em cada posição para indicar uma versão específica: <major>.<minor>.<patch>. Por exemplo, `.1.0.0` Para saber mais sobre o versionamento semântico dos recursos do Image Builder, consulte Versionamento semântico no Image Builder.
- description (string): a descrição da fórmula de imagem.
- platform (string, obrigatório): a plataforma do sistema operacional para a VM importada.
- vmImportTaskId (string, obrigatório) — O `ImportTaskId` (AWS CLI) do processo de importação da Amazon EC2 VM. O Image Builder monitora o processo de importação para extrair a AMI que ele cria e criar um recurso de imagem do Image Builder que pode ser usado em fórmulas imediatamente.
- tags (string map): as tags são pares de chave-valor anexados aos recursos de importação. São permitidos até 50 pares chave-valor.

Salve o arquivo como `import-vm-image.json`, para usar no comando import-vm-image do Image Builder.

```
{
    "name": "example-request",
    "semanticVersion": "1.0.0",
    "description": "vm-import-test",
    "platform": "Linux",
    "vmImportTaskId": "import-ami-01ab234567890cd1e",
    "tags": {
     "Usage": "VMIE"
    }
}
```

3. Importar a imagem

   Execute o comando import-vm-image com o arquivo que você criou como entrada:

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

   > **Note**
   >
   > - É necessário incluir a notação `file://` no início do caminho do arquivo JSON.
   > - O caminho para o arquivo JSON deve seguir a convenção apropriada para o sistema operacional de base no qual você está executando o comando. Por exemplo, o Windows usa a barra invertida (\) para se referir ao caminho do diretório, enquanto o Linux e o macOS usam a barra (/).

Distribua discos VM de sua imagem, crie a partir do AWS CLI

Você pode configurar a distribuição de arquivos de formato de disco VM compatíveis para buckets do S3 nas regiões de destino como parte do seu processo normal de criação de imagens, usando as configurações de distribuição do Image Builder no. AWS CLI Para obter mais informações, consulte Exemplo: Crie configurações de distribuição para discos VM de saída do AWS CLI.

Importe imagens de disco ISO verificadas do Windows com o Image Builder

Um arquivo ISO do sistema operacional Windows é um arquivo de imagem de disco que contém o pacote de instalação completo para uma versão específica do sistema operacional Windows. A Microsoft fornece arquivos ISO oficiais do sistema operacional Windows para download, diretamente do site ou por meio de revendedores autorizados. É importante garantir que você obtenha os arquivos ISO de uma fonte confiável e legítima para evitar possíveis malwares ou versões não autorizadas.

EC2 O Image Builder usa o `build-image-from-iso` fluxo de trabalho de importação para importar o arquivo de disco ISO e criar um volume secundário a partir dele. Após a conclusão da configuração, o Image Builder tira um instantâneo do volume criado a partir da importação e o usa para criar uma Amazon Machine Image (AMI).

Sistemas operacionais compatíveis para importação de imagem de disco ISO

O Image Builder é compatível com as seguintes imagens de disco ISO do sistema operacional Windows:

- Windows 11 Enterprise versão 25H2 (x64)
- Windows 11 Enterprise versão 24H2 (x64)
- Windows 11 Enterprise versão 23H2 (x64)

O Image Builder não oferece suporte às seguintes imagens de disco ISO do sistema operacional Windows:

- Imagens do Long-Term Servicing Channel (LTSC)
- Imagens de disco ISO criadas a partir da Ferramenta de Criação do Windows Media
- Imagens de avaliação

Pré-requisitos para importar uma imagem de disco ISO

Para importar uma imagem de disco ISO, primeiro você deve atender aos seguintes pré-requisitos:

- O sistema operacional da imagem do disco deve ser compatível com o Image Builder. Para obter uma lista dos sistemas operacionais compatíveis, consulteSistemas operacionais compatíveis para importação de imagem de disco ISO.
- Para garantir que você possa importar sua imagem ISO, baixe-a do centro de administração do Microsoft 365.
- Antes de executar o processo de importação, você deve carregar seu arquivo de disco ISO para o Amazon S3 no mesmo Conta da AWS local em Região da AWS que a importação é executada.
- A extensão do arquivo diferencia maiúsculas de minúsculas para o processo de importação e deve ser `.ISO`. Se a extensão do arquivo estiver em minúsculas, você poderá executar um dos seguintes comandos para renomeá-la:

  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
  ```

- O licenciamento da Microsoft não é incluído automaticamente na importação. Você deve trazer sua própria licença (BYOL). Para obter mais informações sobre o licenciamento de software da Microsoft, consulte Licenciamento na página Amazon Web Services e Perguntas frequentes da Microsoft.
- O processo de importação usa duas funções separadas do IAM, da seguinte forma:

  Perfil de execução

  Essa função concede permissão para o Image Builder ligar Serviços da AWS em seu nome. Você pode especificar a função AWSServiceRoleForImageBuildervinculada ao serviço, que inclui as permissões necessárias para a função de execução, ou você pode criar sua própria função.

  Função do perfil da instância

  Esse papel concede permissão para as ações que o serviço executa na EC2 instância. Você pode especificar uma função de perfil de instância em seu recurso de configuração de infraestrutura. Anexe as seguintes políticas gerenciadas à sua função de perfil de instância para garantir que você tenha todas as permissões necessárias para o processo de importação.

- EC2InstanceProfileForImageBuilder
- AmazonSSMManagedInstanceCore

Para obter mais informações, consulte Gerenciar a configuração da infraestrutura do Image Builder.

Importar uma imagem de disco ISO para o Image Builder

Antes de iniciar o processo de importação, verifique se você atendeu a todos osPré-requisitos.

O processo de importação instala os seguintes softwares e drivers em sua imagem:

- EC2Inicie a v2
- AWS Systems Manager agente
- AWS NVMe motorista
- AWS Driver de rede ENA
- AWS Controlador serial PCI
- EC2 Driver de utilitário do Windows
- Kit de atualização do Microsoft Defender

O processo de importação faz as seguintes atualizações de configuração em sua imagem:

- Configura o sistema para usar o servidor Amazon Time.

Console

Para importar uma imagem de disco ISO com o console do Image Builder, siga estas etapas:

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Escolha Imagens no painel de navegação.
3. Para abrir a caixa de diálogo de importação, escolha Importar imagem.
4. Insira as seguintes informações gerais:
   - Especifique um nome exclusivo para sua imagem.
   - Especifique uma Versão para a imagem de base. Use o formato a seguir: *major.minor.patch*.

5. Escolha o tipo de importação: importação ISO.
6. Insira os seguintes detalhes da configuração de importação ISO. Em seguida, selecione Import image ao concluir.

   - URI do S3 — Insira o local onde seu arquivo de disco ISO está armazenado. Para procurar o arquivo, escolha Browse S3.
   - Função do IAM — Para associar uma função do IAM à sua configuração de importação, selecione a função na lista suspensa da função do IAM ou escolha Criar nova função para criar uma nova. Se você criar um novo perfil, a página do console de Perfis do IAM será aberta em uma guia separada.

     Você pode especificar a função AWSServiceRoleForImageBuildervinculada ao serviço ou especificar sua própria função personalizada para acesso ao serviço.
7. Opcionalmente, você pode adicionar tags ao seu recurso de imagem do Image Builder. Isso não adiciona as tags à sua AMI.
8. A configuração da infraestrutura ISO define as configurações da instância que o Image Builder inicia para hospedar o processo de importação. Você pode usar uma configuração de infraestrutura criada pelo Image Builder, com base nos padrões de serviço, ou pode usar uma configuração de infraestrutura existente. Para obter mais informações, consulte Gerenciar a configuração da infraestrutura do Image Builder.

   Para criar uma nova configuração de infraestrutura, escolha Criar configuração de infraestrutura. Isso abre em uma guia separada. Ao terminar de criar o novo recurso, você pode retornar à configuração de importação e escolher Usar configuração de infraestrutura existente.
9. Para iniciar o processo de importação, escolha Importar imagem.

Depois que a importação for concluída, sua imagem aparecerá na lista de imagens que você possui. Consulte mais detalhes em Listar imagens.

AWS CLI

Este exemplo mostra como importar uma imagem de um arquivo de disco ISO e criar uma AMI a partir dela com AWS CLI o.

Aqui está um resumo dos parâmetros que especificamos nestes exemplos:

- name (string, obrigatório): o nome do recurso de imagem do Image Builder a ser criado como saída da importação.
- SemanticVersion (string, obrigatório): a versão semântica da imagem de saída que especifica a versão no formato a seguir, com valores numéricos em cada posição para indicar uma versão específica: <major>.<minor>.<patch>. Por exemplo, `.1.0.0` Para saber mais sobre o versionamento semântico dos recursos do Image Builder, consulte Versionamento semântico no Image Builder.
- description (string): a descrição da fórmula de imagem.
- ExecutionRole (string) — O nome ou Amazon Resource Name (ARN) da função do IAM que concede ao Image Builder acesso para realizar ações de fluxo de trabalho para importar uma imagem de um arquivo ISO da Microsoft. Você pode especificar a função AWSServiceRoleForImageBuildervinculada ao serviço ou especificar sua própria função personalizada para acesso ao serviço.
- platform (string, required) — A plataforma do sistema operacional para a imagem de disco ISO. Os valores válidos são: `Windows`.
- osVersion (string, required) — A versão do sistema operacional para a imagem de disco ISO. Os valores válidos são: `Microsoft Windows 11`.
- infrastructureConfigurationArn(string, required) — O Amazon Resource Name (ARN) do recurso de configuração de infraestrutura usado para iniciar a EC2 instância na qual a imagem ISO é criada.
- uri (string, required) — O URI do arquivo de disco ISO armazenado no 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"
```

Depois que a importação for concluída, sua imagem aparecerá na lista de imagens que você possui. Consulte mais detalhes em Listar imagens.

PowerShell

Este exemplo mostra como importar uma imagem de um arquivo de disco ISO e criar uma AMI a partir dela com PowerShell.

Aqui está um resumo dos parâmetros que especificamos nestes exemplos:

- name (string, obrigatório): o nome do recurso de imagem do Image Builder a ser criado como saída da importação.
- SemanticVersion (string, obrigatório): a versão semântica da imagem de saída que especifica a versão no formato a seguir, com valores numéricos em cada posição para indicar uma versão específica: <major>.<minor>.<patch>. Por exemplo, `.1.0.0` Para saber mais sobre o versionamento semântico dos recursos do Image Builder, consulte Versionamento semântico no Image Builder.
- description (string): a descrição da fórmula de imagem.
- ExecutionRole (string) — O nome ou Amazon Resource Name (ARN) da função do IAM que concede ao Image Builder acesso para realizar ações de fluxo de trabalho para importar uma imagem de um arquivo ISO da Microsoft. Você pode especificar a função AWSServiceRoleForImageBuildervinculada ao serviço ou especificar sua própria função personalizada para acesso ao serviço.
- platform (string, required) — A plataforma do sistema operacional para a imagem de disco ISO. Os valores válidos são: `Windows`.
- osVersion (string, required) — A versão do sistema operacional para a imagem de disco ISO. Os valores válidos são: `Microsoft Windows 11`.
- infrastructureConfigurationArn(string, required) — O Amazon Resource Name (ARN) do recurso de configuração de infraestrutura usado para iniciar a EC2 instância na qual a imagem ISO é criada.
- uri (string, required) — O URI do arquivo de disco ISO armazenado no 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"
```

Depois que a importação for concluída, sua imagem aparecerá na lista de imagens que você possui. Consulte mais detalhes em Listar imagens.

Execute uma instância a partir da AMI de saída

Quando você executa uma instância de uma AMI criada pelo processo de importação, o sistema operacional Windows executa o Sysprep Specialize, que baixa e instala automaticamente o EC2 Launch v2 e o Systems Manager Agent a partir de endpoints públicos do S3. Esses endpoints exigem acesso público à Internet. Se você iniciar uma instância de uma sub-rede privada, o processo Sysprep Specialize não conseguirá acessar os endpoints do S3 e a inicialização falhará.

Gerenciar descobertas de segurança para imagens do Image Builder

Quando você ativa a verificação de segurança com o Amazon Inspector, ele verifica continuamente as imagens da máquina e as instâncias em execução na sua conta em busca de vulnerabilidades do sistema operacional e da linguagem de programação. Se ativada, a verificação de segurança será executada automaticamente e o Image Builder poderá salvar um snapshot das descobertas da sua instância de teste quando você criar uma nova imagem. O Amazon Inspector é um serviço pago.

Quando o Amazon Inspector descobre vulnerabilidades em suas configurações de software ou rede, ele executa as seguintes ações:

- Notifica você de que houve uma descoberta.
- Classifica a severidade da descoberta. A classificação de gravidade categoriza as vulnerabilidades para ajudá-lo a priorizar suas descobertas e inclui os seguintes valores:
  - Não triado
  - Informativo
  - Baixo
  - Médio
  - Alto

  - Crítico
- Fornece informações sobre a descoberta e links para recursos adicionais para obter mais detalhes.
- Oferece orientação de reparos para ajudá-lo a resolver os problemas que geraram a descoberta.

Configure escaneamentos de segurança para imagens do Image Builder no Console de gerenciamento da AWS

Se você ativou o Amazon Inspector para sua conta, o Amazon Inspector digitaliza automaticamente as instâncias EC2 que o Image Builder lança para criar e testar uma nova imagem. Essas instâncias têm uma vida útil curta durante o processo de compilação e teste, e suas descobertas normalmente expiram assim que essas instâncias são encerradas. Para ajudar você a investigar e corrigir as descobertas de sua nova imagem, o Image Builder pode, opcionalmente, salvar como um snapshot todas as descobertas que o Amazon Inspector identificou em sua instância de teste durante o processo de compilação.

Etapa 1: ativar as verificações de segurança do Amazon Inspector para sua conta

Para ativar as verificações de segurança do Amazon Inspector para sua conta a partir do console do Image Builder, siga estas etapas:

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. No painel de navegação, escolha Configurações da verificação de segurança. Isto abre a caixa de diálogo Verificação de segurança.

   A caixa de diálogo exibe o status de digitalização da sua conta. Se o Amazon Inspector já estiver ativado para sua conta, o status mostrará Habilitado.
3. Siga as etapas 1 e 2 das instruções para ativar a verificação do Amazon Inspector.

> **Note**
>
> O Amazon Inspector gera cobranças. Para obter mais informações, consulte a Definição de preço do Amazon Inspector.

Se você ativou a verificação para seu pipeline, o Image Builder tira um snapshot das descobertas da sua instância de compilação ao criar uma nova imagem. Dessa forma, você pode acessar as descobertas depois que o Image Builder encerrar a instância de compilação.

Etapa 2: configurar seu pipeline para salvar snapshots das descobertas de vulnerabilidades

Para configurar snapshots da descoberta de vulnerabilidades para seu pipeline, faça o seguinte:

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. No painel de navegação, selecione Pipelines de imagem.
3. Escolha um dos seguintes métodos para especificar detalhes do pipeline:

   Crie um pipeline

   1. Na página Pipelines de imagens, escolha Criar pipeline de imagem. Isso abre a página Especificar detalhes do pipeline no assistente de pipeline.

   Atualizar um pipeline existente

   1. Na página Pipelines de imagens, escolha o link do nome do pipeline para o pipeline que você deseja atualizar. Isso abre uma visão detalhada do pipeline.

      > **Note**
      >
      > Como alternativa, você pode marcar a caixa de seleção ao lado do nome do pipeline que você deseja atualizar e, em seguida, selecione Exibir detalhes.

   2. Na página de detalhes do pipeline, selecione Editar pipeline no menu Ação. Isso leva você para a página de Editar pipeline.
4. Na seção Geral do assistente de pipeline ou na página Editar pipeline, marque a caixa de seleção Habilitar verificação de segurança.

   > **Note**
   >
   > Se quiser desativar os snapshots posteriormente, você pode editar seu pipeline para desmarcar a caixa de seleção. Isso não desativa a verificação do Amazon Inspector para sua conta. Para desativar a verificação do Amazon Inspector, consulte Desativar o Amazon Inspector no Guia do Usuário do Amazon Inspector.

Gerencie descobertas de segurança para imagens do Image Builder no Console de gerenciamento da AWS

As páginas da lista de descobertas de segurança exibem informações de alto nível sobre as descobertas para seus recursos, com visualizações baseadas em vários filtros diferentes que você pode aplicar. Cada exibição inclui as seguintes opções na parte superior para alterar sua visualização:

- Todas as descobertas de segurança: essa é a exibição padrão se você escolher a página Descobertas de segurança no painel de navegação no console do Image Builder.
- Por vulnerabilidade: essa visualização mostra uma lista de alto nível de todos os recursos de imagem em sua conta que têm descobertas. O ID da descoberta está vinculado a informações mais detalhadas sobre a descoberta. Estas informações aparecem em um painel que se abre no lado direito da página. O painel inclui as seguintes informações:
  - Uma descrição detalhada da descoberta.
  - Uma guia de detalhes da descoberta. Essa guia inclui uma visão geral da descoberta, pacotes afetados, conselhos resumidos sobre reparos, detalhes da vulnerabilidade e vulnerabilidades relacionadas. O ID de vulnerabilidade está vinculado a informações detalhadas sobre vulnerabilidades no National Vulnerability Database (Banco de dados nacional de vulnerabilidades).
  - Uma guia de detalhamento da pontuação. Essa guia inclui uma side-by-side comparação das pontuações do CVSS e do Amazon Inspector para que você possa ver onde o Amazon Inspector modificou uma pontuação, se aplicável.
- Por pipeline de imagens: essa visualização mostra o número de descobertas para cada pipeline de imagens em sua conta. O Image Builder exibe contagens de descobertas de severidade média e mais altas, além de um total de todas as descobertas. Todos os dados na lista estão vinculados, da seguinte forma:
  - A coluna Nome do pipeline de imagem está vinculada à página de detalhes do pipeline de imagem especificado.
  - Os links da coluna de nível de severidade abrem a exibição Todas as descobertas de segurança, filtrada pelo nome do pipeline de imagem associado e pelo nível de gravidade.

  Você também pode usar critérios de pesquisa para refinar os resultados.
- Por imagem: essa visualização mostra o número de descobertas para cada criação de imagens em sua conta. O Image Builder exibe contagens de descobertas de severidade média e mais altas,

além de um total de todas as descobertas. Todos os dados na lista estão vinculados, da seguinte forma:

- A coluna Nome da imagem está vinculada à página de detalhes da imagem para a criação de imagem especificada. Para obter mais informações, consulte Visualizar os detalhes do recurso de imagem.
- Os links da coluna de nível de severidade abrem a visualização Todas as descobertas de segurança, filtrada pelo nome da compilação de imagem associado e pelo nível de gravidade.

Você também pode usar critérios de pesquisa para refinar os resultados.

O Image Builder mostra os seguintes detalhes na seção Lista de descobertas da visualização padrão Todas as descobertas de segurança.

Gravidade

O nível de gravidade da descoberta de CVE. Os valores são os seguintes:

- Não triado
- Informativo
- Baixo
- Médio
- Alto
- Crítico

ID da descoberta

O identificador exclusivo da descoberta do CVE que o Amazon Inspector detectou para sua imagem ao verificar a instância de compilação. O ID está vinculado à página Descobertas de segurança > Por vulnerabilidade.

ARN da imagem

O nome do recurso da Amazon (ARN) da imagem com a descoberta especificada na coluna ID da descoberta.

Pipeline

O pipeline que criou a imagem especificada na coluna ARN da imagem.

Descrição

Uma breve descrição da descoberta.

Pontuação do Inspector

A pontuação que o Amazon Inspector atribuiu para a descoberta do CVE.

Correção

Links para detalhes sobre o curso de ação recomendado para remediar a descoberta.

Data de publicação

A data e hora quando essa vulnerabilidade foi adicionada pela primeira vez ao banco de dados do fornecedor.

Limpar recursos do Image Builder

Para evitar cobranças inesperadas, certifique-se de limpar os recursos e pipelines que você criou com base nos exemplos deste guia. Para obter mais informações sobre exclusão de recursos no Image Builder, consulte Excluir recursos desatualizados ou não utilizados do Image Builder.

Gerenciar políticas de ciclo de vida de imagens do Image Builder

Ao criar imagens personalizadas, é importante que você tenha um plano para retirar essas imagens antes que elas se tornem obsoletas. Os pipelines do Image Builder podem aplicar atualizações e patches de segurança automaticamente. No entanto, cada compilação criará uma nova versão da imagem e de todos os recursos associados que ela distribui. As versões anteriores permanecerão na sua conta até que você as exclua manualmente ou crie um script para realizar a tarefa.

Com as políticas de gerenciamento do ciclo de vida do Image Builder, você pode automatizar o processo de descontinuação, desabilitação e exclusão de imagens desatualizadas e seus recursos associados. Os recursos associados podem incluir imagens de saída que você distribuiu para outras Contas da AWS organizações e unidades organizacionais (OUs) Regiões da AWS. Você define as regras de como e quando realizar cada etapa do processo do ciclo de vida e quais etapas incluir em sua política.

Benefícios do gerenciamento automatizado do ciclo de vida

Os benefícios gerais do gerenciamento automatizado do ciclo de vida incluem:

- Simplifica o gerenciamento do ciclo de vida de suas imagens personalizadas com uma forma automatizada de retirar imagens e recursos associados.
- Ajuda a evitar riscos de conformidade decorrentes do uso de imagens desatualizadas para iniciar novas instâncias.
- Mantém os inventários de imagens atualizados ao remover imagens desatualizadas.
- Pode reduzir os custos de armazenamento e transferência de dados removendo opcionalmente os recursos associados às imagens que são excluídas.

Concretize economia de custos

Não há custo para usar o EC2 Image Builder para criar imagens personalizadas de AMI ou contêiner. No entanto, o preço padrão se aplica a outros serviços que são usados no processo. Ao remover imagens não utilizadas ou desatualizadas e seus recursos associados Conta da AWS, você pode economizar tempo e custos das seguintes maneiras:

- Reduzir o tempo necessário para corrigir imagens existentes quando você também não estiver corrigindo imagens não utilizadas ou desatualizadas.

- Para os recursos de imagem da AMI que você exclui, você também pode optar por remover os snapshots distribuídos AMIs e seus associados. Essa abordagem pode economizar no custo de armazenamento de snapshots.
- Para recursos de imagem de contêiner que você exclui, é possível optar por excluir os recursos subjacentes. Essa abordagem pode economizar os custos de armazenamento e as taxas de transferência de dados do Amazon ECR para suas imagens do Docker armazenadas em repositórios do ECR.

> **Note**
>
> O Image Builder não pode avaliar o possível impacto de todas as possíveis dependências downstream, como grupos do Auto Scaling ou modelos de inicialização. Você deve considerar as dependências downstream de suas imagens ao configurar ações de política.

Conteúdo

- Pré-requisitos do gerenciamento do ciclo de vida de imagens do Image Builder
- Listar políticas de gerenciamento do ciclo de vida para recursos de imagem do Image Builder
- Visualizar detalhes da política de ciclo de vida
- Criar políticas de ciclo de vida
- Como as regras de gerenciamento do ciclo de vida funcionam para os recursos de imagem do Image Builder

Pré-requisitos do gerenciamento do ciclo de vida de imagens do Image Builder

Antes de definir as políticas e regras de gerenciamento do ciclo de vida do EC2 Image Builder para seus recursos de imagem, você deve atender aos seguintes pré-requisitos.

- Crie um perfil do IAM que conceda permissão para o Image Builder executar políticas de ciclo de vida. Para criar o perfil, consulte Criar um perfil do IAM para o gerenciamento do ciclo de vida do Image Builder.
- Crie um perfil do IAM na conta de destino para recursos associados que foram distribuídos entre contas. O perfil concederá permissão para o Image Builder realizar ações de ciclo de vida na conta

de destino para os recursos associados. Para criar o perfil, consulte Criar um perfil do IAM para o gerenciamento do ciclo de vida entre contas do Image Builder.

> **Note**
>
> Esse pré-requisito não se aplicará se você tiver concedido permissões de execução para uma AMI de saída. Com as permissões de execução, a conta com a qual você compartilhou o recurso será proprietária das instâncias que forem executadas diretamente da AMI compartilhada, mas todos os recursos da AMI permanecerão na sua conta.

- Para imagens de contêiner, você deverá adicionar a seguinte tag aos seus repositórios do ECR para permitir que o Image Builder tenha acesso para executar ações de ciclo de vida nas imagens de contêiner armazenadas no repositório: `LifecycleExecutionAccess: EC2 Image Builder`.

Criar um perfil do IAM para o gerenciamento do ciclo de vida do Image Builder

Para conceder permissão ao Image Builder para executar políticas de ciclo de vida, primeiro você deve criar o perfil do IAM que ele usa para realizar ações de ciclo de vida. Siga estas etapas para criar o perfil de serviço que concede permissão.

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. Escolha Funções no painel de navegação.
3. Selecione Criar perfil. Isso abre a primeira etapa do processo, Selecionar uma entidade confiável para criar seu perfil.
4. Selecione a opção Política de confiança personalizada em Tipo de entidade confiável.
5. Copie a política de confiança JSON a seguir e cole na área do texto de Política de confiança personalizada, substituindo o texto de exemplo. Essa política de confiança permite que o Image Builder assuma o perfil que você cria para executar ações do ciclo de vida.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
```

```
        {
            "Action": [
                "sts:AssumeRole"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "imagebuilder.amazonaws.com"
                ]
            }
        }
    ]
}
```

6. Selecione a seguinte política gerenciada na lista: EC2ImageBuilderLifecycleExecutionPolicy e escolha Avançar. Isso exibirá a página Nomear, revisar e criar.

> **Tip**
>
> Filtre por `image` para otimizar os resultados.

7. Insira um nome em Nome da função.
8. Após revisar suas configurações, escolha Criar perfil.

Criar um perfil do IAM para o gerenciamento do ciclo de vida entre contas do Image Builder

Para permitir que o Image Builder realize ações de ciclo de vida em contas de destino para recursos associados, primeiro você deve criar o perfil do IAM que ele usará para realizar ações de ciclo de vida nessas contas. É necessário criar o perfil na conta de destino.

Siga estas etapas para criar o perfil de serviço que concede permissão na conta de destino.

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. Escolha Funções no painel de navegação.
3. Selecione Criar perfil. Isso abre a primeira etapa do processo, Selecionar uma entidade confiável para criar seu perfil.
4. Selecione a opção Política de confiança personalizada em Tipo de entidade confiável.

5. Copie a política de confiança JSON a seguir e cole na área do texto de Política de confiança personalizada, substituindo o texto de exemplo. Essa política de confiança permite que o Image Builder assuma o perfil que você cria para executar ações do ciclo de vida.

> **Note**
>
> Quando o Image Builder usa esse perfil na conta de destino para atuar nos recursos associados que foram distribuídos entre contas, ele estará agindo em nome do proprietário da conta de destino. O Conta da AWS que você configura como `aws:SourceAccount` na política de confiança é a conta em que o Image Builder distribuiu esses recursos.

```
{
    "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. Selecione a seguinte política gerenciada na lista: EC2ImageBuilderLifecycleExecutionPolicy e escolha Avançar. Isso exibirá a página Nomear, revisar e criar.

> **Tip**
>
> Filtre por `image` para otimizar os resultados.

7. Insira `Ec2ImageBuilderCrossAccountLifecycleAccess` como o Nome do perfil.

> **Important**
>
> O nome dessa função deve ser `Ec2ImageBuilderCrossAccountLifecycleAccess`.

8. Após revisar suas configurações, escolha Criar perfil.

Listar políticas de gerenciamento do ciclo de vida para recursos de imagem do Image Builder

Você pode obter uma lista das políticas de gerenciamento do ciclo de vida da imagem que inclui as principais colunas de detalhes na página de listagem de políticas do ciclo de vida no Console de gerenciamento da AWS, ou com comandos ou ações na API Image Builder,, ou. SDKs AWS CLI

Você pode aplicar um dos métodos a seguir para listar recursos de política de ciclo de vida de imagem do Image Builder em sua Conta da AWS. Para a ação da API, consulte ListLifecyclePoliciesna Referência da API do EC2 Image Builder. Para ver a solicitação de SDK associada, consulte o link Consulte também na mesma página.

Console de gerenciamento da AWS

Os detalhes a seguir são exibidos no console para suas políticas existentes. É possível selecionar qualquer coluna para alterar a ordem de classificação dos resultados. A lista de políticas é classificada inicialmente por Nome da política. O nome da coluna para a ordem de classificação atual está em negrito.

Se você tiver mais de uma página de resultados, as setas de paginação no canto superior direito do painel ficarão ativas. É possível filtrar os resultados por nome da política, status da política, tipo de imagem de saída e ARN do recurso de imagem com a barra de pesquisa.

- Nome da política: o nome da política.

- Status da política: se a política está ativa ou inativa.
- Tipo: o tipo de imagem de saída que o Image Builder distribui quando você cria uma nova versão de imagem (AMI ou imagem de contêiner).
- Data da última execução: a última vez em que a política de ciclo de vida foi executada.
- Data de criação: o carimbo de data e hora da criação da política de ciclo de vida.
- ARN: o nome do recurso da Amazon (ARN) do recurso de política de ciclo de vida.

Para listar as políticas de ciclo de vida no Console de gerenciamento da AWS, siga estas etapas:

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Selecione Políticas de ciclo de vida no painel de navegação. Isso exibirá uma lista das políticas de ciclo de vida da imagem em sua conta.

Ações disponíveis

Você também pode realizar as seguintes ações para sua política de ciclo de vida na página de lista Políticas de ciclo de vida.

Para criar uma nova política de ciclo de vida de imagem, escolha Criar política de ciclo de vida. Para obter mais informações sobre como criar uma política, consulte Criar políticas de ciclo de vida.

Para todas as ações a seguir, primeiro você deve selecionar a política. Para selecionar uma política, é possível marcar a caixa de seleção ao lado do Nome da política.

- Para ativar ou desativar a política, selecione Desabilitar política ou Habilitar política no menu Ações.
- Para alterar a política, selecione Editar política no menu Ações.
- Para excluir uma política, selecione Excluir política no menu Ações.
- Para criar uma nova política que use sua política selecionada para configurações de linha de base, selecione Clonar política no menu Ações.

AWS CLI

O exemplo de comando a seguir mostra como usar as políticas AWS CLI de ciclo de vida de imagens para uma imagem específica. Região da AWS Para obter mais informações sobre

os parâmetros e as opções que você pode usar com esse comando, consulte o list-lifecycle-policiescomando na Referência de AWS CLI Comandos.

Exemplo:

```
aws imagebuilder list-lifecycle-policies \
--region us-west-1
```

Saída:

```
{
    "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**
>
> Para usar seu padrão Região da AWS, execute esse comando sem o `--region` parâmetro.

Visualizar detalhes da política de ciclo de vida

A página de detalhes de política de ciclo de vida no console do Image Builder inclui uma seção de resumo, com informações adicionais agrupadas em guias. O cabeçalho de página é o nome da política.

Na página de detalhes da política de ciclo de vida no console do Image Builder, você pode ver detalhes de uma política específica de ciclo de vida. Você também pode usar comandos ou ações com a API Image Builder ou AWS CLI obter detalhes da política. SDKs

Conteúdo

- Visualizar os detalhes da política de ciclo de vida no console do Image Builder

Visualizar os detalhes da política de ciclo de vida no console do Image Builder

A página de detalhes da imagem no console do Image Builder inclui uma seção de resumo, com informações adicionais agrupadas em guias. O título da página é o nome e a versão de compilação da fórmula que criou a imagem.

Seções e guias de detalhes do console

- Seção de resumo
- Guia Regras
- Guia Escopo
- RunLog aba

Seção de resumo

A seção de resumo abrange a largura da página e inclui os detalhes a seguir. Esses detalhes são sempre exibidos.

Status da política

Se a política está ativa ou inativa.

Tipo

O tipo de imagem de saída que o Image Builder distribui quando você cria uma nova versão de imagem (AMI ou imagem de contêiner).

Data da criação

O carimbo de data e hora da criação da política de ciclo de vida.

Data de modificação

A última atualização da política de ciclo de vida.

Data da última execução

A última execução da política de ciclo de vida.

Perfil do IAM

O perfil do IAM que o Image Builder usa para realizar ações de ciclo de vida.

ARN

O nome do recurso da Amazon (ARN) do recurso de política de ciclo de vida.

Descrição

A descrição da política de ciclo de vida, se inserida.

Guia Regras

A guia Regras exibe as regras de ciclo de vida que você configurou para a política que está visualizando. A guia inclui os seguintes detalhes:

- Nome: o nome da regra. Esses nomes são estáticos, com base nas ações de política que você pode configurar.

  - `Deprecation rule`
  - `Disable rule`
  - `Deletion rule`
- Regra: uma breve descrição da ação configurada para a regra.
- Condições da regra: lista a configuração para processamento de recursos associados, exceções à regra e configurações de retenção, se for o caso.

Para obter mais informações sobre a configuração de regra, consulte Como as regras do ciclo de vida funcionam.

Guia Escopo

A guia Escopo exibe os critérios de seleção de recursos configurados para a política que você está visualizando. A guia inclui os seguintes detalhes:

- Filtro: ***`type of filter`*** — O tipo de filtro usado para definir o escopo. O tipo de filtro pode ser um dos seguintes:
  - `recipes`: as fórmulas que foram usadas para criar as imagens às quais a política de ciclo de vida se aplica.
  - `tags`: um conjunto de tags que o Image Builder usa para selecionar recursos de imagem aos quais a política de ciclo de vida se aplica.
- Uma barra de pesquisa: você pode filtrar a lista por Nome para otimizar os resultados exibidos na guia.
- Nome: cada linha contém um nome ou tag que você configurou para os critérios do filtro.
- Versão: se você tiver configurado um filtro de fórmula, o Image Builder exibirá a versão da fórmula.

RunLog aba

Sempre que você executar a política para seus recursos configurados, o Image Builder salvará os detalhes do runtime. Cada linha na tabela representa uma única instância de runtime. A guia inclui os seguintes detalhes:

- ID de execução: identifica a instância de runtime da política de ciclo de vida.
- Status de execução: o status de runtime que informa se a ação de política está em execução, foi executada com êxito, falhou ou foi cancelada.

- Recurso afetado: indica se a instância de runtime identificou algum recurso de imagem para ações do ciclo de vida.
- Data inicial: o carimbo de data e hora em que a instância de runtime foi iniciada.
- Data final: o carimbo de data e hora em que a instância de runtime terminou.

Criar políticas de ciclo de vida

Quando você cria uma nova política de ciclo de vida do EC2 Image Builder, a configuração depende do tipo de imagem para o qual a política se destina. A ação da API para criar uma política de ciclo de vida para recursos de imagem da AMI e recursos de imagem de contêiner é a mesma () CreateLifecyclePolicy. No entanto, a configuração dos recursos de imagem e dos recursos associados é diferente. Esta seção mostra como criar políticas de gerenciamento do ciclo de vida para ambos.

> **Note**
>
> Antes de criar uma política de ciclo de vida, garanta que atendeu a todos os Pré-requisitos.

Criar políticas de gerenciamento do ciclo de vida para recursos de imagem de AMI do Image Builder

Você pode usar um dos métodos a seguir para criar uma política de ciclo de vida de imagem da AMI com o Console de gerenciamento da AWS ou. AWS CLI Você também pode usar a ação CreateLifecyclePolicyda API. Para a solicitação de SDK associada, você pode consultar o link Consulte também desse comando na Referência da API EC2 Image Builder.

Console de gerenciamento da AWS

Para criar uma política de ciclo de vida para recursos de imagem da AMI no Console de gerenciamento da AWS, siga estas etapas:

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Escolha Políticas de ciclo de vida no painel de navegação.
3. Escolha Criar política de ciclo de vida.
4. Defina as configurações de política descritas nos procedimentos a seguir.

5. Para criar a política de ciclo de vida após definir as configurações, escolha Criar política.

Defina as configurações Gerais para sua política.

1. Selecione a opção AMI em Tipo de política.
2. Insira o Nome da política.
3. Opcionalmente, insira uma Descrição para sua política de ciclo de vida.
4. A opção Ativar estará ativada por padrão. A configuração padrão ativa a política de ciclo de vida e a adiciona ao cronograma imediatamente. Para criar uma política que esteja inicialmente desativada, você pode desativar a opção Ativar.
5. Selecione o Perfil do IAM que você criou para as permissões da política de ciclo de vida. Se ainda não tiver criado esse perfil, consulte Pré-requisitos para obter mais informações.

Configure o Escopo da regra para sua política.

Essa seção configura a seleção de recursos para sua política de ciclo de vida com base no tipo de filtro que você usa.

1. Tipo de filtro: fórmulas: para aplicar regras de ciclo de vida aos recursos de imagem com base na fórmula que os criou, selecione até 50 versões da fórmula para a política.
2. Tipo de filtro: tags: para aplicar regras de ciclo de vida a recursos de imagem com base em tags de recursos, insira uma lista de até 50 pares de valor e chave correspondentes à política.

Ative uma ou mais das seguintes regras de ciclo de vida para aplicar aos recursos que a política de ciclo de vida seleciona. Se um recurso corresponder a mais de uma regra de ciclo de vida quando a política for executada, o Image Builder executará ações de regra na seguinte ordem: 1) Descontinuar, 2) Desabilitar, 3) Excluir.

Descontinuar regra

Define o status do recurso de imagem do Image Builder como `Deprecated`. Os pipelines do Image Builder ainda serão executados para imagens obsoletas. Opcionalmente, você pode definir o tempo de suspensão de uso para associados AMIs sem afetar sua capacidade de iniciar novas instâncias.

- Contagem de unidades: especifique o valor inteiro para o período de tempo a decorrer após a criação de um recurso de imagem e antes que esse recurso seja marcado como `Deprecated`.
- Unidade: selecione o intervalo de tempo a ser usado. O valor pode ser `Days`, `Weeks`, `Months` ou `Years`.
- Descontinuar AMIs — Marque a caixa de seleção para marcar a Amazon associada EC2 AMIs com uma data de suspensão de uso. Eles AMIs permanecem disponíveis e você ainda pode iniciar novas instâncias a partir deles.

Desabilitar regra

Define o status do recurso de imagem do Image Builder como `Disabled`. Isso impede que os pipelines do Image Builder sejam executados para essa imagem. Opcionalmente, você pode desativar a AMI associada para evitar a execução de novas instâncias.

- Contagem de unidades: especifique o valor inteiro para o período de tempo a decorrer após a criação de um recurso de imagem e antes que esse recurso seja marcado como `Disabled`.
- Unidade: selecione o intervalo de tempo a ser usado. O valor pode ser `Days`, `Weeks`, `Months` ou `Years`.
- Desativar AMIs — Selecione a caixa de seleção para desativar a Amazon EC2 AMIs associada. Você não pode mais usar AMIs ou executar novas instâncias a partir delas.

Excluir regra

Exclui os recursos de imagem por idade ou por contagem. Você define o limite de acordo com suas necessidades. Quando um recurso de imagem do Image Builder ultrapassar o limite, ele será removido. Opcionalmente, você pode cancelar o registro associado AMIs ou excluir os snapshots desses. AMIs Você também poderá especificar tags para os recursos que deseja reter além do limite.

Quando você configura a Regra excluir por idade, o Image Builder exclui o recurso de imagem após o período configurado. Por exemplo, excluir recursos de imagem após 6 meses. Quando você configura por contagem, o Image Builder retém o número mais recente de imagens que você especifica, ou o mais próximo possível desse número, e exclui as versões anteriores.

- Por idade
  - Contagem de unidades: especifique o valor inteiro para o período de tempo a decorrer após a criação de um recurso de imagem e antes que esse recurso seja excluído.

- Unidade: selecione o intervalo de tempo a ser usado. O valor pode ser `Days`, `Weeks`, `Months` ou `Years`.
- Reter pelo menos uma imagem por fórmula: marque a caixa de seleção para manter o recurso de imagem mais recente disponível para cada versão da fórmula afetada por essa regra.

Por contagem

- Contagem de imagens: especifique o valor inteiro do número de recursos de imagem recentes a serem mantidos em cada versão da fórmula.

- Cancelar registro AMIs — Marque a caixa de seleção para cancelar o registro da Amazon associada. EC2 AMIs Você não pode mais usar AMIs ou executar novas instâncias a partir delas.
- Guarde imagens e instantâneos com tags associadas — Marque a caixa de seleção para inserir uma lista de tags para os recursos de imagem que você deseja manter. AMIs As tags se aplicam aos recursos de imagem e à Amazon EC2 AMIs. É possível adicionar até 50 pares de chave e valor.

Etiquetas (opcional)

Adicione etiquetas à sua política de ciclo de vida.

AWS CLI

Para criar uma nova política de ciclo de vida do Image Builder, você pode usar o comando create-lifecycle-policy na AWS CLI.

Criar políticas de gerenciamento do ciclo de vida para recursos de imagem de contêiner do Image Builder

Você pode usar um dos métodos a seguir para criar uma política de ciclo de vida de imagem de contêiner com o Console de gerenciamento da AWS ou. AWS CLI Você também pode usar a ação CreateLifecyclePolicyda API. Para a solicitação de SDK associada, você pode consultar o link Consulte também desse comando na Referência da API EC2 Image Builder.

Console de gerenciamento da AWS

Para criar uma política de ciclo de vida para recursos de imagem de contêiner no Console de gerenciamento da AWS, siga estas etapas:

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Escolha Políticas de ciclo de vida no painel de navegação.
3. Escolha Criar política de ciclo de vida.
4. Defina as configurações de política descritas nos procedimentos a seguir.
5. Para criar a política de ciclo de vida após definir as configurações, escolha Criar política.

Configuração de política: configurações gerais

Defina as configurações Gerais para sua política.

1. Selecione a opção AMI em Tipo de política.
2. Insira o Nome da política.
3. Opcionalmente, insira uma Descrição para sua política de ciclo de vida.
4. A opção Ativar estará ativada por padrão. A configuração padrão ativa a política de ciclo de vida e a adiciona ao cronograma imediatamente. Para criar uma política que esteja inicialmente desativada, você pode desativar a opção Ativar.
5. Selecione o Perfil do IAM que você criou para as permissões da política de ciclo de vida. Se ainda não tiver criado esse perfil, consulte Pré-requisitos para obter mais informações.

Configure o Escopo da regra para sua política.

Essa seção configura a seleção de recursos para sua política de ciclo de vida com base no tipo de filtro que você usa.

1. Tipo de filtro: fórmulas: para aplicar regras de ciclo de vida aos recursos de imagem com base na fórmula que os criou, selecione até 50 versões da fórmula para a política.
2. Tipo de filtro: tags: para aplicar regras de ciclo de vida a recursos de imagem com base em tags de recursos, insira uma lista de até 50 pares de valor e chave correspondentes à política.

Excluir regra

Para imagens de contêiner, essa regra exclui o recurso de imagem de contêiner do Image Builder. Opcionalmente, você pode remover imagens do Docker que foram distribuídas para repositórios do ECR a fim de evitar que elas sejam usadas para executar novos contêineres.

Quando você configura a Regra excluir por idade, o Image Builder exclui o recurso de imagem após o período configurado. Por exemplo, excluir recursos de imagem após 6 meses. Quando você configura por contagem, o Image Builder retém o número mais recente de imagens que você especifica, ou o mais próximo possível desse número, e exclui as versões anteriores.

- Por idade
  - Contagem de unidades: especifique o valor inteiro para o período de tempo a decorrer após a criação de um recurso de imagem e antes que esse recurso seja excluído.
  - Unidade: selecione o intervalo de tempo a ser usado. O valor pode ser `Days`, `Weeks`, `Months` ou `Years`.
  - Reter pelo menos uma imagem: marque a caixa de seleção para manter apenas o recurso de imagem mais recente disponível para cada versão da fórmula afetada por essa regra.

  Por contagem
  - Contagem de imagens: especifique o valor inteiro do número de recursos de imagem recentes a serem mantidos em cada versão da fórmula.
- Excluir imagens de contêiner do ECR: marque a caixa de seleção para excluir imagens de contêiner associadas armazenadas em um repositório do ECR. Você não pode mais usar a imagem de contêiner como base para criar novas imagens ou executar novos contêineres.
- Reter imagens com tags associadas: marque a caixa de seleção para inserir uma lista de tags para os recursos de imagem que você deseja manter.

Etiquetas (opcional)

Adicione etiquetas à sua política de ciclo de vida.

AWS CLI

Para criar uma nova política de ciclo de vida do Image Builder, você pode usar o comando create-lifecycle-policy na AWS CLI.

Como as regras de gerenciamento do ciclo de vida funcionam para os recursos de imagem do Image Builder

As políticas de ciclo de vida de imagens usam as regras de ciclo de vida que você define para implementar sua estratégia geral de gerenciamento de recursos. As regras que você define ajudam a garantir a atualização das imagens disponíveis e a minimizar os custos da infraestrutura subjacente, como armazenamento de instantâneos para saída AMIs ou armazenamento de repositórios ECR e taxas de transferência de dados para imagens de contêineres.

É possível configurar os seguintes tipos de regras para suas políticas.

Descontinuar regra

Define o status do recurso de imagem do Image Builder como `Deprecated`. Os pipelines do Image Builder ainda serão executados para imagens obsoletas. Opcionalmente, você pode definir o tempo de suspensão de uso para associados AMIs sem afetar sua capacidade de iniciar novas instâncias.

Quando uma AMI é descontinuada, ela é ignorada pelas pesquisas gerais. Por exemplo, se você executar o EC2 describe-images comando da Amazon no AWS CLI, ele não retornará obsoleto AMIs no conjunto de resultados. No entanto, você ainda pode encontrar a versão obsoleta com AMIs sua AMI ID.

Essa regra não está disponível para imagens de contêiner.

Desabilitar regra

Define o status do recurso de imagem do Image Builder como `Disabled`. Isso impede que os pipelines do Image Builder sejam executados para essa imagem. Opcionalmente, você pode desativar a AMI associada para evitar a execução de novas instâncias.

Quando uma AMI é desabilitada, ela passa a ser privada e não pode ser usada para executar novas instâncias. Se você tiver compartilhado a AMI com qualquer conta, organização ou unidade organizacional, ela perderá o acesso à sua AMI quando ela ficar privada.

Essa regra não está disponível para imagens de contêiner.

Excluir regra

Exclui os recursos de imagem por idade ou por contagem. Você define o limite de acordo com suas necessidades. Quando um recurso de imagem do Image Builder ultrapassar o limite, ele será removido. Opcionalmente, você pode cancelar o registro associado AMIs ou excluir os

snapshots desses. AMIs Você também poderá especificar tags para os recursos que deseja reter além do limite.

Para imagens de contêiner, essa regra exclui o recurso de imagem de contêiner do Image Builder. Opcionalmente, você pode remover imagens de contêiner que foram distribuídas para repositórios do ECR a fim de evitar que elas sejam usadas para executar novos contêineres.

Conteúdo

- Regras de exclusão do ciclo de vida da AMI
- Visualizar os detalhes das regras de gerenciamento do ciclo de vida de uma política

Regras de exclusão do ciclo de vida da AMI

As regras de exclusão a seguir definem exceções às regras do ciclo de vida do. AMIs AMIs que atendam aos critérios especificados pelas regras de exclusão são excluídos das ações do ciclo de vida. As regras de exclusão não estão disponíveis no Console de gerenciamento da AWS.

Os termos a seguir usam a notação de API do tipo de dados `LifecyclePolicyDetailExclusionRules`.

Regras de exclusão

amis

Contém as configurações de `LifecyclePolicyDetailExclusionRulesAmis` mostradas na lista a seguir.

tagMap

Você pode fornecer uma lista com até 50 tags que ignoram as ações do ciclo de vida de qualquer tipo de recurso.

Os termos a seguir usam a notação de API do tipo de dados `LifecyclePolicyDetailExclusionRulesAmis`.

Regras de exclusão de AMI

isPublic

Configura se os públicos AMIs são excluídos da ação do ciclo de vida.

lastLaunched

Especifica os detalhes da configuração do Image Builder para excluir os recursos mais recentes das ações do ciclo de vida.

regions

Configura Regiões da AWS que são excluídas da ação do ciclo de vida.

sharedAccounts

Especifica Contas da AWS quais recursos são excluídos da ação do ciclo de vida.

tagMap

Lista as tags que devem ser excluídas das ações do ciclo de vida dos AMIs que as possuem.

Visualizar os detalhes das regras de gerenciamento do ciclo de vida de uma política

As regras são definidas nas políticas de gerenciamento do ciclo de vida que você cria para seus recursos de imagem do Image Builder. No console, a página de detalhes da política de ciclo de vida tem uma Guia Regras que mostra os detalhes das regras que você configurou para a política.

Para obter detalhes da política no AWS CLI, você pode executar o get-lifecycle-policycomando. Os detalhes da política na resposta contêm uma lista das ações (regras) que você definiu para a política, que incluem todas as suas configurações definidas.

Configurar imagens personalizadas com o Image Builder

Os recursos de configuração são os componentes básicos dos pipelines de imagens, bem como as imagens que esses pipelines produzem. Este capítulo aborda a criação, manutenção e compartilhamento de recursos do Image Builder, incluindo componentes, fórmulas e imagens, além de configurações de infraestrutura e configurações de distribuição.

> **Note**
>
> Para ajudar você a gerenciar os recursos do Image Builder, é possível atribuir seus próprios metadados a cada recurso na forma de tags. Você pode usar etiquetas para categorizar seus recursos da AWS de diferentes formas, como por finalidade, proprietário ou ambiente. Isso é útil quando você tem muitos recursos do mesmo tipo. Identifique rapidamente um recurso específico com base nas tags atribuídas a ele.
> Para obter mais informações sobre como marcar seus recursos usando os comandos do Image Builder no AWS CLI, consulte a Marcar recursos seção deste guia.

Conteúdo

- Gerenciar fórmulas no Image Builder
- Gerenciar a configuração da infraestrutura do Image Builder
- Gerenciar as configurações de distribuição do Image Builder

Gerenciar fórmulas no Image Builder

Uma receita do EC2 Image Builder define a imagem base a ser usada como ponto de partida para criar uma nova imagem, junto com o conjunto de componentes que você adiciona para personalizar sua imagem e verificar se tudo funciona conforme o esperado. O Image Builder fornece opções automáticas de versão para cada componente. Por padrão, você pode aplicar até vinte componentes a uma fórmula. Isso inclui componentes de compilação e teste.

Depois de criar uma fórmula, não é possível modificá-la ou substituí-la. Para atualizar os componentes depois de criar uma fórmula, você deve criar uma nova fórmula ou versão da fórmula. Você sempre pode aplicar tags às fórmulas existentes. Para obter mais informações sobre como marcar seus recursos usando os comandos do Image Builder no AWS CLI, consulte a Marcar recursos seção deste guia.

> **Tip**
>
> Você pode usar componentes gerenciados pela Amazon em suas receitas ou desenvolver seus próprios componentes personalizados. Para obter mais informações, consulte Desenvolver componentes personalizados para imagens do Image Builder. Para receitas de imagens que criam saída AMIs, você também pode usar produtos e componentes de AWS Marketplace imagem. Para obter mais informações sobre integração com AWS Marketplace produtos, consulteAWS Marketplace integração no Image Builder.

Esta seção aborda como listar, visualizar e criar fórmulas.

Conteúdo

- Liste e visualize detalhes da fórmula da imagem
- Liste e visualize detalhes da fórmula de contêiner
- Criar uma nova versão de uma fórmula de imagem
- Criar uma nova versão de fórmula de contêiner
- Limpar recursos

Liste e visualize detalhes da fórmula da imagem

Esta seção descreve as várias maneiras pelas quais você pode encontrar informações e visualizar detalhes de suas receitas de imagens do EC2 Image Builder.

Detalhes da fórmula de imagem

- Listar fórmulas de imagem com o console
- Listar receitas de imagens do AWS CLI
- Visualizar detalhes da fórmula de imagem com o console
- Obtenha detalhes da receita da imagem no AWS CLI
- Obtenha detalhes da política de receitas de imagens no AWS CLI

Listar fórmulas de imagem com o console

Para ver uma lista das fórmulas de imagem que foram criadas em sua conta no console do Image Builder, siga estas etapas:

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Escolha Fórmulas de imagem do painel de navegação. Isso mostra uma lista das fórmulas de imagem criadas em sua conta.
3. Para ver detalhes ou criar uma nova versão da fórmula, escolha o link Nome da fórmula. Isso abre a visualização de detalhes da fórmula.

> **Note**
>
> Você também pode selecionar a caixa ao lado de Nome da fórmula, depois escolher Visualizar detalhes.

Listar receitas de imagens do AWS CLI

O exemplo a seguir mostra como listar todas as fórmulas de imagem usando a AWS CLI.

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

Visualizar detalhes da fórmula de imagem com o console

Para ver detalhes de uma fórmula de imagem específica usando o console do Image Builder, selecione a fórmula de imagem a ser revisada e use as etapas descritas em Listar fórmulas de imagem com o console.

Na página de detalhes da fórmula, você pode:

- Excluir a fórmula. Para obter mais informações sobre exclusão de recursos no Image Builder, consulte Excluir recursos desatualizados ou não utilizados do Image Builder.
- Criar uma nova versão.
- Criar um pipeline a partir da fórmula. Depois de escolher Criar pipeline a partir desta fórmula, você será direcionado para o assistente de pipeline. Para obter mais informações sobre como criar um pipeline do Image Builder usando o assistente de pipeline, consulte Tutorial: criar um pipeline de imagens com a AMI de saída por meio do assistente do console do Image Builder

> **Note**
>
> Quando você cria um pipeline a partir de uma fórmula existente, a opção de criar uma nova fórmula não está disponível.

Obtenha detalhes da receita da imagem no AWS CLI

O exemplo a seguir mostra como usar um comando CLI imagebuilder para obter os detalhes de uma fórmula de imagem especificando seu nome do recurso da 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
```

Obtenha detalhes da política de receitas de imagens no AWS CLI

O exemplo a seguir mostra como usar um comando CLI imagebuilder para obter os detalhes de uma política de fórmula especificando seu ARN.

```
aws imagebuilder get-image-recipe-policy --image-recipe-arn arn:aws:imagebuilder:us-
west-2:123456789012:image-recipe/my-example-recipe/2020.12.03
```

Liste e visualize detalhes da fórmula de contêiner

Esta seção descreve as maneiras pelas quais você pode encontrar informações e visualizar detalhes das receitas de contêineres do EC2 Image Builder.

Detalhes da fórmula de contêiner

- Liste as fórmulas de contêiner no console
- Listar as fórmulas de contêiner com a AWS CLI
- Visualizar detalhes da fórmula de contêiner no console
- Obtenha detalhes da fórmula de contêiner com a AWS CLI
- Obtenha detalhes da política de receitas de contêineres com o AWS CLI

Liste as fórmulas de contêiner no console

Para ver uma lista das fórmulas de contêiner que foram criadas em sua conta no console do Image Builder, siga estas etapas:

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Escolha Fórmulas de contêiner do painel de navegação. Isso mostra uma lista das fórmulas de contêiner criadas em sua conta.
3. Para ver detalhes ou criar uma nova versão da fórmula, escolha o link Nome da fórmula. Isso abre a visualização de detalhes da fórmula.

> **Note**
>
> (Você também pode selecionar a caixa ao lado de Nome da fórmula, e depois escolher Exibir detalhes.

Listar as fórmulas de contêiner com a AWS CLI

O exemplo a seguir mostra como listar todas as fórmulas de contêiner, usando a AWS CLI.

```
aws imagebuilder list-container-recipes
```

Visualizar detalhes da fórmula de contêiner no console

Para visualizar detalhes de uma fórmula de contêiner específica com o console do Image Builder, selecione a fórmula de contêiner a ser visualizada e use as etapas descritas em Liste as fórmulas de contêiner no console.

Na página de detalhes da fórmula, você pode fazer o seguinte:

- Excluir a fórmula. Para obter mais informações sobre como excluir recursos no Image Builder, consulte Excluir recursos desatualizados ou não utilizados do Image Builder.
- Criar uma nova versão.
- Criar um pipeline a partir da fórmula. Depois de escolher Criar pipeline a partir desta fórmula, você será direcionado para o assistente de pipeline. Para obter mais informações sobre como criar um pipeline do Image Builder usando o assistente de pipeline, consulte Tutorial: criar um pipeline de imagens com a AMI de saída por meio do assistente do console do Image Builder

> **Note**
>
> Quando você cria um pipeline a partir de uma fórmula existente, a opção de criar uma nova fórmula não está disponível.

Obtenha detalhes da fórmula de contêiner com a AWS CLI

O exemplo a seguir mostra como usar um comando CLI imagebuilder para obter os detalhes de uma fórmula de contêiner especificando seu ARN.

```
aws imagebuilder get-container-recipe --container-recipe-arn arn:aws:imagebuilder:us-
west-2:123456789012:container-recipe/my-example-recipe/2020.12.03
```

Obtenha detalhes da política de receitas de contêineres com o AWS CLI

O exemplo a seguir mostra como usar um comando CLI imagebuilder para obter os detalhes de uma política da fórmula de contêiner especificando seu ARN.

```
aws imagebuilder get-container-recipe-policy --container-recipe-arn
 arn:aws:imagebuilder:us-west-2:123456789012:container-recipe/my-example-
recipe/2020.12.03
```

Criar uma nova versão de uma fórmula de imagem

Esta seção descreve como criar uma nova versão de uma fórmula de imagem.

Conteúdo

- Criar uma versão de uma fórmula de imagem com o console
- Crie uma receita de imagem com o AWS CLI
- Importar uma VM como sua imagem de base no console

Criar uma versão de uma fórmula de imagem com o console

Quando você cria uma nova versão de fórmula, é praticamente o mesmo que criar uma nova fórmula. A diferença é que certos detalhes são pré-selecionados para corresponder à fórmula de

base, na maioria dos casos. A lista a seguir descreve as diferenças entre criar uma nova fórmula e criar uma nova versão de uma fórmula existente.

Detalhes da fórmula de base na nova versão

- Nome: não editável.
- Versão: obrigatória. Insira o número da versão que você deseja criar no formato <major>.<minor>.<patch>. O Image Builder oferece suporte ao incremento automático de versões para receitas, permitindo que você use padrões curinga em suas versões de receitas. Quando você cria uma receita com uma versão curinga`1.0.x`, como, por exemplo, Image Builder incrementa automaticamente a versão (por exemplo,, `1.0.1` `1.0.21.0.3`, e assim por diante). Isso elimina a necessidade de rastrear e incrementar manualmente as versões da receita.
- A opção Selecionar imagem: pré-selecionada, mas você pode editá-la. Se você alterar sua escolha para a fonte da imagem de base, você poderá perder outros detalhes que dependem da opção original escolhida por você.

  Para ver detalhes associados à seleção imagem base, escolha a guia que corresponde à sua seleção.

  Managed image

  - Sistema operacional (OS) de imagem: não editável.
  - Nome da imagem: pré-selecionado, com base na combinação de opções de imagem de base que você fez para a fórmula existente. No entanto, se você alterar a opção Selecionar imagem, você perderá o Nome da imagem pré-selecionado.
  - Opções de autoversionamento: não corresponde à sua fórmula de base. Essa opção de imagem usa como padrão a opção Usar versão do OS selecionada.

  > **Important**
  >
  > Se você estiver usando o auto-versionamento semântico para iniciar as compilações do pipeline, certifique-se de alterar esse valor para Usar a versão do OS mais recente disponível. Para saber mais sobre o versionamento semântico dos recursos do Image Builder, consulte Versionamento semântico no Image Builder.

  AWS Marketplace image

  - Assinaturas — Essa guia deve estar aberta e a imagem assinada AWS Marketplace deve ser pré-selecionada para corresponder à sua receita base. Se você alterar a imagem que

sua fórmula usa como imagem de base, você pode perder outros detalhes que dependem da imagem original que você escolheu.

Para obter mais informações sobre AWS Marketplace produtos, consulte Comprar produtos no Guia do AWS Marketplace comprador.

Custom AMI

Fonte da AMI (obrigatória) - insira a ID da AMI ou um parâmetro AWS Systems Manager (SSM) do Parameter Store que contém uma ID da AMI para usar como imagem base. O agente SSM deve estar pré-instalado na AMI selecionada.

- ID AMI — Essa configuração não está pré-preenchida com sua entrada original. Insira o ID da AMI para sua imagem base. Exemplo: *ami-1234567890abcdef1*.
- Parâmetro SSM — Insira o nome ou ARN do parâmetro SSM Parameter Store que contém o ID da AMI para sua imagem base. Exemplo: */ib/test/param* ou `arn:aws:ssm:`*us-east-1*`:`*111122223333*`:parameter`*/ib/test/param*.

- Configuração da instância: as configurações são pré-selecionadas, mas você pode editá-las.
  - Atendente do Systems Manager: você pode marcar ou desmarcar essa caixa de seleção para controlar a instalação do atendente do Systems Manager na nova imagem. A caixa de seleção é desmarcada por padrão para incluir o atendente do Systems Manager em sua nova imagem. Para remover o atendente do Systems Manager da imagem final, marque a caixa de seleção para que o atendente não seja incluído na sua AMI.
  - Dados do usuário: você pode usar essa área para fornecer comandos, ou um script de comando a executar, quando você inicializa sua instância de compilação. No entanto, esse valor substitui qualquer comando que o Image Builder possa ter adicionado para garantir que o Systems Manager seja instalado. Esses comandos incluem o script de limpeza que o Image Builder normalmente executa para imagens do Linux antes de criar a nova imagem.

    Quando o Image Builder inicia uma instância, os scripts de dados do usuário são executados durante a fase de inicialização da nuvem, antes do início da execução do componente. Essa etapa é registrada no seguinte arquivo na instância:`var/log/cloud-init.log`.

    > **Note**
    >
    > - Se você inserir dados do usuário, certifique-se de que o atendente do Systems Manager esteja pré-instalado na imagem de base ou que você inclua a instalação nos dados do usuário.

- Para imagens do Linux, certifique-se de que as etapas de limpeza sejam executadas incluindo um comando para criar um arquivo vazio nomeado `perform_cleanup` no script de dados do usuário. O Image Builder detecta esse arquivo e executa o script de limpeza antes de criar a nova imagem. Para obter mais informações e um exemplo de script, consulte Práticas recomendadas de segurança para o Image Builder.

- Diretório de trabalho: pré-selecionado, mas você pode editá-lo.
- Componentes: os componentes que já estão incluídos na fórmula são exibidos na seção Componentes selecionados no final de cada uma das listas de componentes (compilação e teste). Você pode remover ou reordenar os componentes selecionados para atender às suas necessidades.

  Os componentes de endurecimento do CIS não seguem as regras padrão de ordenação de componentes nas fórmulas do Image Builder. Os componentes de endurecimento do CIS sempre são executados por último para garantir que os testes de benchmark sejam executados em relação à sua imagem de saída.

  > **Note**
  >
  > As listas de componentes de compilação e teste exibem os componentes disponíveis com base no tipo de proprietário do componente. Para adicionar um componente, escolha Adicionar componentes de construção e selecione o filtro de propriedade aplicável. Por exemplo, para adicionar um componente de compilação associado a um AWS Marketplace produto, selecione`AWS Marketplace`. Isso abre um painel de seleção no lado direito da interface do console que lista AWS Marketplace os componentes.
  > Para o componente CIS, selecione`Third party managed`.

  Você pode definir as seguintes configurações para seu componente selecionado:

  - Opções de versionamento: pré-selecionadas, mas você pode alterá-las. Recomendamos que você escolha a opção Usar a versão mais recente disponível do componente para garantir que suas compilações de imagem sempre usem a versão mais recente do componente. Se precisar usar uma versão específica do componente em sua fórmula, você pode escolher Especificar versão do componente e inserir a versão na caixa Versão do componente que aparece.
  - Parâmetros de entrada: exibe os parâmetros de entrada que o componente aceita. O Valor é pré-preenchido com o valor da versão anterior da fórmula. Se você estiver usando esse

componente pela primeira vez nessa fórmula e um valor padrão tiver sido definido para o parâmetro de entrada, o valor padrão aparecerá na caixa Valor com texto acinzentado. Se nenhum outro valor for inserido, o Image Builder usará o valor padrão.

Se um parâmetro de entrada for necessário, mas não tiver um valor padrão definido no componente, você deverá fornecer um valor. O Image Builder não criará a versão da fórmula se algum parâmetro obrigatório estiver faltando e não houver um valor padrão definido.

> **Important**
>
> Os parâmetros do componente são valores de texto simples e estão logados em AWS CloudTrail. Recomendamos que você use AWS Secrets Manager ou o AWS Systems Manager Parameter Store para armazenar seus segredos. Para obter mais informações sobre o Secrets Manager, consulte O que é o Secrets Manager? no AWS Secrets Manager Guia do usuário. Para obter mais informações sobre o AWS Systems Manager Parameter Store, consulte AWS Systems Manager Parameter Store no AWS Systems Manager Guia do usuário.

Para expandir as configurações das opções de versionamento ou dos parâmetros de entrada, você pode escolher a seta ao lado do nome da configuração. Para expandir todas as configurações de todos os componentes selecionados, você pode ativar e desativar a opção Expandir tudo.

- Armazenamento (volumes): são pré-preenchidos. As seleções Nome do dispositivo, Snapshot e IOPS do volume raiz não são editáveis. No entanto, você pode alterar todas as demais configurações, como o Tamanho. Você também pode adicionar novos volumes e criptografar volumes novos ou existentes.

  Para criptografar volumes para as imagens que o Image Builder cria em sua conta na região de origem (onde a compilação é executada), você deve usar a criptografia do volume de armazenamento na fórmula de imagem. A criptografia executada durante a fase de distribuição da compilação é somente para imagens que são distribuídas para outras contas ou regiões.

  > **Note**
  >
  > Se você usar criptografia para seus volumes, deverá selecionar a chave de cada volume separadamente, mesmo que a chave seja a mesma usada para o volume-raiz.

Para criar uma nova versão de fórmula de imagem:

1. Na parte superior da página de detalhes da fórmula, escolha Criar nova versão. Isso leva você à página Criar fórmula de imagem.
2. Para criar a nova versão, faça suas alterações e depois escolha Criar fórmula.

   Sua imagem final pode conter até quatro códigos de produto de produtos e componentes de AWS Marketplace imagem. Se a imagem base e os componentes selecionados contiverem mais de quatro códigos de produto, o Image Builder retornará um erro quando você tentar criar a receita.

Para obter mais informações sobre como criar uma fórmula de imagem ao criar um pipeline de imagens, consulte Etapa 2: Escolher fórmula na seção Conceitos básicos deste guia.

Crie uma receita de imagem com o AWS CLI

Para criar uma receita de imagem com o `create-image-recipe` comando Image Builder no AWS CLI, siga estas etapas:

Pré-requisitos

Antes de executar os comandos do Image Builder nesta seção para criar uma receita de imagem a partir do AWS CLI, você pode, opcionalmente, criar componentes que a receita usa. O exemplo de fórmula de imagem na etapa a seguir se refere a exemplos de componentes que são criados na seção Crie um componente personalizado a partir do AWS CLI deste guia.

Se você quiser incluir componentes em sua receita, anote o ARNs que você deseja incluir. Você também pode criar receitas sem nenhum componente para testar fluxos de trabalho existentes AMIs ou somente de distribuição.

1. Criar um arquivo JSON de entrada da CLI

   Você pode fornecer toda a entrada para o comando create-image-recipe com parâmetros de comando embutidos. No entanto, o comando resultante pode ser bastante longo. Para simplificar o comando, você pode fornecer um arquivo JSON que contenha todas as configurações da fórmula.

> **Note**
>
> A convenção de nomenclatura para os valores de dados no arquivo JSON segue o padrão especificado para os parâmetros de solicitação de operação da API Image Builder. Para revisar os parâmetros da solicitação de operação da API, consulte o CreateImageRecipecomando na Referência da API EC2 Image Builder.
> Para fornecer os valores dos dados como parâmetros da linha de comando, consulte os nomes dos parâmetros especificados na AWS CLI Referência de comando.

Aqui está um resumo dos parâmetros que esses exemplos especificam:

- name (string, obrigatório): o nome da fórmula de imagem.
- description (string): a descrição da fórmula de imagem.
- parentImage (string, obrigatório): a imagem que a fórmula de imagem usa como base para sua imagem personalizada. Você pode especificar a imagem principal usando uma das seguintes opções:
  - ID DA AMI
  - ARN do recurso de imagem do Image Builder
  - AWS Systems Manager (SSM) Parâmetro do Parameter Store, prefixado por`ssm:`, seguido pelo nome do parâmetro ou ARN.
  - AWS Marketplace ID do produto

  > **Note**
  >
  > Os exemplos de Linux e macOS usam uma AMI do Image Builder, e o exemplo do Windows usa um ARN.

- SemanticVersion <major>(string, obrigatório) — Insira o número da versão que você deseja criar no formato. <minor>. <patch>. O Image Builder oferece suporte ao incremento automático de versões para receitas, permitindo que você use padrões curinga em suas versões de receitas. Quando você cria uma receita com uma versão curinga`1.0.x`, como, por exemplo, Image Builder incrementa automaticamente a versão (por exemplo,, `1.0.1` `1.0.21.0.3`, e assim por diante). Isso elimina a necessidade de rastrear e incrementar

manualmente as versões da receita. Para saber mais sobre o versionamento semântico dos recursos do Image Builder, consulte Versionamento semântico no Image Builder.

- components (matriz, opcional) — Contém uma matriz de `ComponentConfiguration` objetos. Os componentes são opcionais - você pode criar receitas sem nenhum componente para fluxos de trabalho de teste ou distribuição:

  > **Note**
  >
  > O Image Builder instala os componentes na ordem em que você os especificou na fórmula. No entanto, os componentes de fortalecimento do CIS sempre são executados por último para garantir que os testes de benchmark sejam executados em relação à sua imagem de saída.

  - componentARN (string, obrigatório): o ARN do componente.

    > **Tip**
    >
    > Para usar um dos exemplos para criar sua própria receita de imagem, você deve substituir o exemplo ARNs ARNs pelo dos componentes que você está usando para sua receita.

  - parameters (matriz de objetos): contém uma variedade de objetos `ComponentParameter`. Se um parâmetro de entrada for necessário, mas não tiver um valor padrão definido no componente, você deverá fornecer um valor. O Image Builder não criará a versão da fórmula se algum parâmetro obrigatório estiver faltando e não houver um valor padrão definido.

    > **Important**
    >
    > Os parâmetros do componente são valores de texto simples e estão logados em AWS CloudTrail. Recomendamos que você use AWS Secrets Manager ou o AWS Systems Manager Parameter Store para armazenar seus segredos. Para obter mais informações sobre o Secrets Manager, consulte O que é o Secrets Manager? no AWS Secrets Manager Guia do usuário. Para obter mais informações sobre o AWS Systems Manager Parameter Store, consulte AWS Systems Manager Parameter Store no AWS Systems Manager Guia do usuário.

- nome (string, obrigatório): o nome do parâmetro do componente a ser definido.
- value (matriz de strings, obrigatório): contém uma matriz de strings para definir o valor do parâmetro do componente nomeado. Se houver um valor padrão definido para o componente e nenhum outro valor for fornecido, AWSTOE use o valor padrão.

- additionalInstanceConfiguration(objeto) — Especifique configurações adicionais e scripts de lançamento para suas instâncias de compilação.
  - systemsManagerAgent(object) — Contém configurações para o agente Systems Manager em sua instância de compilação.
    - uninstallAfterBuild(Boolean) — Controla se o agente do Systems Manager é removido da imagem de compilação final antes de criar a nova AMI. Se essa opção for definida para `true`, o atendente será removido da imagem final. Se a opção for definida para `false`, o atendente será mantido para que seja incluído na nova AMI. O valor padrão é `false`.

      > **Note**
      >
      > Se o atributo `uninstallAfterBuild` não estiver incluído no arquivo JSON e as condições a seguir forem verdadeiras, o Image Builder removerá o atendente do Systems Manager da imagem final para que ele não fique disponível na AMI:
      >
      > - O `userDataOverride` está vazio ou foi omitido do arquivo JSON.
      > - O Image Builder instalou automaticamente o atendente do Systems Manager na instância de compilação de um sistema operacional que não tinha o atendente pré-instalado na imagem de base.

  - userDataOverride(string) — Forneça comandos ou um script de comando para executar ao iniciar sua instância de compilação.

    > **Note**
    >
    > Os dados do usuário são sempre codificados em base 64.
    > Por exemplo, os seguintes comandos são codificados como `IyEvYmluL2Jhc2gKbWtkaXIgLXAgL3Zhci9iYi8KdG91Y2ggL3Zhcg==`:
    >
    > ```
    > #!/bin/bash
    > mkdir -p /var/bb/
    > touch /var
    > ```

O exemplo do Linux usa esse valor codificado.

Linux

A imagem base (propriedade de `parentImage`) no exemplo a seguir é uma AMI. Ao usar uma AMI, você deve ter acesso à AMI, e a AMI deve estar na região de origem (a mesma região em que o Image Builder executa o comando). Salve o arquivo como `create-image-recipe.json` e use-o no comando create-image-recipe.

```
{
"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

O exemplo a seguir se refere à versão mais recente da imagem de base completa em inglês do Windows Server 2016. O ARN neste exemplo faz referência à imagem mais recente com base nos filtros de versão semântica que você especificou:. `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**
>
> Para saber mais sobre o versionamento semântico dos recursos do Image Builder, consulte Versionamento semântico no Image Builder.

macOS

A imagem base (propriedade de `parentImage`) no exemplo a seguir é uma AMI. Ao usar uma AMI, você deve ter acesso à AMI, e a AMI deve estar na região de origem (a mesma região em que o Image Builder executa o comando). Salve o arquivo como `create-image-recipe.json` e use-o no comando create-image-recipe.

```
{
"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=="
}
}
```

Exemplo: Receita sem componentes

Você pode criar receitas sem nenhum componente para testar fluxos de trabalho existentes AMIs ou somente de distribuição. O exemplo a seguir mostra uma receita que usa uma AMI existente sem aplicar nenhum componente adicional:

```
{
 "name": "Test Distribution Recipe",
 "description": "Recipe for testing and distributing existing AMI without
 modifications.",
 "parentImage": "ami-1234567890abcdef1",
 "semanticVersion": "1.0.0",
 "additionalInstanceConfiguration": {
  "systemsManagerAgent": {
    "uninstallAfterBuild": true
  }
 }
}
```

2. Criar a fórmula

Use o comando a seguir para criar a fórmula. Forneça o nome do arquivo JSON que você criou na etapa anterior no parâmetro `--cli-input-json`:

```
aws imagebuilder create-image-recipe --cli-input-json file://create-image-
recipe.json
```

> **Note**
>
> - É necessário incluir a notação `file://` no início do caminho do arquivo JSON.
> - O caminho para o arquivo JSON deve seguir a convenção apropriada para o sistema operacional de base no qual você está executando o comando. Por exemplo, o Windows usa a barra invertida (\) para se referir ao caminho do diretório, enquanto o Linux e o macOS usam a barra (/).

Sua imagem final pode conter até quatro códigos de produto de produtos e componentes de AWS Marketplace imagem. Se a imagem base e os componentes selecionados contiverem mais de quatro códigos de produto, o Image Builder retornará um erro ao executar o `create-image-recipe` comando.

Importar uma VM como sua imagem de base no console

Nesta seção, vamos nos concentrar em como importar uma máquina virtual (VM) como imagem de base para sua fórmula de imagem. Aqui, não abordamos outras etapas envolvidas na criação de uma fórmula ou versão da fórmula. Para ver as etapas adicionais para criar uma nova fórmula de imagem com o assistente de criação de pipeline no console do Image Builder, consulte Assistente de pipeline: criar AMI. Para ver as etapas adicionais para criar uma nova fórmula de imagem ou versão de fórmula, consulte Criar uma nova versão de uma fórmula de imagem.

Para importar uma VM como imagem de base para sua fórmula de imagem no console do Image Builder, siga estas etapas, junto com todas as outras etapas necessárias, para criar sua fórmula ou versão da fórmula.

1. Na seção Selecionar imagem para a imagem de base, selecione a opção Importar imagem de base.
2. Escolha o Sistema operacional (OS) de imagem e a Versão do OS como você faria normalmente.

Configuração de importação de VM

Quando você exporta sua VM do ambiente de virtualização, esse processo cria um conjunto de um ou mais arquivos de contêiner de disco que atuam como snapshots do ambiente, das configurações e dos dados da sua VM. Você pode usar esses arquivos para importar sua VM como imagem de base para sua fórmula de imagem. Para obter mais informações sobre importação VMs no Image Builder, consulte Importar e exportar imagens de VM

Para especificar a localização da sua fonte de importação, siga estas etapas:

Importar fonte

Especifique a fonte do primeiro contêiner de disco ou snapshot de imagem da VM a ser importado na seção Contêiner de disco 1.

1. Fonte: essa pode ser um bucket do S3 ou um snapshot do EBS.
2. Selecione a localização do disco do S3: insira a localização no Amazon S3 onde suas imagens de disco estão armazenadas. Para procurar o local, escolha Browse S3.
3. Para adicionar um contêiner de disco, escolha Adicionar contêiner de disco.

Perfil do IAM

Para associar um Perfil do IAM à sua configuração de importação de VM, selecione o perfil na lista suspensa de Perfil do IAM ou escolha Criar novo perfil para criar um novo. Se você criar um novo perfil, a página do console de Perfis do IAM será aberta em uma guia separada.

Configurações avançadas: opcionais

As seguintes configurações são opcionais. Com essas configurações, você pode configurar criptografia, licenciamento, tags e muito mais para a imagem de base criada pela importação.

Geral

1. Especifique um Nome exclusivo para a imagem de base. Se você não inserir um valor, a imagem de base herdará o nome da fórmula.
2. Especifique uma Versão para a imagem de base. Use o formato a seguir: *<major>.<minor>.<patch>*. Se você não inserir um valor, a imagem de base herdará a versão da fórmula.
3. Você também pode inserir uma Descrição para a imagem de base.

Arquitetura da imagem de base

Para especificar a arquitetura da sua fonte de importação de VM, selecione um valor na lista Arquitetura.

Criptografia

Se as imagens de disco da VM estiverem criptografadas, você deverá fornecer uma chave para usar no processo de importação. Para especificar um AWS KMS key para a importação, selecione um valor na lista Criptografia (chave KMS). A lista contém chaves KMS às quais sua conta tem acesso na região atual.

Gerenciamento de licença

Quando você importa uma VM, o processo de importação detecta automaticamente o OS da VM e aplica a licença apropriada à imagem de base. Dependendo da plataforma do seu OS, os tipos de licença são os seguintes:

- Licença incluída: Uma licença AWS apropriada para sua plataforma é aplicada à sua imagem de base.
- Traga a sua própria licença (BYOL): Retém a licença da sua VM, se aplicável.

Para anexar as configurações de licença criadas com AWS License Manager à sua imagem base, selecione na lista Nome da configuração da licença. Para obter mais informações sobre o License Manager, consulte Trabalhando com AWS License Manager

> **Note**
>
> - As configurações de licença contêm regras de licenciamento com base nos termos de seus contratos empresariais.
> - O Linux só oferece suporte a licenças BYOL.

Tags (imagem de base)

As tags usam pares de chave–valor para atribuir texto pesquisável ao seu recurso do Image Builder. Para especificar tags para a imagem de base importada, insira pares de chave–valor com as caixas Chave e Valor.

Para adicionar uma tag, escolha Adicionar tag. Para remover uma tag, selecione Remover tag.

Criar uma nova versão de fórmula de contêiner

Esta seção mostra como criar uma nova versão de fórmula de contêiner.

Conteúdo

- Crie uma nova versão de fórmula de contêiner com o console
- Criar uma fórmula de contêiner com o AWS CLI

Crie uma nova versão de fórmula de contêiner com o console

Criar uma nova versão de uma fórmula do contêiner é praticamente o mesmo que criar uma nova fórmula. A diferença é que certos detalhes são pré-selecionados para corresponder à fórmula de base, na maioria dos casos. A lista a seguir descreve as diferenças entre criar uma nova fórmula e criar uma nova versão de uma fórmula existente.

Detalhes da fórmula

- Nome: não editável.
- Versão: obrigatória. Esse detalhe não é pré-preenchido com a versão atual nem com qualquer tipo de sequência. Insira o número da versão que você deseja criar no formato major.minor.patch. Se a versão já existir, você encontrará um erro.

Imagem de base

- Opção Selecionar imagem: pré-selecionada, mas você pode editá-la. Se você alterar sua escolha para a fonte da imagem de base, você poderá perder outros detalhes que dependem da opção original escolhida por você.

  Para imagens de contêiner do Docker, você pode escolher entre imagens públicas hospedadas em DockerHub, imagens de contêiner existentes no Amazon ECR ou imagens de contêiner gerenciadas pela Amazon. Para ver detalhes associados à seleção imagem base, escolha a guia que corresponde à sua seleção.

  Managed images

  - Sistema operacional (OS) de imagem: não editável.
  - Nome da imagem: pré-selecionado, com base na combinação de opções de imagem de base que você fez para a fórmula existente. No entanto, se você alterar a opção Selecionar imagem, você perderá o Nome da imagem pré-selecionado.
  - Opções de autoversionamento: não corresponde à sua fórmula de base. As opções de autoversionamento usam como padrão a opção Usar a versão do sistema operacional selecionada.

    > **Important**
    >
    > Se você estiver usando o versionamento semântico para iniciar as compilações do pipeline, altere esse valor para Usar a versão mais recente do sistema operacional

disponível. Para saber mais sobre o versionamento semântico dos recursos do Image Builder, consulte Versionamento semântico no Image Builder.

ECR image

- Sistema operacional de imagem (OS): pré-selecionado, mas editável.
- Versão do OS: pré-selecionada, mas editável.
- ID de imagem ECR: pré-preenchido, mas editável.

Docker Hub image

- Sistema operacional (OS) de imagem: não editável.
- Versão do OS: pré-selecionada, mas editável.
- ID de imagem do Docker: pré-preenchido, mas editável.

Configuração da instância

- Fonte da AMI (obrigatória) — identifique uma AMI personalizada para usar como imagem base para a criação e a instância de teste do contêiner. Isso pode ser uma ID de AMI ou um parâmetro AWS Systems Manager (SSM) do Parameter Store que contém uma ID de AMI.
  - ID AMI — Essa configuração não está pré-preenchida com sua entrada original. Insira o ID da AMI para sua imagem base. Exemplo: *ami-1234567890abcdef1*.
  - Parâmetro SSM — Insira o nome ou ARN do parâmetro SSM Parameter Store que contém o ID da AMI para sua imagem base. Exemplo: */ib/test/param* ou `arn:aws:ssm:`*us-east-1*`:`*111122223333*`:parameter`*/ib/test/param*.
- Armazenamento (volumes)

  Volume 1 do EBS (raiz da AMI): pré-preenchido. As seleções Nome do dispositivo, Snapshot e IOPS do volume raiz não podem ser editados. No entanto, você pode alterar todas as demais configurações, como o Tamanho. Você também pode adicionar novos volumes.

> **Note**
>
> Se você especificar uma AMI base compartilhada de outra conta com você, os snapshots de qualquer volume secundário especificado também devem ser compartilhados com sua conta.

Diretório de trabalho

- Caminho do diretório de trabalho: pré-preenchido, mas editável.

Componentes

- Componentes: os componentes que já estão incluídos na fórmula são exibidos na seção Componentes selecionados no final de cada uma das listas de componentes (compilação e teste). Você pode remover ou reordenar os componentes selecionados para atender às suas necessidades.

  Os componentes de endurecimento do CIS não seguem as regras padrão de ordenação de componentes nas fórmulas do Image Builder. Os componentes de endurecimento do CIS sempre são executados por último para garantir que os testes de benchmark sejam executados em relação à sua imagem de saída.

  > **Note**
  >
  > As listas de componentes de compilação e teste exibem os componentes disponíveis com base no tipo de proprietário do componente. Para adicionar um componente, escolha Adicionar componentes de construção e selecione o filtro de propriedade aplicável. Por exemplo, para adicionar um componente de compilação associado a um AWS Marketplace produto, selecione`AWS Marketplace`. Isso abre um painel de seleção no lado direito da interface do console que lista AWS Marketplace os componentes.
  > Para o componente CIS, selecione`Third party managed`.

  Você pode definir as seguintes configurações para seu componente selecionado:

  - Opções de versionamento: pré-selecionadas, mas você pode alterá-las. Recomendamos que você escolha a opção Usar a versão mais recente disponível do componente para garantir que suas compilações de imagem sempre usem a versão mais recente do componente. Se precisar usar uma versão específica do componente em sua fórmula, você pode escolher Especificar versão do componente e inserir a versão na caixa Versão do componente que aparece.
  - Parâmetros de entrada: exibe os parâmetros de entrada que o componente aceita. O Valor é pré-preenchido com o valor da versão anterior da fórmula. Se você estiver usando esse componente pela primeira vez nessa fórmula e um valor padrão tiver sido definido para o

parâmetro de entrada, o valor padrão aparecerá na caixa Valor com texto acinzentado. Se nenhum outro valor for inserido, o Image Builder usará o valor padrão.

Se um parâmetro de entrada for necessário, mas não tiver um valor padrão definido no componente, você deverá fornecer um valor. O Image Builder não criará a versão da fórmula se algum parâmetro obrigatório estiver faltando e não houver um valor padrão definido.

> **Important**
>
> Os parâmetros do componente são valores de texto simples e estão logados em AWS CloudTrail. Recomendamos que você use AWS Secrets Manager ou o AWS Systems Manager Parameter Store para armazenar seus segredos. Para obter mais informações sobre o Secrets Manager, consulte O que é o Secrets Manager? no AWS Secrets Manager Guia do usuário. Para obter mais informações sobre o AWS Systems Manager Parameter Store, consulte AWS Systems Manager Parameter Store no AWS Systems Manager Guia do usuário.

Para expandir as configurações das opções de versionamento ou dos parâmetros de entrada, você pode escolher a seta ao lado do nome da configuração. Para expandir todas as configurações de todos os componentes selecionados, você pode ativar e desativar a opção Expandir tudo.

Modelo do Dockerfile

- Modelo do Dockerfile: pré-preenchido, mas editável. Você pode especificar qualquer uma das variáveis contextuais a seguir que o Image Builder substitui por informações de compilação no runtime.

  parentImage (obrigatória)

  No momento da compilação, esta variável é resolvida na imagem de base da sua fórmula.

  Exemplo:

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

environments (necessários se os componentes forem especificados)

Esta variável será resolvida em um script que executa componentes.

Exemplo:

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

components (opcional)

O Image Builder resolve scripts de componentes de compilação e teste para os componentes que a fórmula de contêiner inclui. Esta variável pode ser colocada em qualquer lugar no Dockerfile, depois da variável environments.

Exemplo:

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

Repositório de destino

- Nome do repositório de destino: o repositório Amazon ECR onde sua imagem de saída está armazenada se não houver outro repositório especificado na configuração de distribuição do seu pipeline para a região onde o pipeline é executado (Região 1).

Para criar uma nova fórmula de contêiner:

1. Na parte superior da página de detalhes da fórmula do contêiner, escolha Criar nova versão. Você será direcionado para a página Criar fórmula para fórmulas de contêiner.
2. Para criar a nova versão, faça suas alterações e depois escolha Criar fórmula.

Para obter mais informações sobre como criar uma fórmula de contêiner ao criar um pipeline de imagens, consulte Etapa 2: Escolher fórmula a seção Introdução deste guia.

Criar uma fórmula de contêiner com o AWS CLI

Para criar uma receita de contêiner do Image Builder com o `imagebuilder create-container-recipe` comando no AWS CLI, siga estas etapas:

Pré-requisitos

Antes de executar os comandos do Image Builder nesta seção para criar uma receita de contêiner com o AWS CLI, você deve criar os componentes que a receita usará. O exemplo de fórmula de contêiner na etapa a seguir se refere a exemplos de componentes que são criados na seção Crie um componente personalizado a partir do AWS CLI deste guia.

Depois de criar seus componentes, ou se você estiver usando componentes existentes, anote o ARNs que você deseja incluir na receita.

1. Criar um arquivo JSON de entrada da CLI

   Você pode fornecer toda a entrada para o comando create-container-recipe com parâmetros de comando embutidos. No entanto, o comando resultante pode ser bastante longo. Para simplificar o comando, você pode fornecer um arquivo JSON que contenha todas as configurações da fórmula do contêiner

   > **Note**
   >
   > A convenção de nomenclatura para os valores de dados no arquivo JSON segue o padrão especificado para os parâmetros de solicitação de operação da API Image Builder. Para revisar os parâmetros da solicitação de operação da API, consulte o CreateContainerRecipecomando na Referência da API EC2 Image Builder.
   > Para fornecer os valores dos dados como parâmetros da linha de comando, consulte os nomes dos parâmetros especificados na AWS CLI Referência de comando.

   Aqui está um resumo dos parâmetros neste exemplo:

   - componentes (matriz de objetos, obrigatório): contém uma matriz de `ComponentConfiguration` objetos. Pelo menos um componente da compilação deve ser especificado:

     > **Note**
     >
     > O Image Builder instala os componentes na ordem em que você os especificou na fórmula. No entanto, os componentes de fortalecimento do CIS sempre são

executados por último para garantir que os testes de benchmark sejam executados em relação à sua imagem de saída.

- componentARN (string, obrigatório): o ARN do componente.

  > **Tip**
  >
  > Para usar o exemplo para criar sua própria receita de contêiner, substitua o ARNs exemplo ARNs pelo dos componentes que você está usando para sua receita,. Isso inclui o Região da AWS, nome, e o número da versão de cada um.

- parameters (matriz de objetos): contém uma variedade de objetos `ComponentParameter`. Se um parâmetro de entrada for necessário, mas não tiver um valor padrão definido no componente, você deverá fornecer um valor. O Image Builder não criará a versão da fórmula se algum parâmetro obrigatório estiver faltando e não houver um valor padrão definido.

  > **Important**
  >
  > Os parâmetros do componente são valores de texto simples e estão logados em AWS CloudTrail. Recomendamos que você use AWS Secrets Manager ou o AWS Systems Manager Parameter Store para armazenar seus segredos. Para obter mais informações sobre o Secrets Manager, consulte O que é o Secrets Manager? no AWS Secrets Manager Guia do usuário. Para obter mais informações sobre o AWS Systems Manager Parameter Store, consulte AWS Systems Manager Parameter Store no AWS Systems Manager Guia do usuário.

  - nome (string, obrigatório): o nome do parâmetro do componente a ser definido.
  - value (matriz de strings, obrigatório): contém uma matriz de strings para definir o valor do parâmetro do componente nomeado. Se houver um valor padrão definido para o componente e nenhum outro valor for fornecido, AWSTOE use o valor padrão.

- ContainerType (string, obrigatório): o tipo de contêiner a ser criado. Os valores válidos são: `DOCKER`.
- dockerfileTemplateData(string) — O modelo Dockerfile usado para criar sua imagem, expresso como um blob de dados embutido.

- name (string, obrigatório): o nome da fórmula do contêiner.
- description (string): a descrição da fórmula do contêiner.
- parentImage (string, required) — A imagem do contêiner Docker a ser usada na receita do contêiner como base para sua imagem personalizada.
  - Imagens públicas hospedadas em DockerHub
  - Imagens de contêiner existentes no Amazon ECR
  - Imagens de contêiner gerenciadas pela Amazon
- platformOverride (string) - Especifica a plataforma do sistema operacional quando você usa uma imagem base personalizada.
- SemanticVersion (string, obrigatório): a versão semântica da fórmula do contêiner especificada no formato a seguir, com valores numéricos em cada posição para indicar uma versão específica: <major>,<minor>,<patch> . Um exemplo seria `1.0.0`. Para saber mais sobre o versionamento semântico dos recursos do Image Builder, consulte Versionamento semântico no Image Builder.
- tags (string map) - Tags que estão anexadas à fórmula do contêiner.
- instanceConfiguration (objeto) - Um grupo de opções que pode ser utilizado para configurar uma instância para compilar e testar imagens de contêiner.
  - image (string) — A imagem base para uma instância de criação e teste de contêiner. Isso pode conter uma ID de AMI ou especificar um parâmetro AWS Systems Manager (SSM) do Parameter Store, prefixado por`ssm:`, seguido pelo nome do parâmetro ou ARN. Se você usar um parâmetro SSM, o valor do parâmetro deverá conter uma ID de AMI. Se você não especificar uma imagem base, o Image Builder usa a AMI otimizada apropriada do Amazon ECS como imagem base.
  - blockDeviceMappings(matriz de objetos) — Define os dispositivos de bloco a serem conectados para criar uma instância a partir da AMI do Image Builder especificada no image parâmetro.
    - deviceName (string): o dispositivo ao qual esses mapeamentos se aplicam.
    - ebs (objeto) - Use para gerenciar a configuração específica do Amazon EBS para esse mapeamento.
      - deleteOnTermination(Boolean) — Usado para configurar a exclusão no encerramento do dispositivo associado.
      - encrypted (booleano): usado para configurar a criptografia do dispositivo.
      - volumeSize (inteiro): usado para substituir o tamanho do volume do dispositivo

    - volumeSize (string): usado para substituir o tipo do volume do dispositivo.
- targetRepository (objeto, obrigatório): o repositório de destino da imagem do contêiner, se não houver outro repositório especificado na configuração de distribuição do pipeline para a região onde o pipeline é executado (Região 1).
  - repositoryName (string, obrigatório) - O nome do repositório de contêiner onde a imagem do contêiner de saída é armazenada. Este nome é prefixado pelo local do repositório.
  - serviço (string, obrigatório) - Especifica o serviço no qual esta imagem foi registrada.
- workingDirectory (string) - O diretório de trabalho a ser usado durante os fluxos de trabalho de compilação e teste.

```
{
 "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. Criar a fórmula

   Use o comando a seguir para criar a fórmula. Forneça o nome do arquivo JSON que você criou na etapa anterior no parâmetro `--cli-input-json`:

   ```
   aws imagebuilder create-container-recipe --cli-input-json file://create-container-recipe.json
   ```

   > **Note**
   >
   > - É necessário incluir a notação `file://` no início do caminho do arquivo JSON.
   > - O caminho para o arquivo JSON deve seguir a convenção apropriada para o sistema operacional de base no qual você está executando o comando. Por exemplo, o Windows usa a barra invertida (\) para se referir ao caminho do diretório, enquanto o Linux e o macOS usam a barra (/).

Limpar recursos

Para evitar cobranças inesperadas, certifique-se de limpar os recursos e pipelines que você criou com base nos exemplos deste guia. Para obter mais informações sobre exclusão de recursos no Image Builder, consulte Excluir recursos desatualizados ou não utilizados do Image Builder.

Gerenciar a configuração da infraestrutura do Image Builder

Você pode usar configurações de infraestrutura para especificar a EC2 infraestrutura da Amazon que o Image Builder usa para criar e testar sua EC2 imagem do Image Builder. As configurações de infraestrutura incluem:

- Tipos de instância para sua infraestrutura de compilação e teste. Recomendamos que você especifique mais de um tipo de instância, pois isso permite que o Image Builder execute uma instância a partir de um grupo com capacidade suficiente. Isto pode reduzir suas falhas transitórias de compilação.

  Para imagens do Mac, é possível escolher tipos de instância que ofereçam suporte nativo ao sistema operacional macOS. Para obter mais informações, consulte as instâncias do Amazon EC2 Mac no Guia EC2 do usuário da Amazon.
- Configurações de posicionamento de instância nas quais você pode especificar o host, o grupo de posicionamento do host ou a zona de disponibilidade para onde devem ir as instâncias executadas em sua imagem.
- Um perfil de instância que fornece às suas instâncias de compilação e teste as permissões necessárias para realizar atividades de personalização. Por exemplo, se você tem um componente que recupera recursos do Amazon S3, o perfil de instância exige permissões para acessar esses arquivos. O perfil da instância também exige um conjunto mínimo de permissões para que o EC2 Image Builder se comunique com sucesso com a instância. Para obter mais informações, consulte Configurar compilação de imagens personalizadas com o Image Builder.
- A VPC, a sub-rede e grupos de segurança para as instâncias de compilação e teste do seu pipeline.
- O local do Amazon S3 onde o Image Builder armazena os logs do aplicativo de sua compilação e teste. Se você configurar o registro em log, o perfil de instância especificado na configuração da sua infraestrutura deverá ter `s3:PutObject` permissões para o bucket de destino (`arn:aws:s3:::BucketName/*`).
- Um par de EC2 chaves da Amazon que permite que você faça login na sua instância para solucionar problemas se sua compilação falhar e você `terminateInstanceOnFailure` definir `false` como.
- Um tópico do SNS em que o Image Builder envia notificações de eventos. Para obter mais informações sobre como o Image Builder se integra ao Amazon SNS, consulte Integração do Amazon SNS no Image Builder.

> **Note**
>
> Se o tópico do SNS estiver criptografado, a chave que criptografa esse tópico deverá residir na conta em que o serviço Image Builder é executado. O Image Builder não pode enviar notificações para tópicos do SNS criptografados com chaves de outras contas.

Você pode criar e gerenciar as configurações da infraestrutura usando o console do Image Builder através do Image Builder API ou com os comandos imagebuilder no AWS CLI.

Conteúdo

- Listar e visualizar detalhes de uma configuração de infraestrutura
- Criar uma configuração de infraestrutura
- Atualizar uma configuração de infraestrutura
- Image Builder e AWS PrivateLink interface VPC endpoints

> **Tip**
>
> Quando você tem vários recursos do mesmo tipo, a marcação com tags ajuda a identificar um recurso específico com base nas tags que você atribuiu a ele. Para obter mais informações sobre como marcar seus recursos usando os comandos do Image Builder no AWS CLI, consulte a Marcar recursos seção deste guia.

Listar e visualizar detalhes de uma configuração de infraestrutura

Esta seção descreve as várias maneiras pelas quais você pode encontrar informações e visualizar detalhes das configurações de infraestrutura do EC2 Image Builder.

Detalhes da configuração de infraestrutura

- Listar as configurações de infraestrutura do AWS CLI
- Obtenha detalhes da configuração da infraestrutura no AWS CLI

Listar as configurações de infraestrutura do AWS CLI

O exemplo a seguir mostra como listar todas as suas configurações de infraestrutura por meio do comando list-infrastructure-configurations na AWS CLI.

```
aws imagebuilder list-infrastructure-configurations
```

Obtenha detalhes da configuração da infraestrutura no AWS CLI

O exemplo a seguir mostra como usar o get-infrastructure-configuration comando no AWS CLI para obter os detalhes de uma configuração de infraestrutura especificando seu 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
```

Criar uma configuração de infraestrutura

Esta seção descreve como você pode usar o console do Image Builder ou imagebuilder os comandos no AWS CLI para criar uma configuração de infraestrutura,

Console

Para criar um recurso de configuração de infraestrutura a partir do console Image Builder, siga estas etapas:

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. No painel de navegação, escolha Configuração de infraestrutura.
3. Escolha Criar configuração de infraestrutura.
4. Na seção Geral, insira as seguintes informações obrigatórias:
   - Insira o nome do seu recurso de configuração de infraestrutura.
   - Selecione um perfil do IAM que você deseja associar ao perfil de instância para permissões de componente em suas instâncias de compilação e teste. O Image Builder usa essas permissões para baixar e executar seus componentes, fazer upload de registros e realizar quaisquer ações adicionais especificadas pelos componentes em sua receita. CloudWatch
5. No painel Infraestrutura da AWS, você pode definir as configurações de infraestrutura restantes que estão disponíveis. Insira as seguintes informações obrigatórias:
   - Tipo de instância – Você pode especificar um ou mais tipos de instância para usar nessa compilação. O serviço escolherá um desses tipos de instância com base na disponibilidade.

> **Note**
>
> Instâncias do Mac são executadas em instâncias do tipo `.metal` em um host dedicado. Seu tipo de instância deve corresponder a um dos tipos definidos para o host em que ela é executada. Para obter mais informações sobre instâncias do Mac e uma lista dos tipos de instância que oferecem suporte nativo ao sistema operacional macOS, consulte Instâncias do EC2 Amazon Mac no Guia do usuário da EC2 Amazon.

- Tópico do SNS (opcional) — Selecione um tópico do SNS para receber notificações e alertas do EC2 Image Builder.

Se você não fornecer valores para as configurações a seguir, elas usarão padrões específicos do serviço, quando aplicável.

- VPC, sub-rede e grupos de segurança: o Image Builder usa sua VPC e a sub-rede padrão. Para obter mais informações sobre como configurar os endpoint da interface da VPC, consulte Image Builder e AWS PrivateLink interface VPC endpoints.
- Na seção Configurações para solução de problemas, você pode definir os seguintes valores:
  - Por padrão, a caixa de seleção Encerrar instância em caso de falha está marcada. No entanto, quando uma compilação falha, você pode fazer login na EC2 instância para solucionar o problema. Se você quiser que sua instância continue em execução após uma falha de compilação, desmarque a caixa de seleção.
  - Par de chaves — Se sua EC2 instância continuar em execução após uma falha de compilação, você poderá criar um par de chaves ou usar um par de chaves existente para fazer login na instância e solucionar problemas.
  - Logs: você pode especificar um bucket do S3 em que o Image Builder pode gravar registros de aplicativos para ajudar a solucionar problemas de compilação e testes. Se você não especificar um bucket do S3, o Image Builder grava os logs do aplicativo na instância.
- Na seção Configurações de metadados da instância, você pode configurar os seguintes valores para serem aplicados às EC2 instâncias que o Image Builder usa para criar e testar sua imagem:

- Selecione a versão dos metadados para determinar se EC2 requer um cabeçalho de token assinado para solicitações de recuperação de metadados da instância.
  - V1 e V2 (token opcional): valor padrão se você não selecionar nada.
  - V2 (token obrigatório)

  > **Note**
  >
  > Recomendamos que você configure todas as EC2 instâncias que o Image Builder executa a partir de um pipeline criado para uso, de IMDSv2 forma que as solicitações de recuperação de metadados da instância exijam um cabeçalho de token assinado.

  - Metadata token response hop limit (Limite de salto de resposta do token de metadados) - número permitido de saltos de rede para o token de metadados. Saltos mínimos: 1, saltos máximos: 64, com o padrão de um salto.
- Na seção Configurações de posicionamento da instância, você pode configurar os seguintes valores para serem aplicados às EC2 instâncias que o Image Builder usa para criar e testar sua imagem:
  - Você pode selecionar a Zona de disponibilidade em que o Image Builder inicia instâncias durante a criação da imagem.
  - Como opção, selecione Locação para os servidores que executam as instâncias que você inicia. Por padrão, EC2 as instâncias são executadas em hardware de locação compartilhada. Isso significa que várias Contas da AWS podem compartilhar o mesmo hardware físico. Uma instância com locação `dedicated` é executada em um hardware de locatário único. Uma instância com locação `host` é executada em um host dedicado.

    As instâncias do Mac exigem um host dedicado criado como pré-requisito antes de você compilar uma imagem personalizada. Selecione `host` para sua imagem do macOS. Em seguida, você pode selecionar um host de destino ou um grupo de recursos do host para iniciar instâncias, mas isso não é necessário se seu host dedicado tiver o posicionamento automático habilitado. Para obter mais informações, consulte Posicionamento automático no Guia do EC2 usuário da Amazon.
    - ID do host de locação: o ID do host dedicado no qual as instâncias são executadas.
    - Grupo de recursos do host de locação: o nome do recurso da Amazon (ARN) do grupo de recursos do host no qual as instâncias serão executadas.

6. Na seção Tags de infraestrutura (opcional), você pode atribuir tags de metadados à EC2 instância da Amazon que o Image Builder executa durante o processo de criação. As tags são inseridas como pares de chave-valor.
7. Na seção Tags (opcional), você pode atribuir tags de metadados ao recurso de configuração de infraestrutura que o Image Builder cria como saída. As tags são inseridas como pares de chave-valor.

AWS CLI

O procedimento a seguir mostra como configurar a infraestrutura da sua imagem com o comando create-infrastructure-configuration do Image Builder na AWS CLI. O comando na Etapa 2 incorpora o arquivo que você criou na Etapa 1. Para esses exemplos, o arquivo da Etapa 1 é chamado `create-infrastructure-configuration.json`.

1. Criar um arquivo JSON de entrada da CLI

   Os exemplos a seguir mostram variações do arquivo JSON que você pode criar para a configuração da infraestrutura. Use uma ferramenta de edição de arquivos para criar seu próprio arquivo JSON.

   Exemplo 1: configuração para reter uma instância de uma compilação com falha

   Este exemplo especifica dois tipos de instância: `m5.large` e `m5.xlarge`. Recomendamos que você especifique mais de um tipo de instância, pois isso permite que o Image Builder execute uma instância a partir de um grupo com capacidade suficiente. Isto pode reduzir suas falhas transitórias de compilação.

   O `instanceProfileName` especifica o perfil de instância que fornece à instância as permissões que o perfil exige para realizar atividades de personalização. Por exemplo, se você tem um componente que recupera recursos do Amazon S3, o perfil de instância exige permissões para acessar esses arquivos. O perfil da instância também exige um conjunto mínimo de permissões para que o EC2 Image Builder se comunique com sucesso com a instância. Para obter mais informações, consulte Configurar compilação de imagens personalizadas com o 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"
}
```

Exemplo 2: configuração do macOS com posicionamento automático

Este exemplo especifica os tipos de instância e o posicionamento de uma instância do Mac em que o host dedicado tem o posicionamento automático habilitado.

```
{
   "name": "macOSInfraConfigAutoPlacement",
   "description": "An example infrastructure configuration for macOS.",
   "instanceProfileName": "EC2InstanceProfileForImageBuilder",
   "instanceTypes": ["mac1.metal, mac2.metal"],
   "terminateInstanceOnFailure": false,
   "placement": {
      "tenancy": "host"
   }
}
```

Exemplo 3: configuração do macOS com o ID do host especificado

Este exemplo especifica o tipo de instância e o posicionamento de uma instância do Mac que tem como destino um host dedicado específico.

```
{
   "name": "macOSInfraConfigHostPlacement",
   "description": "An example infrastructure configuration for macOS.",
```

```
   "instanceProfileName": "EC2InstanceProfileForImageBuilder",
   "instanceTypes": ["mac2-m1ultra.metal"],
   "terminateInstanceOnFailure": false,
   "placement": {
      "tenancy": "host",
      "hostId" : "h-1234567890abcdef0"
   }
}
```

2. Use o arquivo que você criou como entrada quando executar o seguinte comando.

```
aws imagebuilder create-infrastructure-configuration --cli-input-json
 file://create-infrastructure-configuration.json
```

Atualizar uma configuração de infraestrutura

Esta seção aborda como você pode usar o console do Image Builder ou imagebuilder os comandos no AWS CLI para atualizar um recurso de configuração de infraestrutura. Para monitorar seus recursos, é possível aplicar tags da maneira a seguir. As tags são inseridas como pares de chave-valor.

- As tags de recursos atribuem tags de metadados à EC2 instância da Amazon que o Image Builder executa durante o processo de criação.
- Em Tags, atribua tags de metadados ao recurso de configuração de infraestrutura que o Image Builder cria como saída.

Console

Você pode editar os seguintes detalhes de configuração da infraestrutura no console do Image Builder:

- A descrição da configuração da infraestrutura.
- Associe o perfil do IAM para associar ao perfil de instância.
- AWS infraestrutura, incluindo o tipo de instância e um tópico de SNS para notificações.
- VPC, sub-rede e grupos de segurança.
- Configurações de solução de problemas, incluindo Encerrar instância em caso de falha, o par de chaves para conexão e um local opcional do bucket S3 para os logs da instância.

Para atualizar um recurso de configuração de infraestrutura do console do Image Builder, siga estas etapas:

Escolha uma configuração de infraestrutura existente do Image Builder

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Para ver uma lista dos recursos de configuração de infraestrutura em sua conta, escolha Configuração de infraestrutura no painel de navegação.
3. Para ver detalhes ou editar uma configuração de infraestrutura, escolha o link Nome da configuração. Isto abre a visualização detalhada da configuração da infraestrutura.

> **Note**
>
> Você também pode selecionar a caixa ao lado do nome da configuração e, em seguida, escolher Exibir detalhes.

4. No canto superior direito do painel Detalhes da infraestrutura, escolha Editar.
5. Quando estiver pronto para salvar as atualizações feitas na configuração da sua infraestrutura, escolha Salvar alterações .

AWS CLI

O exemplo a seguir mostra como atualizar a configuração da infraestrutura da sua imagem com o comando update-infrastructure-configuration do Image Builder na AWS CLI.

1. Criar um arquivo JSON de entrada da CLI

   Este exemplo de configuração de infraestrutura usa as mesmas configurações do exemplo de criação, exceto que atualizamos a configuração `terminateInstanceOnFailure` para`false`. Depois de executarmos o comando update-infrastructure-configuration, os pipelines que usam esta configuração de infraestrutura encerram as instâncias de compilação e teste quando a compilação falha.

   Use uma ferramenta de edição de arquivos para criar um arquivo JSON com as chaves mostradas no exemplo a seguir, além de valores válidos para seu ambiente. Este exemplo usa um arquivo denominado `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. Use o arquivo que você criou como entrada quando executar o seguinte comando.

```
aws imagebuilder update-infrastructure-configuration --cli-input-json
 file://update-infrastructure-configuration.json
```

Image Builder e AWS PrivateLink interface VPC endpoints

Você pode estabelecer uma conexão privada entre sua VPC e o EC2 Image Builder criando uma interface VPC endpoint. Os endpoints de interface são alimentados por AWS PrivateLinkuma tecnologia que permite acessar de forma privada o Image Builder APIs sem um gateway de internet, dispositivo NAT, conexão VPN ou conexão.Direct Connect As instâncias em sua VPC não precisam de endereços IP públicos para se comunicar com o Image Builder. APIs O tráfego entre sua VPC e o Image Builder não deixa a rede da Amazon.

Cada endpoint de interface é representado por uma ou mais Interfaces de Rede Elástica nas sub-redes. Ao criar uma nova imagem, você pode especificar o ID de sub-rede da VPC na configuração da sua infraestrutura.

> **Note**
>
> Cada serviço que você acessa de dentro de uma VPC tem seu próprio endpoint de interface, com sua própria política de endpoint. O Image Builder baixa o aplicativo gerenciador de AWSTOE componentes e acessa os recursos gerenciados dos buckets do S3 para criar imagens personalizadas. Para conceder acesso a esses buckets, você deve atualizar a política de endpoint do S3 para permitir isso. Para obter mais informações, consulte Políticas personalizadas para acesso ao bucket do S3.

Para obter mais informações sobre endpoints da VPC, consulte endpoints da VPC de interface (AWS PrivateLink) no Guia do usuário da Amazon VPC.

Considerações sobre endpoints VPC do Image Builder

Antes de configurar um endpoint da VPC de interface para o Image Builder, certifique-se de revisar as Propriedades e limitações do endpoint de interface no Guia do usuário da Amazon VPC.

O Image Builder oferece suporte a chamadas para todas as ações de API da sua VPC.

Criar um endpoint da VPC de interface para o Image Builder

Para criar um VPC endpoint para o serviço Image Builder, você pode usar o console Amazon VPC ou o ().AWS Command Line InterfaceAWS CLI Para obter mais informações, consulte Criar um endpoint de interface no Guia do usuário da Amazon VPC.

Crie um endpoint da VPC para o Image Builder usando o seguinte nome de serviço:

- com.amazonaws. *region*.construtor de imagens

Se você habilitar o DNS privado para o endpoint, poderá fazer solicitações de API para o Image Builder usando seu nome DNS padrão para a região, por exemplo, `imagebuilder.us-east-1.amazonaws.com`. Para pesquisar o endpoint que se aplica à sua região de destino, consulte endpoints e cotas do EC2 Image Builder no. Referência geral da Amazon Web Services

Para mais informações, consulte Acessar um serviço por um endpoint de interface no Guia do usuário da Amazon VPC.

Criar uma política de endpoint da VPC o Image Builder

É possível anexar uma política de endpoint ao endpoint da VPC que controla o acesso ao Image. Essa política especifica as seguintes informações:

- A entidade principal que pode realizar ações.
- As ações que podem ser realizadas.
- Os recursos aos quais as ações podem ser aplicadas.

Se você estiver usando componentes gerenciados pela Amazon em sua fórmula, o endpoint da VPC para Image Builder deve permitir acesso à seguinte biblioteca de componentes de propriedade do serviço:

`arn:aws:imagebuilder:region:aws:component/*`

> **Important**
>
> Quando uma política não padrão é aplicada a um endpoint VPC de interface para Image EC2 Builder, certas solicitações de API com falha, como aquelas que falharam, podem não ser `RequestLimitExceeded` registradas na Amazon.AWS CloudTrail CloudWatch

Para mais informações, consulte Controlar o acesso a serviços com VPC endpoints no Guia do usuário da Amazon VPC.

Políticas personalizadas para acesso ao bucket do S3

O Image Builder usa um bucket do S3 disponível publicamente para armazenar e acessar recursos gerenciados, como componentes. Ele também baixa o aplicativo de gerenciamento de AWSTOE componentes de um bucket S3 separado. Se estiver usando um endpoint da VPC para o Amazon S3 em seu ambiente, será necessário garantir que sua política de endpoint da VPC do S3 permita que o construtor de imagens acesse os seguintes buckets do S3. Os nomes dos buckets são exclusivos por AWS região (*region*) e pelo ambiente do aplicativo (*environment*). Image Builder e dêAWSTOE suporte aos seguintes ambientes de aplicativos: `prod`, `preprod`, `beta` e.

- O bucket AWSTOE do gerenciador de componentes:

```
s3://ec2imagebuilder-toe-region-environment
```

Exemplo: s3://ec2 imagebuilder-toe-us-west -2-prod/*

- O bucket de recursos gerenciados do Image Builder:

```
s3://ec2imagebuilder-managed-resources-region-environment/components
```

Exemplo: s3://ec2 imagebuilder-managed-resources-us -west-2-prod/components/*

Exemplos de política de endpoint da VPC

Esta seção inclui exemplos de políticas personalizadas de endpoint da VPC.

Política geral de endpoint da VPC para ações do Image Builder

O exemplo de política de endpoint a seguir para o Image Builder nega permissão para excluir imagens e componentes do Image Builder. O exemplo de política também concede permissão para realizar todas as outras ações do EC2 Image Builder.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": "imagebuilder:*",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "imagebuilder:DeleteImage",
            "Effect": "Deny",
            "Resource": "*"
        },
        {
            "Action": "imagebuilder:DeleteComponent",
            "Effect": "Deny",
            "Resource": "*"
        }
    ]
}
```

Restrinja o acesso por organização, permita o acesso gerenciado aos componentes

O exemplo de política de endpoint a seguir mostra como restringir o acesso a identidades e recursos que pertencem à sua organização e conceder acesso aos componentes do Image Builder gerenciados pela Amazon. Substitua *regionprincipal-org-id*, e *resource-org-id* pelos valores da sua organização.

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:us-east-1:aws:component/*"
            ]
        }
    ]
}
```

Política de endpoint da VPC para acesso ao bucket do Amazon S3

O exemplo de política de endpoint do S3 a seguir mostra como fornecer acesso aos buckets do S3 que o Image Builder usa para criar imagens personalizadas. *environment*Substitua *region* e pelos valores da sua organização. Adicione outras permissões necessárias à política com base nos requisitos do seu aplicativo.

> **Note**
>
> Com relação a imagens do Linux, se você não especificar dados do usuário em sua fórmula de imagem, o Image Builder adicionará um script para baixar e instalar o Systems Manager Agent nas instâncias de compilação e teste da sua imagem. Para baixar o agente, o Image Builder acessa o bucket do S3 referente à sua região de compilação.
> Para garantir que o Image Builder possa inicializar as instâncias de compilação e teste, adicione o seguinte recurso complementar à sua política de endpoint do S3: `"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/*"
      ]
    }
  ]
}
```

Gerenciar as configurações de distribuição do Image Builder

Antes de definir as configurações de distribuição de imagens de saída, recomendamos que você verifique a disponibilidade de qualquer infraestrutura subjacente ou outros requisitos para instâncias que são executadas na imagem de saída nas regiões de destino da distribuição. Por exemplo, nem todas as regiões oferecem suporte a hosts dedicados do EC2 Mac, que são necessários para iniciar instâncias a partir de uma imagem do macOS. Para obter mais informações sobre tipos de instância e preços para hosts dedicados, consulte Preços de hosts EC2 dedicados da Amazon.

Depois de criar as configurações de distribuição com o Image Builder, você pode gerenciá-las usando o console do Image Builder, a API do Image Builder ou comandos imagebuilder na AWS CLI. Com as configurações de distribuição, você pode executar as seguintes ações:

Distribuição da AMI

- Especifique o nome e a descrição da sua AMI de saída.
- Autorize outras Contas da AWS organizações e a iniciar OUs a AMI a partir da conta do proprietário. A conta do proprietário é cobrada pelas cobranças que são associadas à AMI.

> **Note**
>
> Para tornar uma AMI pública, defina as contas autorizadas com permissão de execução como `all`. Para obter informações e exemplos, consulte ModifyImageAttribute a Amazon EC2 API Reference.

- Crie uma cópia da AMI de saída para cada uma das contas, organizações e organizações de destino especificadas OUs na região de destino. As contas, organizações e organizações de destino OUs possuem suas cópias da AMI e são cobradas por quaisquer cobranças associadas. Para obter mais informações sobre como distribuir sua AMI para AWS Organizations e OUs, consulte Compartilhar uma AMI com organizações ou OUs.
- Copie a AMI para a conta do proprietário em outra Regiões da AWS.
- Exporte discos de imagem de VM para o Amazon Simple Storage Service (Amazon S3). Para obter mais informações, consulte Exemplo: Crie configurações de distribuição para discos VM de saída do AWS CLI.

Distribuição de imagens de contêiner

- Especifique o repositório ECR em que o Image Builder armazena a imagem de saída na região de distribuição.

Você pode usar suas configurações de distribuição das seguintes formas para entregar imagens às regiões, contas AWS Organizations e unidades organizacionais (OUs) de destino uma vez ou com cada criação de funil:

- Para entregar automaticamente imagens atualizadas para regiões, contas, Organizations especificadas e OUs use as configurações de distribuição com um pipeline do Image Builder que é executado de acordo com um cronograma.
- Para criar uma nova imagem e entregá-la às regiões, contas e organizações especificadas OUs, use as configurações de distribuição com um pipeline do Image Builder que você executa uma vez no console do Image Builder, usando Run pipeline no menu Actions.
- Para criar uma nova imagem e entregá-la às regiões, contas, organizações e organizações especificadas OUs, use as configurações de distribuição com a seguinte ação de API ou comando Image Builder no AWS CLI:
  - A ação CreateImage na API do Image Builder.
  - O comando create-image na AWS CLI.
- Para exportar discos de imagem de máquina virtual (VM) para buckets do S3 nas regiões de destino como parte do seu processo normal de compilação de imagens.

> **Tip**
>
> Quando você tem vários recursos do mesmo tipo, a marcação com tags ajuda a identificar um recurso específico com base nas tags que você atribuiu a ele. Para obter mais informações sobre como marcar seus recursos usando os comandos do Image Builder no AWS CLI, consulte a Marcar recursos seção deste guia.

Conteúdo

- Listar e visualizar detalhes de uma configuração de distribuição
- Criar e atualizar as configurações de distribuição da AMI
- Criar e atualizar configurações de distribuição para imagens de contêiner

- Configurar a distribuição da AMI entre contas com o Image Builder
- Configurar a distribuição da AMI com um modelo de EC2 lançamento
- Use recursos aprimorados de distribuição de AMI

Listar e visualizar detalhes de uma configuração de distribuição

Esta seção descreve as várias maneiras pelas quais você pode encontrar informações e visualizar detalhes da configuração de distribuição do EC2 Image Builder.

Detalhes da configuração de distribuição

- Listar configurações de distribuição com o console
- Visualizar detalhes da configuração de distribuição com o console
- Listar distribuições do AWS CLI
- Obtenha detalhes da configuração de distribuição no AWS CLI

Listar configurações de distribuição com o console

Para ver uma lista das configurações de distribuição que foram criadas em sua conta no console do Image Builder, siga estas etapas:

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Escolha Configurações de distribuição no painel de navegação. Isso mostra uma lista das configurações de distribuição criadas em sua conta.
3. Para ver detalhes ou criar uma nova configuração de distribuição, escolha o link Nome da configuração. Isso abre a visualização detalhada das configurações de distribuição.

> **Note**
>
> Você também pode selecionar a caixa ao lado do nome da configuração e, em seguida, escolher Exibir detalhes.

Visualizar detalhes da configuração de distribuição com o console

Para ver detalhes de uma configuração de distribuição específica usando o console do Image Builder, selecione a configuração a ser revisada e use as etapas descritas em Listar configurações de distribuição com o console.

Na página de detalhes da distribuição, você pode:

- Excluir a configuração de distribuição. Para obter mais informações sobre exclusão de recursos no Image Builder, consulte Excluir recursos desatualizados ou não utilizados do Image Builder.
- Editar detalhes de distribuição.

Listar distribuições do AWS CLI

O exemplo a seguir mostra como usar o list-distribution-configurations comando no AWS CLI para listar todas as suas distribuições.

```
aws imagebuilder list-distribution-configurations
```

Obtenha detalhes da configuração de distribuição no AWS CLI

O exemplo a seguir mostra como usar o get-distribution-configuration comando no AWS CLI para obter os detalhes de uma configuração de distribuição especificando seu 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
```

Criar e atualizar as configurações de distribuição da AMI

Esta seção aborda a criação e a atualização de configurações de distribuição para uma AMI do Image Builder.

Conteúdo

- Pré-requisitos de distribuição da AMI
- Criar uma configuração de distribuição da AMI

- Atualizar uma configuração de distribuição da AMI
- Exemplo: habilitar o EC2 Fast Launch com um modelo de lançamento para saída AMIs
- Exemplo: Crie configurações de distribuição para discos VM de saída do AWS CLI

Pré-requisitos de distribuição da AMI

Algumas configurações de distribuição têm pré-requisitos, como segue:

Tópicos

- Pré-requisitos para parâmetros de saída SSM
- Pré-requisitos para o lançamento rápido EC2

Pré-requisitos para parâmetros de saída SSM

Antes de criar uma nova configuração de distribuição da AMI que defina um AWS Systems Manager parâmetro do Parameter Store (parâmetro SSM), verifique se você atendeu aos seguintes pré-requisitos.

Perfil de execução

Ao criar um pipeline ou usar o comando create-image no AWS CLI, você só pode especificar uma função de execução do Image Builder. Se você tiver definido uma função de execução do fluxo de trabalho do Image Builder, você adicionaria quaisquer permissões de recursos adicionais a essa função. Caso contrário, você criaria um novo papel personalizado que inclua as permissões necessárias.

- Para armazenar o ID da AMI de saída em um parâmetro SSM durante a distribuição, você deve especificar a `ssm:PutParameter` ação na função de execução do Image Builder, com o parâmetro listado como um recurso.
- Ao definir o tipo de dados do parâmetro para sinalizar `AWS EC2 Image` ao Systems Manager que valide o valor do parâmetro como uma ID de AMI, você também deve adicionar a `ec2:DescribeImages` ação.

Pré-requisitos para o lançamento rápido EC2

Antes de criar uma nova configuração de distribuição para o EC2 Fast Launch for Windows AMIs, verifique se você atendeu aos seguintes pré-requisitos.

- Se você fornecer um modelo de execução personalizado ao configurar o EC2 Fast Launch, o serviço usará a VPC e outras configurações que você definiu no modelo de execução. Para obter mais informações, consulte Usar um modelo de lançamento ao configurar o EC2 Fast Launch.
- Se você não usar um modelo de lançamento personalizado para definir suas configurações, deverá anexar a EC2FastLaunchFullAccesspolítica à função do IAM que o Image Builder usa para criar sua imagem. Ao criar um pipeline ou usar o comando create-image no AWS CLI, você só pode especificar uma função de execução do Image Builder. Se você tiver definido uma função de execução do fluxo de trabalho do Image Builder, você adicionaria quaisquer permissões de recursos adicionais a essa função. Caso contrário, você criaria um novo papel personalizado que inclua as permissões necessárias.

  Então, quando o Image Builder copia sua imagem, o EC2 Fast Launch cria automaticamente uma CloudFormation pilha com os seguintes recursos em sua Conta da AWS.

  - Uma nuvem privada virtual (VPC).
  - Sub-redes privadas em várias zonas de disponibilidade
  - Um modelo de lançamento configurado com o Instance Metadata Service versão 2 () IMDSv2
  - Um grupo de segurança sem regras de entrada e saída

> **Note**
>
> O Image Builder não oferece suporte à distribuição entre contas AMIs com o EC2 Fast Launch pré-ativado. EC2 O Fast Launch deve ser ativado na conta de destino.

Criar uma configuração de distribuição da AMI

As configurações de distribuição incluem o nome da AMI de saída, configurações de região específicas para criptografia, permissões de lançamento e Contas da AWS organizações e unidades organizacionais (OUs) que podem iniciar a AMI de saída e configurações de licença.

Uma configuração de distribuição permite que você especifique o nome e a descrição da sua AMI de saída, autorize outras pessoas Contas da AWS a iniciar a AMI, copie a AMI para outras contas e replique a AMI para outras AWS regiões. Também permite exportar a AMI para o Amazon Simple Storage Service (Amazon S3) ou EC2 configurar o Fast Launch para o Windows de saída. AMIs Para tornar uma AMI pública, defina as contas autorizadas com permissão de execução como `all`. Veja os exemplos de como tornar uma AMI pública em EC2 ModifyImageAttribute.

Console

Siga estas etapas para criar uma nova configuração de distribuição da AMI no Console de gerenciamento da AWS:

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Escolha Configurações de distribuição no painel de navegação. Isso mostra uma lista das configurações de distribuição criadas em sua conta.
3. Escolha Criar configurações de distribuição na parte superior do painel Configurações de distribuição.
4. Na seção Tipo de imagem, escolha imagem de máquina da Amazon (AMI) como o tipo de saída.
5. Na seção Geral, insira um Nome para sua configuração de distribuição e uma descrição opcional.
6. Na seção Configurações da região, insira os seguintes detalhes para cada região em que você está distribuindo sua AMI:

    a. A AMI é distribuída para a região atual (Região 1), por padrão. A Região 1 é a fonte da distribuição. Algumas configurações da Região 1 não estão abertas para edição. Para qualquer região que você adicionar, você pode escolher uma região na lista suspensa Região.

    A chave Kms identifica o AWS KMS key que é usado para criptografar os volumes do EBS para sua imagem na região de destino. É importante observar que isso não se aplica à AMI original que a compilação cria em sua conta na região de origem (Região 1). A criptografia executada durante a fase de distribuição da compilação é somente para imagens que são distribuídas para outras contas ou regiões.

    Para criptografar os volumes do EBS para a AMI criada na região de origem da sua conta, você deve definir a chave KMS no mapeamento de dispositivos de blocos de fórmula de imagem (Armazenamento (volumes) no console).

    O Image Builder copia a AMI para as contas de destino que você especifica para a região.

> **Pré-requisito**
>
> Para copiar uma imagem entre contas, você deve criar o perfil `EC2ImageBuilderDistributionCrossAccountRole` em todas as contas de distribuição de destino e anexar a política gerenciada Política Ec2ImageBuilderCrossAccountDistributionAccess ao perfil.

O nome da AMI de saída é opcional. Se você fornecer um nome, o nome final da AMI de saída incluirá um timestamp anexado de quando a AMI foi criada. Se você não especificar um nome, o Image Builder acrescentará o timestamp de compilação ao nome da fórmula. Isso garante nomes exclusivos de AMI para cada compilação.

i. Com o compartilhamento da AMI, você pode conceder acesso a AWS diretores específicos para executar instâncias da sua AMI. Se você expandir a seção de compartilhamento da AMI, poderá inserir os seguintes detalhes:

   - Permissões de execução — selecione Privada se quiser manter sua AMI privada e permitir o acesso de AWS diretores específicos para iniciar uma instância a partir da sua AMI privada. Selecione Público se quiser tornar sua AMI pública. Qualquer AWS diretor pode iniciar uma instância a partir da sua AMI pública.
   - Diretores — Você pode conceder acesso aos seguintes tipos de AWS Diretores para iniciar instâncias:
     - AWS conta — Conceda acesso a uma AWS conta específica
     - Unidade organizacional (OU): conceda acesso a uma OU e a todas as suas entidades secundárias. As entidades OUs secundárias incluem AWS contas.
     - Organização — conceda acesso à sua AWS Organizations entidade e a todas as suas entidades secundárias. As entidades OUs secundárias incluem AWS contas.

       Primeiro, selecione o tipo de entidade principal. Em seguida, insira o ID da entidade principal da AWS a qual deseja conceder acesso na caixa à direita da lista suspensa. Você pode inserir vários IDs tipos diferentes.

ii. Você pode expandir a seção Configuração de licença para anexar as configurações de licença criadas com AWS License Manageràs suas imagens do Image Builder. As configurações de licença contêm regras de licenciamento com base nos termos

de seus contratos empresariais. O Image Builder inclui automaticamente as configurações de licença associadas à sua AMI básica.

iii. Você pode expandir a seção Configuração do modelo de inicialização para especificar um modelo de EC2 execução a ser usado para iniciar instâncias da AMI que você criou.

Se você estiver usando um modelo de EC2 execução, poderá instruir o Image Builder a criar uma nova versão do seu modelo de execução que inclua a ID de AMI mais recente após a conclusão da compilação. Para atualizar o modelo de lançamento, defina as configurações da seguinte forma:

- Nome do modelo de inicialização: selecione o nome do modelo de inicialização que você deseja que o Image Builder atualize.
- Definir a versão padrão: marque essa caixa de seleção para atualizar a versão padrão do modelo de execução para a nova versão.

Para adicionar outra configuração de modelo de inicialização, escolha Adicionar configuração de modelo de inicialização. Você pode ter até cinco configurações de modelo de inicialização por região.

iv. Você pode expandir a seção de configurações de parâmetros do SSM para configurar um parâmetro do SSM que armazenará o ID da AMI de saída da imagem que é distribuída para a região de destino. Opcionalmente, você pode especificar uma conta de distribuição na Região.

Nome do parâmetro — insira o nome do seu parâmetro. Por exemplo, `/output/image/param`.

Tipo de dados — Mantenha o valor padrão (`AWS EC2 Image`). Isso faz com que o Systems Manager valide o valor do parâmetro para garantir que seja um ID de AMI válido.

b. Para adicionar configurações de distribuição para outra região, escolha Adicionar região.

7. Quando concluir, escolha Criar configurações.

AWS CLI

O exemplo a seguir mostra como usar o comando create-distribution-configuration para criar uma nova configuração de distribuição para AMI, usando a AWS CLI.

1. Criar um arquivo JSON de entrada da CLI

   Use uma ferramenta de edição de arquivos para criar um arquivo JSON com as chaves mostradas em um dos exemplos a seguir e valores válidos para seu ambiente. Esses exemplos definem quais Contas da AWS unidades organizacionais (OUs) têm permissão para iniciar a AMI que você distribui para as regiões especificadas.AWS Organizations Nomeie o arquivo `create-ami-distribution-configuration.json` para uso na próxima etapa:

   Exemplo 1: Distribuir para Contas da AWS

   Este exemplo distribui uma AMI para duas regiões e especifica Contas da AWS que têm permissões de execução em cada região.

```
{
 "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"
     ]
    }
   }
  }
 ]
}
```

Exemplo 2: Distribute to Organizations e OUs

Este exemplo distribui uma AMI para a região de origem e especifica as permissões de lançamento da organização e da 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"
     ]
    }
   }
  }
 ]
}
```

Exemplo 3: armazenar o ID da AMI de saída em um parâmetro SSM

Este exemplo armazena o ID da AMI de saída em um AWS Systems Manager parâmetro do Parameter Store na região de distribuição.

```
{
```

```
 "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. Execute o seguinte comando, usando o arquivo que você criou como entrada.

```
aws imagebuilder create-distribution-configuration --cli-input-json
 file://create-ami-distribution-configuration.json
```

> **Note**
>
> - É necessário incluir a notação `file://` no início do caminho do arquivo JSON.
> - O caminho para o arquivo JSON deve seguir a convenção apropriada para o sistema operacional de base no qual você está executando o comando. Por exemplo, o Windows usa a barra invertida (\) para se referir ao caminho do diretório, enquanto o Linux e o macOS usam a barra (/).

Para obter mais informações detalhadas, consulte create-distribution-configuration na Referência de comandos da AWS CLI.

Atualizar uma configuração de distribuição da AMI

Você pode alterar a configuração de distribuição da AMI. No entanto, as alterações feitas não se aplicam a nenhum recurso que o Image Builder já tenha distribuído. Por exemplo, se você distribuiu uma AMI para uma região que posteriormente removeu da sua distribuição, a AMI que já foi distribuída permanece nessa região até que você a remova manualmente.

Console de gerenciamento da AWS

Siga estas etapas para obter uma configuração de distribuição da AMI no Console de gerenciamento da AWS:

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Escolha Configurações de distribuição no painel de navegação. Isso mostra uma lista das configurações de distribuição criadas em sua conta.
3. Para ver detalhes ou atualizar uma configuração de distribuição, escolha o link Nome da configuração. Isso abre a visualização detalhada das configurações de distribuição.

   > **Note**
   >
   > Você também pode selecionar a caixa ao lado do nome da configuração e, em seguida, escolher Exibir detalhes.

4. Para editar a configuração da distribuição, escolha Editar no canto superior direito da seção Detalhes da distribuição. Alguns campos estão bloqueados, como o Nome da configuração de distribuição e a Região padrão que é exibida como Região 1. Para obter mais informações sobre as configurações de distribuição, consulte Criar uma configuração de distribuição da AMI.
5. Quando concluir, escolha Salvar alterações.

AWS CLI

O exemplo a seguir mostra como usar o comando update-distribution-configuration para atualizar as configurações de distribuição da AMI, usando a AWS CLI.

1. Criar um arquivo JSON de entrada da CLI

   Use uma ferramenta de edição de arquivos para criar um arquivo JSON com as chaves mostradas no exemplo a seguir e valores válidos para seu ambiente. Este exemplo usa um arquivo denominado `update-ami-distribution-configuration.json`.

```
{
 "distributionConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/update-ami-distribution-
configuration.json",
 "description": "Copies AMI to eu-west-2, and specifies accounts that can launch
 instances in each Region.",
 "distributions": [
   {
   "region": "us-west-2",
   "amiDistributionConfiguration": {
    "name": "Name {{imagebuilder:buildDate}}",
    "description": "An example image name with parameter references",
    "launchPermissions": {
     "userIds": [
      "987654321012"
     ]
    }
   }
  },
  {
   "region": "eu-west-2",
   "amiDistributionConfiguration": {
    "name": "My {{imagebuilder:buildVersion}} image {{imagebuilder:buildDate}}",
    "tags": {
     "KeyName": "Some value"
    },
    "launchPermissions": {
     "userIds": [
      "100000000001"
     ]
    }
   }
  }
 ]
}
```

2. Execute o seguinte comando, usando o arquivo que você criou como entrada.

```
aws imagebuilder update-distribution-configuration --cli-input-json
 file://update-ami-distribution-configuration.json
```

> **Note**
>
> - É necessário incluir a notação `file://` no início do caminho do arquivo JSON.
> - O caminho para o arquivo JSON deve seguir a convenção apropriada para o sistema operacional de base no qual você está executando o comando. Por exemplo, o Windows usa a barra invertida (\) para se referir ao caminho do diretório, enquanto o Linux e o macOS usam a barra (/).

Para obter mais informações detalhadas, consulte update-distribution-configuration na Referência de comandos da AWS CLI. Para atualizar as tags do seu recurso de configuração de distribuição, consulte a seção Marcar recursos.

Exemplo: habilitar o EC2 Fast Launch com um modelo de lançamento para saída AMIs

O exemplo a seguir mostra como usar o create-distribution-configuration comando com um modelo de execução para criar configurações de distribuição que tenham o EC2 Fast Launch configurado para sua AMI, a partir do AWS CLI.

Para definir as configurações do EC2 Fast Launch sem um modelo de lançamento, certifique-se de ter cumprido todos os EC2 Pré-requisitos do Fast Launch requisitos antes de criar sua configuração de distribuição.

1. Criar um arquivo JSON de entrada da CLI

   Use uma ferramenta de edição de arquivos para criar um arquivo JSON com as chaves como mostrado no exemplo a seguir, além de valores válidos para seu ambiente.

   Este exemplo executa instâncias para todos os seus recursos de destino simultaneamente, porque o número máximo de inicializações paralelas é maior que a contagem de recursos de destino. Esse arquivo é nomeado `ami-dist-config-win-fast-launch.json` no exemplo de comando mostrado na próxima etapa.

```
{
"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**
>
> Você pode especificar o `launchTemplateName` em vez do `launchTemplateId` na seção `launchTemplate`, mas não pode especificar o nome e o ID.

2. Execute o seguinte comando, usando o arquivo que você criou como entrada.

```
aws imagebuilder create-distribution-configuration --cli-input-json file://ami-dist-config-win-fast-launch.json
```

> **Note**
>
> - É necessário incluir a notação `file://` no início do caminho do arquivo JSON.
> - O caminho para o arquivo JSON deve seguir a convenção apropriada para o sistema operacional de base no qual você está executando o comando. Por exemplo, o Windows usa a barra invertida (\) para se referir ao caminho do diretório, enquanto o Linux e o macOS usam a barra (/).

Para obter mais informações detalhadas, consulte create-distribution-configuration na Referência de comandos da AWS CLI.

Exemplo: Crie configurações de distribuição para discos VM de saída do AWS CLI

O exemplo a seguir mostra como usar o comando create-distribution-configuration para criar configurações de distribuição que exportarão discos de imagem da VM para o Amazon S3 a cada compilação de imagem.

1. Criar um arquivo JSON de entrada da CLI

   Você pode simplificar o comando create-distribution-configuration que você usa na AWS CLI. Para fazer isso, crie um arquivo JSON que contenha toda a configuração de exportação que você deseja passar para o comando.

   > **Note**
   >
   > A convenção de nomenclatura para os valores de dados no arquivo JSON segue o padrão especificado para os parâmetros de solicitação de operação da API Image Builder. Para revisar os parâmetros da solicitação de operação da API, consulte o CreateDistributionConfiguration comando na Referência da API EC2 Image Builder.

Para fornecer os valores dos dados como parâmetros da linha de comando, consulte os nomes dos parâmetros especificados na Referência de comando do AWS CLI para o comando create-distribution-configuration como opções.

Aqui está um resumo dos parâmetros que especificamos no objeto JSON `s3ExportConfiguration` deste exemplo:

- roleName (string, required) — O nome da função que concede Import/Export permissão à VM para exportar imagens para seu bucket do S3.
- diskImageFormat(string, obrigatório) — Exporte a imagem de disco atualizada para um dos seguintes formatos compatíveis:
  - Virtual Hard Disk (VHD): compatível com os produtos de virtualização Citrix Xen e Microsoft Hyper-V.
  - Disco de máquina virtual ESX (VMDK) otimizado para streaming — compatível com as versões 4, 5 e 6 do ESX VMware e VMware vSphere.
  - Raw: formato bruto.
- S3bucket (string, obrigatório): o bucket S3 no qual armazenar as imagens de disco de saída para sua VM.

Salve o arquivo como `export-vm-disks.json`. Use o nome do arquivo no comando create-distribution-configuration.

```
{
 "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. Execute o seguinte comando, usando o arquivo que você criou como entrada.

```
aws imagebuilder create-distribution-configuration --cli-input-json file://export-vm-disks.json
```

> **Note**
>
> - É necessário incluir a notação `file://` no início do caminho do arquivo JSON.
> - O caminho para o arquivo JSON deve seguir a convenção apropriada para o sistema operacional de base no qual você está executando o comando. Por exemplo, o Windows usa a barra invertida (\) para se referir ao caminho do diretório, enquanto o Linux e o macOS usam a barra (/).

Para obter mais informações detalhadas, consulte create-distribution-configuration na Referência de comandos da AWS CLI.

Criar e atualizar configurações de distribuição para imagens de contêiner

Esta seção aborda a criação e a atualização de configurações de distribuição de imagens de contêiner do Image Builder.

Conteúdo

- Crie configurações de distribuição para imagens de contêiner do Image Builder a partir do AWS CLI
- Atualize as configurações de distribuição da imagem do seu contêiner a partir do AWS CLI

Crie configurações de distribuição para imagens de contêiner do Image Builder a partir do AWS CLI

Uma configuração de distribuição permite que você especifique o nome e a descrição da imagem do contêiner de saída e replique a imagem do contêiner para outras AWS regiões. Você também pode

aplicar tags separadas ao recurso de configuração de distribuição e às imagens do contêiner em cada região.

1. Criar um arquivo JSON de entrada da CLI

   Use a sua ferramenta de edição de arquivos favorita para criar um arquivo JSON com as chaves mostradas no exemplo a seguir, além de valores válidos para seu ambiente. Este exemplo usa um arquivo denominado `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. Execute o seguinte comando, usando o arquivo que você criou como entrada.

```
aws imagebuilder create-distribution-configuration --cli-input-json file://create-container-distribution-configuration.json
```

> **Note**
>
> - É necessário incluir a notação `file://` no início do caminho do arquivo JSON.
> - O caminho para o arquivo JSON deve seguir a convenção apropriada para o sistema operacional de base no qual você está executando o comando. Por exemplo, o Windows usa a barra invertida (\) para se referir ao caminho do diretório, enquanto o Linux e o macOS usam a barra (/).

Para obter mais informações detalhadas, consulte create-distribution-configuration na Referência de comandos da AWS CLI.

Atualize as configurações de distribuição da imagem do seu contêiner a partir do AWS CLI

O exemplo a seguir mostra como usar o comando update-distribution-configuration para atualizar as configurações de distribuição da imagem de contêiner usando a AWS CLI. Você também pode atualizar as tags das imagens de contêiner em cada região.

1. Criar um arquivo JSON de entrada da CLI

   Use a sua ferramenta de edição de arquivos favorita para criar um arquivo JSON com as chaves mostradas no exemplo a seguir, além de valores válidos para seu ambiente. Este exemplo usa um arquivo denominado `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. Execute o seguinte comando, usando o arquivo que você criou como entrada:

```
aws imagebuilder update-distribution-configuration --cli-input-json file://update-container-distribution-configuration.json
```

> **Note**
>
> - É necessário incluir a notação `file://` no início do caminho do arquivo JSON.
> - O caminho para o arquivo JSON deve seguir a convenção apropriada para o sistema operacional de base no qual você está executando o comando. Por exemplo, o Windows usa a barra invertida (\) para se referir ao caminho do diretório, enquanto o Linux e o macOS usam a barra (/).

Para obter mais informações detalhadas, consulte update-distribution-configuration na Referência de comandos da AWS CLI. Para atualizar as tags do seu recurso de configuração de distribuição, consulte a seção Marcar recursos.

Configurar a distribuição da AMI entre contas com o Image Builder

Esta seção descreve como você pode definir as configurações de distribuição para entregar uma AMI do Image Builder para outras contas que você especificar.

A conta de destino pode então inicializar ou modificar a AMI, conforme necessário.

> **Note**
>
> AWS CLI os exemplos de comando nesta seção pressupõem que você tenha criado anteriormente arquivos JSON de receita de imagem e configuração de infraestrutura. Para criar o arquivo JSON para uma fórmula de imagem, consulte Crie uma receita de imagem com o AWS CLI. Para criar o arquivo JSON para uma configuração de infraestrutura, consulte Criar uma configuração de infraestrutura.

Pré-requisitos para a distribuição da AMI entre contas

Para garantir que as contas de destino possam inicializar as instâncias com sucesso a partir da sua imagem do Image Builder, você deve configurar as permissões apropriadas para todas as contas de destino em todas as regiões.

Se você criptografar sua AMI usando AWS Key Management Service(AWS KMS), deverá configurar uma AWS KMS key para sua conta que seja usada para criptografar a nova imagem.

Quando o Image Builder realiza a distribuição entre contas para criptografia AMIs, a imagem na conta de origem é descriptografada e enviada para a região de destino, onde é recriptografada usando a chave designada para essa região. Como o Image Builder age em nome da conta de destino e usa um perfil do IAM que você cria na região de destino, essa conta deve ter acesso às chaves nas regiões de origem e de destino.

Chaves de criptografia

Os pré-requisitos a seguir são necessários se sua imagem for criptografada usando AWS KMS. Os pré-requisitos do IAM são abordados na próxima seção.

Requisitos da conta de origem

- Crie uma chave do KMS em sua conta em todas as regiões em que você compila e distribui sua AMI. Você também pode usar uma chave existente.
- Atualize a política de chave para todas essas chaves, para permitir que as contas de destino usem sua chave.

Requisitos da conta de destino

- Adicione uma política em linha para `EC2ImageBuilderDistributionCrossAccountRole` que permita que o perfil execute as ações necessárias para distribuir uma AMI criptografada. Para ver as etapas de configuração do IAM, consulte a seção de pré-requisitos Políticas do IAM.

Para obter mais informações sobre o uso do acesso entre contas AWS KMS, consulte Permitir que usuários em outras contas usem uma chave KMS no Guia do AWS Key Management Service desenvolvedor.

Especifique sua chave de criptografia na fórmula da imagem, da seguinte forma:

- Se você estiver usando o console do Image Builder, escolha sua chave de criptografia na lista suspensa Criptografia (alias KMS) na seção Armazenamento (volumes) da sua fórmula.
- Se você estiver usando a ação da CreateImageRecipe API ou o create-image-recipe comando no AWS CLI, configure sua chave na `ebs` seção abaixo `blockDeviceMappings` em sua entrada JSON.

  O snippet de JSON a seguir mostra as configurações de criptografia de uma fórmula de imagem. Além de fornecer sua chave de criptografia, você também deve definir o sinalizador `encrypted` como `true`.

```
{
 ...
 "blockDeviceMappings": [
 {
  "deviceName": "Example root volume",
```

```
  "ebs": {
   "deleteOnTermination": true,
   "encrypted": true,
   "iops": 100,
   "kmsKeyId": "image-owner-key-id",
   ...
  },
  ...
 }],
 ...
}
```

Políticas do IAM

Para configurar as permissões de distribuição entre contas no AWS Identity and Access Management(IAM), siga estas etapas:

1. Para usar o Image Builder AMIs distribuído entre contas, o proprietário da conta de destino deve criar uma nova função do IAM em sua conta chamada`EC2ImageBuilderDistributionCrossAccountRole`.
2. Ele deve anexar o Política Ec2ImageBuilderCrossAccountDistributionAccess ao perfil para permitir a distribuição entre contas. Para obter mais informações sobre políticas gerenciadas, consulte Políticas gerenciadas e políticas em linha no Guia do usuário do AWS Identity and Access Management.
3. Verifique se o ID da conta de origem foi adicionado à política de confiança anexada ao perfil do IAM da conta de destino. O exemplo a seguir mostra uma política de confiança na conta de destino que especifica o ID da conta de origem.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::444455556666:root"
        },
        "Action": "sts:AssumeRole"
    }]
```

```
}
```

Para obter mais informações sobre políticas de confiança, consulte Políticas Baseadas em Recursos no Guia do Usuário do AWS Identity and Access Management.

4. Se a AMI que você distribui for criptografada, o proprietário da conta de destino deverá adicionar a seguinte política em linha ao `EC2ImageBuilderDistributionCrossAccountRole` na conta dele para que ele possa usar suas chaves KMS. A seção `Principal` contém o número da conta dele. Isso permite que o Image Builder aja em seu nome quando usado AWS KMS para criptografar e descriptografar a AMI com as chaves apropriadas para cada região.

   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": "*"
     }
    ]
   }
   ```

   Para obter mais informações sobre políticas em linha, consulte Políticas em linha no Guia do usuário do AWS Identity and Access Management.

5. Se você estiver usando `launchTemplateConfigurations` para especificar um modelo de EC2 lançamento da Amazon, você também deve adicionar a seguinte política à sua conta `EC2ImageBuilderDistributionCrossAccountRole` em cada destino.

JSON

```
{
    "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. Se você usar um AWS Systems Manager parâmetro do Parameter Store para armazenar o ID da AMI de saída para a conta de distribuição e a região, deverá adicionar a seguinte política à sua `EC2ImageBuilderDistributionCrossAccountRole` em cada conta de destino.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:*:111122223333:parameter/ImageBuilder-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeImages"
            ],
            "Resource": "*"
        }
    ]
}
```

Limites para a distribuição entre contas

Há algumas limitações ao distribuir imagens do Image Builder entre contas:

- A conta de destino está limitada a 50 cópias simultâneas da AMI para cada região de destino.
- Se você quiser copiar uma AMI de virtualização paravirtual (PV) para outra região, a região de destino deve oferecer suporte à virtualização fotovoltaica. AMIs Para obter mais informações, consulte Linux AMI virtualization types.
- Você não pode criar uma cópia não criptografada de um snapshot criptografado. Se você não especificar uma chave gerenciada pelo cliente AWS Key Management Service(AWS KMS) para o parâmetro `KmsKeyId`, o Image Builder usa a chave padrão para o Amazon Elastic Block Store (Amazon EBS). Para obter mais informações, consulte Criptografia do Amazon EBS no Guia do usuário do Amazon Elastic Compute Cloud.

Para obter mais informações, consulte CreateDistributionConfigurationa Referência da API EC2 Image Builder.

Configurar a distribuição entre contas para uma AMI do Image Builder com o console

Esta seção descreve como criar e definir configurações de distribuição para distribuição entre contas do Image Builder AMIs usando o.Console de gerenciamento da AWS A configuração da distribuição entre contas requer permissões específicas do IAM. Você deve preencher o Pré-requisitos para a distribuição da AMI entre contas para esta seção antes de continuar.

Para criar configurações de distribuição no console do Image Builder, siga estas etapas:

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Escolha Configurações de distribuição no painel de navegação. Isso mostra uma lista das configurações de distribuição criadas em sua conta.
3. Na parte superior da página Configurações de distribuição, escolha Criar configurações de distribuição. Isso o leva para a página Criar configurações de distribuição.
4. Na seção Tipo de imagem, escolha imagem de máquina da Amazon (AMI) como o Tipo de saída. Essa é a configuração padrão.
5. Na seção Geral, insira o Nome do recurso de configurações de distribuição que você deseja criar (obrigatório).
6. Na seção Configurações de região, insira um ID de conta de 12 dígitos para o qual você deseja distribuir sua AMI em Contas de destino da região selecionada e pressione Enter. Isso verifica se a formatação está correta e, em seguida, exibe o ID da conta que você inseriu abaixo da caixa. Repita o processo para adicionar mais contas.

   Para remover uma conta que você inseriu, escolha o X exibido à direita do ID da conta.

   Insira o Nome da AMI de saída para cada região.
7. Continue especificando quaisquer configurações adicionais necessárias e escolha Criar configurações para criar seu novo recurso de configurações de distribuição.

Configure a distribuição entre contas para uma AMI do Image Builder a partir do AWS CLI

Esta seção descreve como definir um arquivo de configurações de distribuição e usar o create-image comando AWS CLI para criar e distribuir uma AMI do Image Builder entre contas.

A configuração da distribuição entre contas requer permissões específicas do IAM. Você deve preencher o Pré-requisitos para a distribuição da AMI entre contas para esta seção antes de executar o comando create-image.

1. Configurar um arquivo de configurações de distribuição

   Antes de usar o create-image comando no AWS CLI para criar uma AMI do Image Builder que seja distribuída para outra conta, você deve criar uma estrutura `DistributionConfiguration` JSON que especifique a conta de destino IDs nas `AmiDistributionConfiguration` configurações. Você deve especificar pelo menos um `AmiDistributionConfiguration` na região de origem.

   O arquivo de amostra a seguir, denominado `create-distribution-configuration.json`, mostra a configuração para distribuição de imagem entre contas na região de origem.

```
{
 "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. Criar as configurações de distribuição

   Para criar um recurso de configurações de distribuição do Image Builder usando o create-distribution-configurationcomando no AWS CLI, forneça os seguintes parâmetros no comando:

   - Insira o nome da distribuição no parâmetro `--name`.
   - Anexe o arquivo JSON de configuração de distribuição que você criou no parâmetro `--cli-input-json`.

```
aws imagebuilder create-distribution-configuration --name my distribution name --
cli-input-json file://create-distribution-configuration.json
```

> **Note**
>
> - É necessário incluir a notação `file://` no início do caminho do arquivo JSON.
> - O caminho para o arquivo JSON deve seguir a convenção apropriada para o sistema operacional de base no qual você está executando o comando. Por exemplo, o Windows usa a barra invertida (\) para se referir ao caminho do diretório, enquanto o Linux e o macOS usam a barra (/).

Você também pode fornecer o JSON diretamente no comando, usando o parâmetro `--distributions`.

Configurar a distribuição da AMI com um modelo de EC2 lançamento

Para ajudar a garantir uma experiência de lançamento consistente para sua AMI do Image Builder nas contas e regiões de destino, você pode especificar um modelo de EC2 lançamento da Amazon em suas configurações de distribuição, usando`launchTemplateConfigurations`. Quando `launchTemplateConfigurations` estão presentes durante o processo de distribuição, o Image Builder cria uma nova versão do modelo de lançamento que inclui todas as configurações originais do modelo e a nova AMI ID da compilação. Para obter mais informações sobre como iniciar uma EC2 instância usando um modelo de execução, consulte um dos links a seguir, dependendo do sistema operacional de destino.

- Executar uma instância Linux a partir de um modelo de execução
- Executar uma instância Windows a partir de um modelo de execução

> **Note**
>
> Quando você inclui um modelo de execução para habilitar a inicialização rápida do Windows em sua imagem, o modelo de execução deve incluir a tag a seguir para que o Image Builder possa habilitá-la em seu nome.

```
    CreatedBy: EC2 Image Builder
```

Adicione um modelo de EC2 lançamento às configurações de distribuição da AMI a partir do console

Para fornecer um modelo de execução com sua AMI de saída, siga estas etapas no console:

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Escolha Configurações de distribuição no painel de navegação. Isso mostra uma lista das configurações de distribuição criadas em sua conta.
3. Na parte superior da página Configurações de distribuição, escolha Criar configurações de distribuição. Isso abre a página Criar configurações de distribuição.
4. Na seção Tipo de imagem, escolha imagem de máquina da Amazon (AMI) como o tipo de saída. Essa é a configuração padrão.
5. Na seção Geral, insira o Nome do recurso de configurações de distribuição que você deseja criar (obrigatório).
6. Na seção Configurações da região, selecione o nome de um modelo de EC2 lançamento na lista. Se não houver modelos de lançamento em sua conta, escolha Criar novo modelo de lançamento, que abre os modelos de lançamento no EC2painel.

   Selecione a caixa de seleção Definir a versão padrão para atualizar a versão padrão do modelo de execução para a nova versão que o Image Builder cria com sua AMI de saída.

   Para adicionar outro modelo de lançamento à região selecionada, escolha Adicionar configuração de modelo de lançamento.

   Para remover um modelo de lançamento, escolha Remover.
7. Continue especificando quaisquer configurações adicionais necessárias e escolha Criar configurações para criar seu novo recurso de configurações de distribuição.

Adicione um modelo de EC2 lançamento às configurações de distribuição da AMI a partir do AWS CLI

Esta seção descreve como definir um arquivo de configurações de distribuição com um modelo de execução e usar o create-image comando em AWS CLI para criar e distribuir uma AMI do Image Builder e uma nova versão do modelo de execução que a usa.

1. Configurar um arquivo de configurações de distribuição

   Antes de criar uma AMI do Image Builder com um modelo de execução, usando o AWS CLI, você deve criar uma estrutura JSON de configuração de distribuição que especifique as `launchTemplateConfigurations` configurações. Você deve especificar pelo menos uma entrada `launchTemplateConfigurations` na região de origem.

   O arquivo de exemplo a seguir, denominado`create-distribution-config-launch-template.json`, mostra alguns cenários possíveis para a configuração do modelo de execução na região de origem.

```
{
    "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. Criar as configurações de distribuição

   Para criar um recurso de configurações de distribuição do Image Builder usando o create-distribution-configurationcomando no AWS CLI, forneça os seguintes parâmetros no comando:

   - Insira o nome da distribuição no parâmetro `--name`.
   - Anexe o arquivo JSON de configuração de distribuição que você criou no parâmetro `--cli-input-json`.

```
aws imagebuilder create-distribution-configuration --name my distribution name--
cli-input-json file://create-distribution-config-launch-template.json
```

   > **Note**
   >
   > - É necessário incluir a notação `file://` no início do caminho do arquivo JSON.
   > - O caminho para o arquivo JSON deve seguir a convenção apropriada para o sistema operacional de base no qual você está executando o comando. Por exemplo, o Windows usa a barra invertida (\) para se referir ao caminho do diretório, enquanto o Linux e o macOS usam a barra (/).

Você também pode fornecer o JSON diretamente no comando, usando o parâmetro `--distributions`.

Use recursos aprimorados de distribuição de AMI

O Image Builder oferece recursos avançados de distribuição que oferecem flexibilidade e controle sobre como AMIs você é distribuído entre regiões e contas. Esses recursos separam a distribuição do processo de criação, permitindo que você distribua imagens existentes sob demanda, recupere-se de falhas de distribuição com eficiência e implemente estratégias de distribuição controladas em vários estágios por meio de fluxos de trabalho personalizáveis.

Você pode usar recursos aprimorados de distribuição de AMI no Image Builder para realizar atividades de distribuição diretamente sem a necessidade de executar novamente uma criação de imagem completa.

Distribuição desacoplada

A DistributeImage API aceita três tipos de referências de imagem de origem:

- ID da AMI — Um identificador padrão da AMI (por exemplo,`ami-0abcdef1234567890`)
- Parâmetro SSM — Um parâmetro SSM que armazena uma ID de AMI (por exemplo,) `ssm:/my/ami/parameter`
- Versão ARN do Image Builder — Uma versão ARN da imagem do Image Builder

Tentativa de distribuição

Se uma distribuição de imagem falhar, use a `RetryImage` API para tentar novamente a distribuição. Isso reduz o tempo para solucionar a causa da falha, evitando a reconstrução completa da imagem. Use `RetryImage` depois de resolver a causa subjacente da falha de distribuição.

A RetryImage API aceita uma versão ARN de criação de imagem (por exemplo,`arn:aws:imagebuilder:us-west-2:123456789012:image/my-image/1.0.0/1`). Quando você invoca a API, o Image Builder retoma automaticamente o processo de distribuição a partir do ponto de falha usando a configuração e as configurações originais da distribuição. A `RetryImage` API pode tentar novamente distribuições que falharam durante a fase de distribuição, a fase de teste ou a fase de integração. Ele funciona AMIs nos seguintes estados: pendente, com falha, excluído ou disponível.

Pré-requisitos

Antes de tentar novamente uma distribuição, verifique o seguinte:

- Você identificou e resolveu a causa raiz da falha. Revise os registros de distribuição no CloudWatch Logs para obter detalhes do erro.
- Você tem as permissões do IAM necessárias para tentar criar novamente a imagem.
- Para falhas de distribuição entre contas, verifique se a conta `EC2ImageBuilderDistributionCrossAccountRole` de destino tem a `Ec2ImageBuilderCrossAccountDistributionAccess` política anexada.

Importante: tentar novamente sem corrigir o problema subjacente resultará em falhas repetidas.

Fluxos de trabalho de distribuição

Os fluxos de trabalho de distribuição são um novo tipo de fluxo de trabalho que complementa os fluxos de trabalho de criação e teste, permitindo que você defina e controle o processo de distribuição com etapas sequenciais. Com fluxos de trabalho de distribuição, você pode criar processos de distribuição personalizados que incluem operações de cópia da AMI, wait-for-action pontos de verificação, modificações de atributos de imagem e outras etapas relacionadas à distribuição. Isso fornece controle estruturado sobre como AMIs você é distribuído, com visibilidade em nível de etapas, recursos de distribuição paralela e relatórios granulares de erros.

Para saber mais sobre como criar e personalizar fluxos de trabalho, consulte Gerenciar fluxos de trabalho de imagem.

Compartilhe recursos do Image Builder com AWS RAM

EC2 O Image Builder se integra com AWS Resource Access Manager (AWS RAM) para que você possa compartilhar os seguintes tipos de recursos do Image Builder com qualquer um Conta da AWS ou por meio AWS Organizations deles.

- Componentes
- Imagens
- Fórmulas

Para compartilhar recursos AWS RAM, você deve criar um compartilhamento de recursos. Um compartilhamento de recursos especifica os recursos a serem compartilhados, e os consumidores com os quais compartilhá-los. Os consumidores podem ser indivíduos Contas da AWS, unidades organizacionais ou uma organização inteira em AWS Organizations. A lista a seguir inclui os tipos de contas e organizações com as quais você pode compartilhar.

- Específico Contas da AWS dentro ou fora de sua organização em AWS Organizations.
- Uma unidade organizacional dentro da organização no AWS Organizations.
- Toda a organização no AWS Organizations.
- AWS Organizations ou OUs fora de sua organização.

Nesse modelo, o Conta da AWS proprietário do recurso (proprietário) o compartilha com outros Contas da AWS ou por meio de AWS Organizations (consumidores) na mesma região. Quando um recurso compartilhado é atualizado, os consumidores recebem essas atualizações automaticamente.

> **Note**
>
> Componentes, imagens e fórmulas de imagens compartilhadas contam somente para os limites de recursos correspondentes do proprietário. Os limites de recurso dos consumidores não são afetados por recursos que são compartilhados com eles.

Tópicos

- Proprietários de recurso

- Consumidores de recursos
- Crie um compartilhamento AWS RAM de recursos para seus recursos do Image Builder
- Cancelar o compartilhamento de um recurso do Image Builder do AWS RAM

Proprietários de recurso

Os recursos do Image Builder só podem ser compartilhados no Região da AWS local em que foram criados. Quando você compartilha esses recursos, eles não se replicam em todas as regiões.

Para obter uma lista dos recursos do Image Builder que você possui e pode compartilhar, especifique o filtro de propriedade no console ou execute o comando na AWS CLI.

- Listar componentes do Image Builder
- Listar imagens
- Liste e visualize detalhes da fórmula da imagem
- Liste e visualize detalhes da fórmula de contêiner

Para obter mais informações sobre AWS RAM, consulte o Guia AWS RAM do usuário.

Pré-requisitos para compartilhar recursos do Image Builder

Como compartilhar um recurso do Image Builder, como um componente, uma imagem ou uma fórmula:

- Você Conta da AWS deve possuir o recurso Image Builder que deseja compartilhar. Não é possível compartilhar recursos que tenham sido compartilhados com você.
- A chave AWS Key Management Service (AWS KMS) associada aos recursos criptografados deve ser explicitamente compartilhada com as contas, organizações ou OUs contas de destino.
- Para compartilhar seus recursos do Image Builder com AWS Organizations e OUs usando AWS RAM, você deve habilitar o compartilhamento. Para obter mais informações, consulte Habilitar o compartilhamento com o AWS Organizations no Guia do usuário do AWS RAM .
- Se você distribuir uma imagem criptografada com AWS KMS várias contas em diferentes regiões, deverá criar uma chave KMS e um alias em cada região de destino. Além disso, as pessoas que lançarão instâncias nessas regiões precisarão acessar a chave KMS especificada por meio da Política de chaves.

Os seguintes recursos que o Image Builder cria a partir da sua compilação de pipeline não são considerados recursos do Image Builder, mas sim recursos externos que o Image Builder distribui em sua conta e para as Regiões da AWS contas e organizações ou unidades organizacionais (OUs) que você especifica em sua configuração de distribuição.

- Imagens de máquinas da Amazon (AMIs)
- Imagens de contêiner que residem no Amazon ECR

Para obter mais informações sobre as configurações de distribuição para suas AIM, consulte Criar e atualizar as configurações de distribuição da AMI. Para obter mais informações sobre as configurações de distribuição da sua imagem de contêiner no Amazon ECR, consulte Criar e atualizar configurações de distribuição para imagens de contêiner.

Para obter mais informações sobre como compartilhar sua AMI com AWS Organizations e OUs, consulte Compartilhar uma AMI com organizações ou OUs.

Consumidores de recursos

Os consumidores podem usar um recurso compartilhado, mas não podem modificá-lo de forma alguma. Ao criar fórmulas do Image Builder, eles podem especificar uma imagem compartilhada como imagem de base e adicionar componentes compartilhados. Eles também podem especificar uma fórmula compartilhada ao criar um pipeline de imagem do Image Builder ou ao usar o comando create-image na AWS CLI.

Se você pertence a uma organização e o compartilhamento dentro da sua organização está ativado, os consumidores da sua organização recebem automaticamente acesso ao recurso compartilhado. AWS Organizations Caso contrário, os consumidores receberão um convite para participar do compartilhamento de recursos e acesso ao recurso compartilhado depois de aceitar o convite.

Crie um compartilhamento AWS RAM de recursos para seus recursos do Image Builder

Para compartilhar um componente, imagem ou receita do Image Builder, você deve adicioná-lo a um compartilhamento AWS Resource Access Manager de recursos. O compartilhamento de recursos especifica os recursos a serem compartilhados e os consumidores com os quais compartilhá-los.

As seguintes opções estão disponíveis para compartilhar recursos:

Opção 1: criar um compartilhamento de recursos de RAM

Ao criar um compartilhamento de recursos de RAM, você pode compartilhar um componente, imagem ou fórmula de sua propriedade em uma única etapa. Use um dos métodos a seguir para criar seu compartilhamento de recursos:

- Console

  Para criar seu compartilhamento de recursos usando o AWS RAM console, consulte Compartilhar AWS recursos de sua propriedade no Guia AWS RAM do usuário.
- AWS CLI

  Para criar seu compartilhamento de recursos usando a interface da linha de AWS RAM comando, execute o create-resource-share comando no AWS CLI.

Opção 2: aplicar uma política de recursos e promover um compartilhamento de recursos existente

A segunda opção para compartilhar seus recursos envolve duas etapas, executando comandos no AWS CLI para ambas. A primeira etapa usa os comandos do Image Builder no AWS CLI para aplicar políticas baseadas em recursos ao recurso compartilhado. A segunda etapa promove o recurso para um compartilhamento de recursos de RAM usando o promote-resource-share-created-from-policy AWS RAM comando no AWS CLI para garantir que o recurso esteja visível para todos os diretores com quem você o compartilhou.

1. Aplicar a política de recursos

   Para aplicar com êxito a política de recursos, você deve garantir que a conta com a qual está compartilhando tenha permissão para acessar quaisquer recursos subjacentes.

   Escolha a guia que corresponde ao seu tipo de recurso para o comando aplicável.

   Image

   Você pode aplicar uma política de recursos a uma imagem para permitir que outras pessoas a usem como imagem base em suas fórmulas.

   Execute o comando put-image-policy Image Builder no AWS CLI, para identificar os AWS principais com os quais compartilhar a imagem.

```
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

Você pode aplicar uma política de recursos a um componente de compilação ou teste para permitir o compartilhamento entre contas. Este comando permite que outras contas usem seu componente em suas fórmulas. Para aplicar com êxito a política de recursos, você deve garantir que a conta com a qual você está compartilhando tenha permissão para acessar quaisquer recursos referidos pelo componente compartilhado, tais como arquivos hospedados em repositórios privados.

Execute o comando put-component-policy Image Builder no AWS CLI, para identificar os AWS principais com os quais compartilhar o componente.

```
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

Você pode aplicar uma política de recursos a uma fórmula de imagem para permitir o compartilhamento entre contas. Este comando permite que outras contas usem sua fórmula para criar imagens em suas contas. Para aplicar com êxito a política de recursos, você deve garantir que a conta com a qual você está compartilhando tenha permissão para acessar quaisquer recursos à qual a fórmula faz referência, tais como imagens de base e componentes selecionados.

Execute o comando put-image-recipe-policy Image Builder no AWS CLI, para identificar os AWS principais com os quais compartilhar a imagem.

```
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

Você pode aplicar uma política de recursos a uma fórmula de contêiner para permitir o compartilhamento entre contas. Este comando permite que outras contas usem sua fórmula para criar imagens em suas contas. Para aplicar com êxito a política de recursos, você deve garantir que a conta com a qual você está compartilhando tenha permissão para acessar quaisquer recursos à qual a fórmula faz referência, tais como imagens de base e componentes selecionados.

Execute o comando put-container-recipe-policy Image Builder no AWS CLI, para identificar os AWS principais com os quais compartilhar a imagem.

```
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**
>
> Para definir as políticas corretas para compartilhar e não compartilhar um recurso, o proprietário do recurso deve ter permissões do `imagebuilder:put*`.

2. Promover como compartilhamento de recursos de RAM

   Para garantir que o recurso esteja visível para todos os diretores com quem você o compartilhou, execute o promote-resource-share-created-from-policy AWS RAM comando no AWS CLI.

Cancelar o compartilhamento de um recurso do Image Builder do AWS RAM

Para cancelar o compartilhamento de um recurso do Image Builder que você possui, como um componente compartilhado, imagem ou receita, você deve removê-lo do compartilhamento de AWS Resource Access Manager recursos. É possível fazer isso usando o console do AWS RAM ou a AWS CLI.

> **Note**
>
> Os proprietários só podem excluir um recurso compartilhado quando ele deixar de ser compartilhado. Um proprietário não pode deixar de compartilhar esses recursos até que nenhum dos consumidores dependa deles.

Para cancelar o compartilhamento de um componente, imagem ou fórmula compartilhada de sua propriedade usando o console AWS Resource Access Manager

Consulte Atualização de um compartilhamento de atributos no Guia do usuário do AWS RAM .

Para cancelar o compartilhamento de um componente, imagem ou fórmula compartilhada de sua propriedade usando o AWS CLI

Use o comando disassociate-resource-share para interromper o compartilhamento do recurso.

Marcar recursos de saída do Image Builder

Marcar seus recursos pode ser útil para filtrar e monitorar os custos dos recursos ou outras categorias. Você também pode controlar o acesso com base em tags. Para obter mais informações sobre a autorização baseada em tags, consulte Autorização baseada em tags do construtor de imagens do construtor.

O Image Builder oferece suporte às seguintes tags dinâmicas:

- `- {{imagebuilder:buildDate}}`

  Resolve para a compilação date/time no momento da construção.

- `- {{imagebuilder:buildVersion}}`

  Será resolvido como versão de compilação, que é um número localizado no final de um nome de recurso da Amazon (ARN) do Image Builder. Por exemplo, `"arn:aws:imagebuilder:us-west-2:123456789012:component/myexample-component/2019.12.02/1"` mostra a versão de compilação como 1.

Para ajudá-lo a acompanhar as Amazon Machine Images (AMIs) que você distribuiu, o Image Builder adiciona automaticamente as seguintes tags à sua saída AMIs.

- `"CreatedBy":"EC2 Image Builder"`
- `"Ec2ImageBuilderArn":"arn:aws:imagebuilder:us-west-2:123456789012:image/simple-recipe-linux/1.0.0/10"`: essa tag contém o ARN do recurso de imagem do Image Builder que foi usado para criar a AMI.

Conteúdo

- Marque um recurso do AWS CLI
- Desmarque um recurso do AWS CLI
- Liste todas as tags de um recurso específico do AWS CLI

Marque um recurso do AWS CLI

O exemplo a seguir mostra como usar um comando da imagebuilder CLI para adicionar e marcar um recurso no EC2 Image Builder. Você deve fornecer os `resourceArn` e as tags que serão aplicadas a eles.

O conteúdo do `tag-resource.json` de exemplo é o seguinte:

```
{
    "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/my-example-pipeline",
    "tags": {
        "KeyName": "KeyValue"
    }
}
```

Execute o seguinte comando, que faz referência ao arquivo anterior do `tag-resource.json`.

```
aws imagebuilder tag-resource --cli-input-json file://tag-resource.json
```

Desmarque um recurso do AWS CLI

O seguinte exemplo mostra como usar um comando da CLI do imagebuilder para desmarcar um recurso. Você deve fornecer as `resourceArn` e as chaves para desmarcar.

O conteúdo do `untag-resource.json` de exemplo é o seguinte:

```
{
    "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/my-example-pipeline",
    "tagKeys": [
        "KeyName"
    ]
}
```

Execute o seguinte comando, que faz referência ao arquivo anterior do `untag-resource.json`.

```
aws imagebuilder untag-resource --cli-input-json file://untag-resource.json
```

Liste todas as tags de um recurso específico do AWS CLI

O exemplo a seguir mostra como usar um comando da CLI do imagebuilder para indicar todas as tags de um recurso específico.

```
aws imagebuilder list-tags-for-resource --resource-arn arn:aws:imagebuilder:us-
west-2:123456789012:image-pipeline/my-example-pipeline
```

Excluir recursos desatualizados ou não utilizados do Image Builder

Seu ambiente Image Builder, assim como sua casa, precisa de manutenção regular para ajudá-lo a encontrar o que precisa e concluir suas tarefas sem bagunça. Certifique-se de limpar regularmente os recursos temporários que você criou para testes. Caso contrário, você pode esquecer esses recursos e, mais tarde, não se lembrar para que eles foram usados. E então, talvez você não possa se livrar deles com segurança.

A exclusão de recursos não exclui nenhuma imagem de contêiner da Amazon EC2 AMIs ou do Amazon ECR criada durante o processo de criação da imagem. Você deve limpá-los separadamente, usando as ações apropriadas do console Amazon EC2 ou Amazon ECR, ou a API ou AWS CLI os comandos.

> **Tip**
>
> Para evitar erros de dependência ao excluir recursos, certifique-se de excluir seus recursos na seguinte ordem:
>
> 1. Pipeline de imagens
> 2. Fórmula da imagem
> 3. Todos os recursos restantes

Exclua recursos do Console de gerenciamento da AWS

Para excluir um pipeline de imagens e seus recursos, siga estas etapas:

Exclua o pipeline

1. Para ver uma lista dos pipelines de compilação criados em sua conta, escolha Pipelines de imagem no painel de navegação.
2. Marque a caixa de seleção ao lado de Pipeline de nome para selecionar o pipeline que deseja excluir.
3. Na parte superior do painel Pipelines de imagem, no menu Ações, escolha Excluir.
4. Insira `Delete` para confirmar a exclusão e depois escolha Excluir.

Exclua a fórmula

1. Para ver uma lista das fórmulas criadas em sua conta, escolha Fórmulas de imagem no painel de navegação.
2. Marque a caixa de seleção ao lado de nome da fórmula para selecionar a fórmula que deseja excluir.
3. Na parte superior do painel Fórmulas de imagens, no menu Ações, escolha Excluir fórmula.
4. Insira `Delete` para confirmar a exclusão e depois escolha Excluir.

Excluir uma configuração de infraestrutura

1. Para ver uma lista das configurações de infraestrutura criadas em sua conta, escolha Configuração de infraestrutura no painel de navegação.
2. Marque a caixa de seleção ao lado de Nome da configuração para selecionar a configuração de infraestrutura que deseja excluir.
3. Na parte superior do painel Configurações de infraestrutura, escolha Excluir.
4. Insira `Delete` para confirmar a exclusão e depois escolha Excluir.

Excluir configurações de distribuição

1. Para ver uma lista das configurações de distribuição criadas em sua conta, escolha Configurações de distribuição no painel de navegação.
2. Marque a caixa de seleção ao lado de Nome da configuração para selecionar as configurações de distribuição que você criou para este tutorial.
3. Na parte superior do painel Configurações de distribuição, escolha Excluir.
4. Insira `Delete` para confirmar a exclusão e depois escolha Excluir.

Excluir uma imagem

1. Para ver uma lista das imagens criadas em sua conta, escolha Imagens no painel de navegação.
2. Escolha a Versão da imagem que deseja remover. Isso abre a página Versões de compilação de imagens.
3. Marque a caixa de seleção ao lado da Versão da imagem que você deseja excluir. Você pode selecionar mais de uma versão de imagem de cada vez.

4. Na parte superior do painel Versões de compilação de imagens, escolha Excluir versão.
5. Insira `Delete` para confirmar a exclusão e depois escolha Excluir.

Exclua um pipeline de imagens do AWS CLI

Os exemplos a seguir mostram como excluir recursos do Image Builder usando AWS CLI o. Conforme mencionado anteriormente, os recursos devem ser excluídos na seguinte ordem para evitar erros de dependência:

1. Pipeline de imagens
2. Fórmula da imagem
3. Todos os recursos restantes

Exclua o pipeline de imagens do AWS CLI

O exemplo a seguir mostra como excluir um pipeline de imagem especificando seu ARN.

```
aws imagebuilder delete-image-pipeline --image-pipeline-arn arn:aws:imagebuilder:us-
west-2:123456789012:image-pipeline/my-example-pipeline
```

Exclua a receita da imagem do AWS CLI

O exemplo a seguir mostra como excluir uma fórmula de imagem especificando seu ARN.

```
aws imagebuilder delete-image-recipe --image-recipe-arn arn:aws:imagebuilder:us-
west-2:123456789012:image-recipe/my-example-recipe/2019.12.03
```

Excluir uma configuração de infraestrutura

O exemplo a seguir mostra como excluir um recurso de configuração de infraestrutura especificando seu ARN.

```
aws imagebuilder delete-infrastructure-configuration --infrastructure-configuration-arn
 arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/my-example-
infrastructure-configuration
```

Excluir configurações de distribuição

O exemplo a seguir mostra como excluir um recurso de configurações de distribuição especificando seu ARN.

```
aws imagebuilder delete-distribution-configuration --distribution-configuration-arn
 arn:aws:imagebuilder:us-west-2:123456789012:distribution-configuration/my-example-
distribution-configuration
```

Excluir uma imagem

O exemplo a seguir mostra como excluir uma versão de compilação de imagem especificando seu ARN.

```
aws imagebuilder delete-image --image-build-version-arn arn:aws:imagebuilder:us-
west-2:123456789012:image/my-example-image/2019.12.02/1
```

Excluir um componente

O exemplo a seguir mostra como usar um comando da imagebuilder CLI para excluir uma versão de compilação do componente especificando seu ARN.

```
aws imagebuilder delete-component --component-build-version-arn
 arn:aws:imagebuilder:us-west-2:123456789012:component/my-example-
component/2019.12.02/1
```

> **Important**
>
> Certifique-se de que não haja fórmulas que façam referência à versão de compilação do componente antes de excluí-la. Não fazer isso pode causar falhas na tubulação.

Gerenciar fluxos de trabalho de compilação e teste para imagens do Image Builder

Um fluxo de trabalho de imagem define a sequência de etapas que o EC2 Image Builder executa durante os estágios de criação, teste e distribuição do processo de criação da imagem. Isso faz parte do framework do fluxo de trabalho do Image Builder.

Benefícios do fluxo de trabalho de imagem

- Com os fluxos de trabalho de imagem, você tem mais flexibilidade, visibilidade e controle sobre o processo de criação de imagens.
- Você pode adicionar etapas personalizadas do fluxo de trabalho ao definir o documento do fluxo de trabalho ou optar por usar o fluxo de trabalho padrão do Image Builder.
- É possível excluir as etapas do fluxo de trabalho que estejam incluídas nos fluxos de trabalho de imagem padrão.
- Você também pode criar fluxos de trabalho somente de teste que ignoram totalmente o processo de compilação. Você pode fazer o mesmo para criar fluxos de trabalho somente de compilação ou somente de distribuição.

> **Note**
>
> Você não pode modificar um fluxo de trabalho existente, mas pode cloná-lo ou criar uma nova versão.

tópicos de fluxo de trabalho

- Framework do fluxo de trabalho: estágios
- Acesso ao serviço
- Use fluxos de trabalho gerenciados para suas imagens
- Listar fluxos de trabalho de imagem
- Criar um fluxo de trabalho de imagem
- Criar um documento de fluxo de trabalho do YAML

Framework do fluxo de trabalho: estágios

Para personalizar fluxos de trabalho de imagem, é importante entender os estágios do fluxo de trabalho que compõem o framework do fluxo de trabalho de criação de imagens.

A estrutura do fluxo de trabalho de criação de imagens inclui os seguintes estágios distintos.

1. Estágio de construção (pré-instantâneo) — Durante o estágio de construção, você faz alterações na instância de EC2 compilação da Amazon que está executando sua imagem base, para criar a linha de base para sua nova imagem. Por exemplo, sua fórmula pode incluir componentes que instalam uma aplicação ou modificam as configurações do firewall do sistema operacional.

   Depois que esse estágio for concluído com êxito, o Image Builder cria um snapshot ou imagem de contêiner que ele usa para o estágio de teste e além.
2. Estágio de teste (pós-captura instantânea) — Durante o estágio de teste, há algumas diferenças entre imagens que criam AMIs e imagens de contêineres. Para fluxos de trabalho da AMI, o Image Builder executa uma EC2 instância a partir do snapshot que ele criou como etapa final do estágio de criação. Os testes são executados na nova instância para validar as configurações e garantir que a instância esteja funcionando conforme o esperado. Para fluxos de trabalho de contêineres, os testes são executados na mesma instância usada para a compilação.
3. Estágio de distribuição (pós-construção) — Durante o estágio de distribuição, uma vez que a criação da imagem esteja concluída e tenha passado com sucesso em todos os testes, a imagem de saída passa por várias fases de distribuição. Essas fases abrangem operações de cópia da AMI, modificações de atributos de imagem, compartilhamento de imagens e outras etapas de distribuição relacionadas.

Acesso ao serviço

Para executar fluxos de trabalho de imagem, o Image Builder precisa de permissão para realizar ações de fluxo de trabalho. Veja a seguir como especificar um perfil vinculado a serviço AWSServiceRoleForImageBuilder ou especificar seu próprio perfil personalizado para acesso ao serviço.

- Console: na Etapa 3: definir o processo de criação de imagem do assistente de pipeline, selecione o perfil vinculado a serviço ou seu próprio perfil personalizado na lista de perfis do IAM no painel Acesso ao serviço.

- API Image Builder — Na solicitação de CreateImageação, especifique a função vinculada ao serviço ou sua própria função personalizada como o valor do parâmetro. `executionRole`

Para saber mais sobre como criar uma função de serviço, consulte Criação de uma função para delegar permissões a um AWS serviço no Guia do AWS Identity and Access Management usuário.

Use fluxos de trabalho gerenciados para suas imagens

Os fluxos de trabalho gerenciados são criados e mantidos por AWS. Ao usar fluxos de trabalho gerenciados em seus pipelines de imagens ou para criar imagens únicas, você pode selecionar o Amazon Resource Name (ARN) do fluxo de trabalho gerenciado que deseja usar. A Amazon fornece as versões mais recentes que têm patches e outras atualizações aplicadas. Para obter uma lista de fluxos de trabalho gerenciadosListar fluxos de trabalho de imagem, consulte e filtre em Owner = Amazon (console).

Listar fluxos de trabalho de imagem

Na página de lista Fluxos de trabalho de imagem no console do Image Builder, você pode obter uma lista dos recursos de fluxo de trabalho de imagem do Image Builder que você possui ou aos quais tem acesso, junto com alguns detalhes importantes sobre esses recursos. Você também pode usar comandos ou ações com a API Image Builder ou AWS CLI listar fluxos de trabalho de imagens na sua conta. SDKs

Você pode usar um dos métodos a seguir para listar os recursos de fluxo de trabalho de imagem de sua propriedade ou aos quais você tem acesso. Para a ação da API, consulte ListWorkflowsna Referência da API do EC2 Image Builder. Para ver a solicitação de SDK associada, consulte o link Consulte também na mesma página.

Você pode filtrar os detalhes a seguir para simplificar a lista de resultados. Por exemplo, se você filtrar por Proprietário = Amazon no console, a lista exibirá somente fluxos de trabalho AWS gerenciados.

- Fluxo de trabalho
- Versão
- Tipo
- Proprietário

Console

Detalhes do fluxo de trabalho

Os detalhes na página da lista Fluxos de trabalho de imagens no console do Image Builder incluem o seguinte:

- Fluxo de trabalho: o nome da versão mais recente do recurso de fluxo de trabalho de imagem. No console do Image Builder, a coluna Fluxo de trabalho está vinculada à página de detalhes do fluxo de trabalho.
- Versão: a versão mais recente do recurso de fluxo de trabalho de imagem.
- Tipo — O tipo de fluxo de trabalho: `BUILDTEST`,, ou`DISTRIBUTION`.
- Proprietário: o proprietário do recurso de fluxo de trabalho.
- Data de criação: a data e a hora em que o Image Builder criou a versão mais recente do recurso de fluxo de trabalho de imagem.
- ARN: o nome do recurso da Amazon (ARN) da versão atual do recurso de fluxo de trabalho de imagem.

Listar fluxos de trabalho de imagem

Para listar recursos fluxo de trabalho de imagem no console do Image Builder, execute as seguintes etapas:

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Escolha Fluxos de trabalho de imagem no painel de navegação.

Filtrar resultados

Na página da lista Fluxos de trabalho de imagem, você pode pesquisar fluxos de trabalho de imagem específicos para filtrar seus resultados. Os filtros a seguir estão disponíveis para fluxos de trabalho de imagem:

`Workflow`

Para otimizar os resultados, você pode inserir o nome completo ou parte do nome de um fluxo de trabalho. O padrão é mostrar todos os fluxos de trabalho na lista.

`Version`

Para otimizar os resultados, você pode inserir o número de versão completo ou parte do número de versão. O padrão é mostrar todas as versões na lista.

`Type`

Você pode filtrar pelo tipo de fluxo de trabalho ou visualizar todos os tipos. O padrão é mostrar todos os tipos de fluxo de trabalho na lista.

- BUILD
- TEST
- DISTRIBUIÇÃO

`Owner`

Quando você seleciona o filtro de proprietário na barra de pesquisa, o Image Builder mostra uma lista dos proprietários dos fluxos de trabalho de imagem em sua conta. É possível selecionar um proprietário na lista para otimizar os resultados. O padrão é mostrar todos os proprietários na lista.

- Conta da AWS: a conta do proprietário do recurso de fluxo de trabalho.
- Amazon: recursos de imagem de propriedade e gerenciados pela Amazon.

AWS CLI

Ao executar o list-workflows comando no AWS CLI, você pode obter uma lista dos fluxos de trabalho de imagem que você possui ou aos quais tem acesso.

O exemplo de comando a seguir mostra como usar o comando list-workflows sem filtros para listar todos os recursos de fluxo de trabalho de imagem do Image Builder fluxos de sua propriedade ou aos quais você tem acesso.

Exemplo: listar todos os fluxos de trabalho de imagem

```
aws imagebuilder list-workflows
```

Saída:

```
{
    "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"
        },
        {
            "name": "example-distribution-workflow",
            "dateCreated": "2025-11-19T10:25:10.425Z",
            "version": "1.0.0",
            "owner": "111122223333",
            "type": "DISTRIBUTION",
            "arn": "arn:aws:imagebuilder:us-west-2:111122223333:workflow/
distribution/example-distribution-workflow/1.0.0"
        }
    ]
}
```

Ao executar o comando list-workflows, você pode aplicar filtros para otimizar os resultados, como mostra o exemplo a seguir. Para obter mais informações sobre como filtrar seus resultados, consulte o comando list-workflows na Referência de comandos da AWS CLI .

Exemplo: filtro para fluxos de trabalho de compilação

```
aws imagebuilder list-workflows --filters name="type",values="BUILD"
```

Saída:

```
{
    "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"
        }
    ]
}
```

Criar um fluxo de trabalho de imagem

Ao criar um fluxo de trabalho de imagem, você tem mais controle sobre o processo de criação de imagens. É possível especificar qual fluxo de trabalho será executado quando o Image Builder criar sua imagem e quais fluxos de trabalho serão executados quando ele testar a imagem. Você também pode especificar uma chave gerenciada pelo cliente para criptografar os recursos do fluxo de trabalho. Para saber mais sobre criptografia para seus recursos de fluxo de trabalho, consulte Gerenciamento de criptografia e chaves no Image Builder.

Para a criação de imagens, você pode especificar um fluxo de trabalho do estágio de compilação e um ou mais fluxos de trabalho do estágio de teste. Dependendo de suas necessidades, é possível até mesmo ignorar totalmente o estágio de compilação ou teste. Você configura as ações que seu fluxo de trabalho executa no documento de definição do YAML que o fluxo de trabalho usa. Para obter mais informações sobre sintaxe para seu documento do YAML, consulte Criar um documento de fluxo de trabalho do YAML.

Para obter as etapas de criação de um novo fluxo de trabalho de compilação ou teste, selecione a guia que corresponde ao ambiente que você usará.

Console de gerenciamento da AWS

É possível seguir o processo a seguir para criar um novo fluxo de trabalho no console do Image Builder.

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Escolha Fluxos de trabalho de imagem no painel de navegação. Isso exibe uma lista de fluxos de trabalho de imagem de propriedade da sua conta ou aos quais sua conta tem acesso.

> **Note**
>
> Você sempre verá os recursos de fluxo de trabalho gerenciado pela Amazon que o Image Builder usa para seus fluxos de trabalho padrão em sua lista. Para exibir os detalhes desses fluxos de trabalho, você pode selecionar o link Fluxo de trabalho.

3. Para criar um novo fluxo de trabalho, escolha Criar fluxo de trabalho de imagem. Isso exibe a página Criar fluxo de trabalho de imagem.
4. Configure os detalhes do seu novo fluxo de trabalho. Para criar um fluxo de trabalho de compilação, selecione a opção Compilar na parte superior do formulário. Para criar um fluxo de trabalho de teste, selecione a opção Testar na parte superior do formulário. O Image Builder preencherá a lista Modelos com base nessa opção. Todas as outras etapas são as mesmas para os fluxos de trabalho de compilação e teste.

   Geral

   A seção geral inclui configurações que se aplicam ao seu recurso de fluxo de trabalho, como nome e descrição. As configurações gerais incluem o seguinte:

   - Nome do fluxo de trabalho de imagem (obrigatório): o nome do seu fluxo de trabalho de imagem. O nome deve ser exclusivo em sua conta. Os nomes podem ter até 128 caracteres. Os caracteres válidos incluem letras, números, espaços - e _.
   - Versão (obrigatória): a versão semântica do recurso de fluxo de trabalho a ser criado (major.minor.patch).
   - Descrição (opcional): opcionalmente, adicione uma descrição para seu fluxo de trabalho.
   - Chave do KMS (opcional): você pode criptografar seus recursos de fluxo de trabalho com uma chave gerenciada pelo cliente. Para obter mais informações, consulte Criptografar fluxos de trabalho de imagens com uma chave gerenciada pelo cliente.

   Documento de definição

   O documento de fluxo de trabalho do YAML contém todas as configurações do seu fluxo de trabalho.

Conceitos básicos

- Para começar com um modelo padrão do Image Builder como linha de base para seu fluxo de trabalho, selecione a opção Iniciar com base em modelos. Essa opção é selecionada por padrão. Após escolher qual modelo usar na lista Modelos, o sistema vai copiar a configuração padrão do modelo selecionado para o Conteúdo do seu novo documento de fluxo de trabalho, no qual você poderá fazer alterações.
- Para definir seu documento de fluxo de trabalho do zero, selecione a opção Começar do zero. Isso preencherá o Conteúdo com um breve resumo de algumas partes importantes do formato do documento para ajudar você a começar.

O painel Conteúdo inclui uma barra de status na parte inferior que mostra avisos ou erros do seu documento do YAML. Para obter mais informações sobre como criar um documento de fluxo de trabalho do YAML, consulte Criar um documento de fluxo de trabalho do YAML.

5. Após concluir seu fluxo de trabalho ou se quiser salvar o progresso e retornar posteriormente, escolha Criar fluxo de trabalho.

AWS CLI

Antes de executar o create-workflow comando no AWS CLI, você deve criar o documento YAML que contém toda a configuração do seu fluxo de trabalho. Para obter mais informações, consulte Criar um documento de fluxo de trabalho do YAML.

O exemplo a seguir mostra como criar um fluxo de trabalho de compilação com o comando create-workflow da AWS CLI . O parâmetro `--data` se refere a um documento do YAML que contém a configuração de compilação para o fluxo de trabalho que você cria.

Exemplo: criar fluxo de trabalho

```
aws imagebuilder create-workflow --name example-build-workflow --semantic-
version 1.0.0 --type BUILD --data file://example-build-workflow.yml
```

Saída:

```
{
 "workflowBuildVersionArn": "arn:aws:imagebuilder:us-west-2:111122223333:workflow/
build/example-build-workflow/1.0.0/1",
```

```
 "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

O exemplo a seguir mostra como criar um fluxo de trabalho de teste com o comando create-workflow da AWS CLI . O parâmetro `--data` se refere a um documento do YAML que contém a configuração de compilação para o fluxo de trabalho que você cria.

Exemplo: criar fluxo de trabalho de teste

```
aws imagebuilder create-workflow --name example-test-workflow --semantic-
version 1.0.0 --type TEST --data file://example-test-workflow.yml
```

Saída:

```
{
 "workflowBuildVersionArn": "arn:aws:imagebuilder:us-west-2:111122223333:workflow/
test/example-test-workflow/1.0.0/1",
 "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

Criar um documento de fluxo de trabalho do YAML

O documento de definição do formato do YAML configura as etapas de entrada, saída e fluxo de trabalho para os estágios de compilação e teste do processo de compilação da imagem. Você pode começar com base em modelos que incluem etapas padronizadas ou começar do zero para definir seu próprio fluxo de trabalho. Seja com um modelo ou começando do zero, você pode personalizar o fluxo de trabalho para atender às suas necessidades.

Estrutura de um documento de fluxo de trabalho do YAML

O documento de fluxo de trabalho YAML que o Image Builder usa para realizar ações de compilação e teste de imagens está estruturado da forma a seguuir.

- Identificação do documento de fluxo de trabalho
- Parâmetros de entrada do documento de fluxo de trabalho
- Etapas do documento de fluxo de trabalho
- Saídas do documento de fluxo de trabalho

Identificação do documento de fluxo de trabalho

Identifica exclusivamente o fluxo de trabalho. Essa seção pode incluir os atributos a seguir.

Campo	Description	Tipo	Obrigatório
name	O nome do documento do fluxo de trabalho.	String	Não
descrição	A descrição do documento.	String	Não
schemaVersion	A versão do esquema do documento, atualmente 1.0.	String	Sim

Exemplo

```
---
name: sample-test-image
description: Workflow for a sample image, with extra configuration options exposed
 through workflow parameters.
schemaVersion: 1.0
```

Parâmetros de entrada do documento de fluxo de trabalho

Essa parte do documento do fluxo de trabalho define os parâmetros de entrada que o chamador pode especificar. Caso não tenha nenhum parâmetro, você poderá deixar essa seção de fora. Se você especificar parâmetros, cada parâmetro poderá incluir os atributos a seguir.

Campo	Description	Tipo	Obrigatório	Restrições
name		String	Sim	

Campo	Description	Tipo	Obrigatório	Restrições
	O nome do parâmetro.			
descrição	A descrição do parâmetro.	String	Não	
padrão	O valor padrão do parâmetro se não houver um valor. Se você não incluir um valor padrão na definição do parâmetro, o valor do parâmetro será obrigatório em runtime.	Corresponde ao tipo de dados do parâmetro.	Não	
type	O tipo de dado do parâmetro. Se você não incluir o tipo de dados na definição do parâmetro, o tipo de dados adotará por padrão um valor de string exigido em runtime.	String	Sim	O tipo de dados do parâmetro deve ser um dos seguintes: • `string` • `integer` • `boolean` • `stringList`

Exemplo

Especifique o parâmetro no documento de fluxo de trabalho.

```
parameters:
  - name: waitForActionAtEnd
    type: boolean
    default: true
    description: "Wait for an external action at the end of the workflow"
```

Use o valor do parâmetro no documento de fluxo de trabalho.

```
$.parameters.waitForActionAtEnd
```

Etapas do documento de fluxo de trabalho

Especifica ações com até 15 etapas para o fluxo de trabalho. As etapas são executadas na ordem em que são definidas no documento do fluxo de trabalho. Em caso de falha, uma reversão é executada na ordem inversa, começando com a etapa que falhou e retrocedendo no sentido das etapas anteriores.

Cada etapa pode indicar a saída de qualquer ação da etapa anterior. Isso é conhecido como encadeamento ou referência. Para se referir à saída de uma ação da etapa anterior, você pode usar um JSONPath seletor. Por exemplo:

```
$.stepOutputs.step-name.output-name
```

Para obter mais informações, consulte Use variáveis dinâmicas em seu documento de fluxo de trabalho.

> **Note**
>
> Mesmo que a etapa não tenha um atributo de saída, qualquer saída de uma ação de etapa será incluída em `stepOutput` para a etapa.

Cada etapa pode incluir os atributos a seguir.

Campo	Description	Tipo	Obrigatório	Valor padrão	Restrições
action	A ação do fluxo de trabalho que essa etapa executa.	String	Sim		Deve ser uma ação de etapa compatível com documentos de fluxo de trabalho do Image Builder.
`if`, seguido por um conjunto de declarações condicionais que modificam o operador `if`.	As declarações condicionais adicionam pontos de decisão de fluxo de controle ao corpo das etapas do fluxo de trabalho.	Dict	Não		O Image Builder é compatível com as seguintes declarações condicionais como modificadores do operador `if`: • Condições de ramificação e modificadores: `if`, `and`, `or`, `not`. As condições de

Campo	Description	Tipo	Obrigatório	Valor padrão	Restrições
					ramificação são especificadas individualmente em uma linha. • Operadores de comparação: `booleanEquals`, `numberEquals`, `numberGreaterThan`, `numberGreaterThanEquals`, `numberLessThan`, `numberLessThanEquals`, `stringEquals`.

Campo	Description	Tipo	Obrigatório	Valor padrão	Restrições
descrição	A descrição da etapa.	String	Não		Não é permitido ter strings vazias. Se incluída, o comprimento deve ser de 1 a 1.024 caracteres.
inputs	Contém os parâmetros de que a ação de etapa precisa para funcionar. Você pode especificar os valores-chave como valores estáticos ou com uma JSONPath variável que seja resolvida para o tipo de dados correto.	Dict	Sim		

Campo	Description	Tipo	Obrigatório	Valor padrão	Restrições
name	O nome da etapa. Esse nome deve ser exclusivo no documento de fluxo de trabalho.	String	Sim		Deve ter de 3 a 128 caracteres. Pode incluir caracteres alfanumér icos e _. Sem espaços.

Campo	Description	Tipo	Obrigatório	Valor padrão	Restrições
onFailure	Configurada da seguinte maneira a ação a ser tomada se a etapa falhar. Comportamento • `Abort`: resulta em falha na etapa, falha no fluxo de trabalho e não executa nenhuma etapa restante após a etapa que falhou. Se a reversão estiver ativada, ela começará com a etapa que falhou e continuará	String	Não	`Abort`	`Abort` | `Continue`

Campo	Description	Tipo	Obrigatório	Valor padrão	Restrições
	até reverter todas as etapas que a viabiliza m. • `Continue`: resulta em falha na etapa, mas continua executand o as etapas restantes após a etapa que falhou. Nesse caso, não há reversão.				

Campo	Description	Tipo	Obrigatório	Valor padrão	Restrições
rollbackEnabled	Configura se a etapa será revertida se ocorrer uma falha. Você pode usar um valor booleano estático ou uma JSONPath variável dinâmica que se resolve para um valor booleano.	Booleano	Não	`true`	`true` | `false` | ou uma JSONPath variável que é resolvida como verdadeira ou falsa.
timeoutSeconds	O tempo máximo, em segundos, de execução da etapa antes de falhar e tentar novamente, caso haja novas tentativas.	Inteiro	Não	Depende do padrão definido para a ação da etapa, se for o caso.	Não pode ser maior do que o tempo limite máximo da ação da etapa

Campo	Description	Tipo	Obrigatório	Valor padrão	Restrições
Espere segundos	O tempo, em segundos, durante o qual a execução da etapa será pausada.	Inteiro	Não	0	Não pode ser maior que timeoutSeconds da ação da etapa

Exemplo

```
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"
```

Saídas do documento de fluxo de trabalho

Define as saídas para o fluxo de trabalho. Cada saída é um par de valor e chave que especifica o nome da saída e o valor. Você pode usar saídas para exportar dados em runtime que os fluxos de trabalho subsequentes possam usar. Esta seção é opcional.

Cada saída que você define inclui os atributos a seguir.

Campo	Description	Tipo	Obrigatório
name	O nome da saída. O nome deve ser exclusivo em todos os fluxos de trabalho que você incluir no pipeline.	String	Sim
valor	O valor para a saída. O valor da string pode ser uma variável dinâmica, como um arquivo de saída de uma ação de etapa. Para obter mais informaçõ es, consulte Use variáveis dinâmicas em seu documento de fluxo de trabalho.	String	Sim

Exemplo

Crie uma ID de imagem de saída para o documento do fluxo de trabalho com a saída da etapa `createProdImage`.

```
outputs:
```

```
  - name: 'outputImageId'
    value: '$.stepOutputs.createProdImage.imageId'
```

Consulte a saída do fluxo de trabalho no próximo fluxo de trabalho.

```
$.workflowOutputs.outputImageId
```

Ações de etapa compatíveis com seu documento de fluxo de trabalho

Esta seção inclui detalhes das ações de etapa compatíveis com o Image Builder.

Termos usados nesta seção

AMI

Imagem de máquina da Amazon (AMI)

ARN

Nome do recurso da Amazon

Ações compatíveis

- ApplyImageConfigurations
- BootstrapInstanceForContainer
- CollectImageMetadata
- CollectImageScanFindings
- CreateImage
- DistributeImage
- ExecuteComponents
- ExecuteStateMachine
- LaunchInstance
- ModifyImageAttributes
- RegisterImage
- RunCommand
- RunSysPrep
- SanitizeInstance

- TerminateInstance
- WaitForAction
- WaitForSSMAgent

ApplyImageConfigurations

Essa ação de etapa aplica várias configurações e integrações distribuídas AMIs, como configurações de licença, configurações de modelo de lançamento, configurações de exportação do S3, configurações do EC2 Fast Launch e configurações de parâmetros do Systems Manager.
As configurações se aplicam a imagens distribuídas somente na conta de origem, exceto às configurações de parâmetros SSM, que podem ser aplicadas entre contas.

Tempo limite padrão: 360 minutos

Tempo limite máximo: 720 minutos

Reversão: não há reversão para essa ação de etapa.

Entradas: a tabela a seguir inclui entradas compatíveis com essa ação de etapa.

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
region	A região da imagem.	String	Sim		
licenseConfigurationArns	O ARN de configuração da licença para a imagem.	Array	Não		
launchTemplateConfigurations		Array	Não		
launchTemplateConfiguration	O ID do modelo de lançamento a	String	Sim, se `launchTemplateConf`		

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
s:launchTemplateId	ser aplicado à imagem.		`igurations` for especificado		
launchTemplateConfigurations: ID da conta	A conta do modelo de lançamento IDs a ser aplicada à imagem.	String	Não		
launchTemplateConfigurations:setDefaultVersion	A configuração da versão padrão do modelo de lançamento para a imagem.	Booleano	Não		
s3 ExportConfiguration		Array	Não		
ExportConfigurationS3: Nome da função	O nome da função de configuração de exportação do S3 para a imagem.	String	Sim, se `s3ExportConfiguration` for especificado		

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
s3: ExportConfiguration diskImageFormat	O formato de imagem de disco de configuração de exportação do S3 para a imagem.	String	Sim, se `s3ExportConfiguration` for especificado		Valores permitidos - VMDK|RAW|VHD
S3: balde S3 ExportConfiguration	O nome do bucket de configuração de exportação do S3 para a imagem.	String	Sim, se `s3ExportConfiguration` for especificado		
S3: prefixo S3 ExportConfiguration	O prefixo do bucket de configuração de exportação do S3 para a imagem.	String	Não		
fastLaunchConfigurations	A configuração do EC2 Fast Launch para a imagem.	Array	Não		
fastLaunchConfigurations:ativado	EC2 Início rápido enabled/disabled para a imagem.	Booleano	Sim, se `fastLaunchConfigurations` for especificado		

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
fastLaunchConfigurations: Configuração do snapshot	EC2 Início rápido enabled/disabled para a imagem.	Mapa	Não		
fastLaunchConfigurations: Configuração do snapshot: targetResourceCount	EC2 Contagem de recursos alvo do Fast Launch para a imagem.	Inteiro	Não		
fastLaunchConfigurations:maxParallelLaunches	EC2 Fast Launch: máximo de lançamentos paralelos para a imagem.	Inteiro	Não		
fastLaunchConfigurations: Modelo de lançamento			Não		

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
fastLaunchConfigurations: Modelo de lançamento: launchTemplateId	EC2 ID do modelo de lançamento do Fast Launch para a imagem.	String	Não		
fastLaunchConfigurations: Modelo de lançamento: launchTemplateName	EC2 Nome do modelo de lançamento do Fast Launch para a imagem.	String	Não		
fastLaunchConfigurations: Modelo de lançamento: launchTemplateVersion	EC2 Versão do modelo de lançamento do Fast Launch para a imagem.	String	Não		
ssmParameterConfigurations	A configuração do parâmetro SSM para a imagem.	Mapa	Não		

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
ssmParameterConfigurations: amiAccountId	O ID da conta da AMI do parâmetro SSM para a imagem.	String	Não		
ssmParameterConfigurations: Nome do parâmetro	O nome do parâmetro SSM da imagem.	String	Sim, se `ssmParameterConfigurations` for especificado		
ssmParameterConfigurations:Tipo de dados	O tipo de dados do parâmetro SSM para a imagem.	String	Não		Valores permitidos - text|aws:ec2:image)

Saídas: a tabela a seguir inclui saídas para essa ação de etapa.

Nome da saída	Description	Tipo
Imagens configuradas	Uma lista de imagens configuradas.	Array
Imagens configuradas: ID da conta	O ID da conta de destino da imagem distribuída.	String
Imagens configuradas: nome	O nome da AMI.	String
Imagens configuradas: AMIID	O ID da AMI da imagem distribuída.	String

Nome da saída	Description	Tipo
Imagens configuradas: data de início	Hora UTC quando a distribuição começou.	String
Imagens configuradas: data de parada	Hora UTC quando a distribuição foi concluída.	String
Configurar imagens: etapa	A etapa na qual a distribuição foi interrompida.	Concluído| AssociateLicensesRunning | UpdateLaunchTemplateRunning | | PutSsmParametersRunning | UpdateFastLaunchConfiguration ExportAmiQueued ExportAmiRunning
Imagens configuradas: região	Tom AWS da imagem distribuída	String
Imagens configuradas: status	Status de distribuição.	Concluído | Falha|Cancelado| TimedOut
Imagens configuradas: mensagem de erro	Mensagem de erro, se houver.	String

Exemplo

Especifique a ação de etapa no documento de fluxo de trabalho.

```
- name: ApplyImageConfigurations
  action: ApplyImageConfigurations
  onFailure: Abort
  inputs:
    distributedImages.$: $.stepOutputs.DistributeImageStep.distributedImages
```

Use a saída do valor da ação da etapa no documento do fluxo de trabalho.

```
$.stepOutputs.ApplyImageConfigurationsStep.configuredImages
```

BootstrapInstanceForContainer

Essa ação de etapa executa um script de serviço para fazer o bootstrap da instância com requisitos mínimos para executar fluxos de trabalho de contêiner. O Image Builder usa o sendCommand na API do Systems Manager para executar esse script. Para obter mais informações, consulte Comando Run do AWS Systems Manager.

> **Note**
>
> O script bootstrap instala os pacotes Docker AWS CLI e Docker, que são pré-requisitos para que o Image Builder crie contêineres do Docker com sucesso. Se você não incluir essa ação, a compilação da imagem poderá falhar.

Tempo limite padrão: 60 minutos

Tempo limite máximo: 720 minutos

Reversão: não há reversão para essa ação de etapa.

Entradas: a tabela a seguir inclui entradas compatíveis com essa ação de etapa.

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
instanceId	O ID da instância para o bootstrap.	String	Sim		Deve ser o ID da instância de saída da etapa do fluxo de trabalho que iniciou a instância para esse fluxo de trabalho.

Saídas: a tabela a seguir inclui saídas para essa ação de etapa.

Nome da saída	Description	Tipo
runCommandId	O ID do sendCommand do Systems Manager que executou o script de bootstrap na instância.	String
status	O status retornado do sendCommand do Systems Manager.	String
saída	A saída retornada do sendCommand do Systems Manager.	String

Exemplo

Especifique a ação de etapa no documento de fluxo de trabalho.

```
- name: ContainerBootstrapStep
  action: BootstrapInstanceForContainer
  onFailure: Abort
  inputs:
      instanceId.$: $.stepOutputs.LaunchStep.instanceId
```

Use a saída do valor da ação da etapa no documento do fluxo de trabalho.

```
$.stepOutputs.ContainerBootstrapStep.status
```

CollectImageMetadata

Essa ação de etapa só é válida para fluxos de trabalho de compilação.

EC2 O Image Builder executa o Agente AWS Systems Manager (Systems Manager) nas EC2 instâncias que ele executa para criar e testar sua imagem. O Image Builder coleta informações adicionais sobre a instância usada durante a fase de compilação com o Systems Manager Inventory. Essas informações incluem o nome e a versão do sistema operacional (SO), bem como a lista de pacotes e suas respectivas versões, conforme relatado pelo seu sistema operacional.

> **Note**
>
> Essa ação de etapa só funciona para imagens criadas AMIs.

Tempo limite padrão: 30 minutos

Tempo limite máximo: 720 minutos

Reversão: o Image Builder reverte todos os recursos do Systems Manager que foram criados durante essa etapa.

Entradas: a tabela a seguir inclui entradas compatíveis com essa ação de etapa.

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
instanceId	A instância de compilação à qual aplicar as configurações de metadados.	String	Sim		Deve ser o ID da instância de saída da etapa do fluxo de trabalho que iniciou a instância de compilação para esse fluxo de trabalho.

Saídas: a tabela a seguir inclui saídas para essa ação de etapa.

Nome da saída	Description	Tipo
osVersion	O nome e a versão do sistema operacional coletados da instância de compilação.	String

Nome da saída	Description	Tipo
associationId	O ID de associação do Systems Manager usado para coleta de inventário.	String

Exemplo

Especifique a ação de etapa no documento de fluxo de trabalho.

```
- name: CollectMetadataStep
  action: CollectImageMetadata
  onFailure: Abort
  inputs:
      instanceId: $.stepOutputs.LaunchStep.instanceId
```

Use a saída da ação da etapa no documento do fluxo de trabalho.

```
$.stepOutputs.CollectMetadataStep.osVersion
```

CollectImageScanFindings

Se o Amazon Inspector estiver habilitado para sua conta e a digitalização de imagens estiver habilitada para seu pipeline, essa ação de etapa coletará descobertas de digitalização de imagens relatadas pelo Amazon Inspector para sua instância de teste. Essa ação de etapa não está disponível para fluxos de trabalho de compilação.

Tempo limite padrão: 120 minutos

Tempo limite máximo: 720 minutos

Reversão: não há reversão para essa ação de etapa.

Entradas: a tabela a seguir inclui entradas compatíveis com essa ação de etapa.

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
instanceId	O ID da instância	String	Sim		Deve ser o ID da instância

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
	na qual a varredura foi executada.				de saída da etapa do fluxo de trabalho que iniciou a instância para esse fluxo de trabalho.

Saídas: a tabela a seguir inclui saídas para essa ação de etapa.

Nome da saída	Description	Tipo
runCommandId	O ID do sendCommand do Systems Manager que executou o script para coletar as descobertas.	String
status	O status retornado do sendCommand do Systems Manager.	String
saída	A saída retornada do sendCommand do Systems Manager.	String

Exemplo

Especifique a ação de etapa no documento de fluxo de trabalho.

```
- name: CollectFindingsStep
  action: CollectImageScanFindings
  onFailure: Abort
```

```
  inputs:
      instanceId.$: $.stepOutputs.LaunchStep.instanceId
```

Use a saída do valor da ação da etapa no documento do fluxo de trabalho.

```
$.stepOutputs.CollectFindingsStep.status
```

CreateImage

Essa ação de etapa cria uma imagem de uma instância em execução com a EC2 `CreateImage` API da Amazon. Durante o processo de criação, antes prosseguir, a ação de etapa aguarda conforme necessário para verificar se os recursos alcançaram o estado correto.

Tempo limite padrão: 720 minutos

Tempo limite máximo: 3 dias

Reversão: não há reversão para essa ação de etapa.

Entradas: a tabela a seguir inclui entradas compatíveis com essa ação de etapa.

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
instanceId	A instância com base na qual criar a nova imagem.	String	Sim		A instância do ID de instância fornecido deverá estar em um estado `running` quando essa etapa for iniciada.

Saídas: a tabela a seguir inclui saídas para essa ação de etapa.

Nome da saída	Description	Tipo
imageId	O ID de AMI da imagem que foi criada.	String

Exemplo

Especifique a ação de etapa no documento de fluxo de trabalho.

```
- name: CreateImageFromInstance
  action: CreateImage
  onFailure: Abort
  inputs:
      instanceId.$: "i-1234567890abcdef0"
```

Use a saída do valor da ação da etapa no documento do fluxo de trabalho.

```
$.stepOutputs.CreateImageFromInstance.imageId
```

DistributeImage

Essa ação de etapa distribui uma AMI para regiões e contas especificadas. Ele cria cópias da AMI nas regiões e contas de destino com base na configuração de distribuição fornecida nas solicitações de CreateImage ou CreateImagePipeline APIs ou nas configurações de distribuição personalizadas fornecidas no fluxo de trabalho para substituir as configurações na configuração de distribuição.

Tempo limite padrão: 360 minutos

Tempo limite máximo: 720 minutos

Reversão: não há reversão para essa ação de etapa.

Entradas: a tabela a seguir inclui entradas compatíveis com essa ação de etapa.

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
region	A lista de regiões para	String	Sim		Tamanho mínimo de 1.

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
	distribuir a imagem.				O comprimento máximo é 1.024.
name	O nome da configuração de distribuição.	String	Não		
descrição	As distribuições da configuração de distribuição.	String	Não		
targetAccountIds	Conta IDs para a qual distribuir a imagem.	Array	Não		
Etiquetas Ami	As tags da configuração de distribuição.	Mapa	Não		
kmsKeyId	Chaves KMS a serem aplicadas à imagem distribuída.	String	Não		

Saídas: a tabela a seguir inclui saídas para essa ação de etapa.

Nome da saída	Description	Tipo
Imagens distribuídas	Uma lista de imagens distribuídas	Array
Imagens distribuídas: Região	A AWS região da imagem distribuída.	String
Imagens distribuídas: nome	O nome da AMI.	String
Imagens distribuídas: Amiid	O ID da AMI da imagem distribuída.	String
Imagens distribuídas: ID da conta	O ID da conta de destino da imagem distribuída.	String
Imagens distribuídas: data de início	Hora UTC quando a distribuição começou.	String
Imagens distribuídas: data de parada	Hora UTC quando a distribuição foi concluída.	String
Imagens distribuídas: status	Status de distribuição.	Concluído | Falha|Cancelado| TimedOut
Imagens distribuídas: etapa	A etapa na qual a distribuição foi interrompida.	Concluído| CopyAmiRunning
Imagens distribuídas: mensagem de erro	Mensagem de erro, se houver.	String

Exemplo

Especifique a ação de etapa no documento de fluxo de trabalho.

```
- name: DistributeImage
  action: DistributeImage
  onFailure: Abort
  inputs:
```

```
    distributions:
      - region.$: "$.parameters.SourceRegion"
        description: "AMI distribution to source region"
        amiTags:
          DistributionTest: "SourceRegion"
          WorkflowStep: "DistributeToSourceRegion"
          BuildDate: "{{imagebuilder:buildDate:yyyyMMHHss}}"
          BuildVersion: "{{imagebuilder:buildVersion}}"
```

Use a saída do valor da ação da etapa no documento do fluxo de trabalho.

```
$.stepOutputs.DistributeImageStep.distributedImages
```

ExecuteComponents

Essa ação de etapa executa componentes especificados na fórmula da imagem atual que está sendo criada. Fluxos de trabalho de compilação executam componentes de compilação na instância de compilação. Os fluxos de trabalho de teste executam somente componentes de teste na instância de teste.

O Image Builder usa o sendCommand na API do Systems Manager para executar esse componentes. Para obter mais informações, consulte Comando Run do AWS Systems Manager.

Tempo limite padrão: 720 minutos

Tempo limite máximo: 1 dia

Reversão: não há reversão para essa ação de etapa.

Entradas: a tabela a seguir inclui entradas compatíveis com essa ação de etapa.

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
instanceId	O ID da instância na qual os component	String	Sim		Deve ser o ID da instância de saída da etapa

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
	es devem ser executados.				do fluxo de trabalho que iniciou a instância para esse fluxo de trabalho.

Saídas: a tabela a seguir inclui saídas para essa ação de etapa.

Nome da saída	Description	Tipo
runCommandId	O ID do sendCommand do Systems Manager que executou os componentes na instância.	String
status	O status retornado do sendCommand do Systems Manager.	String
saída	A saída retornada do sendCommand do Systems Manager.	String

Exemplo

Especifique a ação de etapa no documento de fluxo de trabalho.

```
- name: ExecComponentsStep
  action: ExecuteComponents
  onFailure: Abort
  inputs:
      instanceId: $.stepOutputs.LaunchStep.instanceId
```

Use a saída da ação da etapa no documento do fluxo de trabalho.

```
$.stepOutputs.ExecComponentsStep.status
```

ExecuteStateMachine

Essa ação de etapa inicia a execução de uma máquina de AWS Step Functions estado a partir de um fluxo de trabalho do Image Builder. O Image Builder usa a `StartExecution` API Step Functions para iniciar a máquina de estado e esperar que ela seja concluída. Isso é útil para integrar fluxos de trabalho complexos, validação de conformidade ou processos de certificação em seu pipeline de criação de imagens.

Para obter mais informações, consulte Saiba mais sobre máquinas de estado em Step Functions no AWS Step Functions Developer Guide.

Tempo limite padrão: 6 horas

Tempo limite máximo: 24 horas

Reversão: não há reversão para essa ação de etapa.

Entradas: a tabela a seguir inclui entradas compatíveis com essa ação de etapa.

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
stateMachineArn	O ARN da máquina de estados Step Functions a ser executada.	String	Sim		Deve ser um ARN de máquina de estado válido.
input	Dados de entrada JSON para fornecer à máquina de estado.	String	Não	{}	Deve ser uma string JSON válida, tamanho máximo: 16 KiB.

Saídas: a tabela a seguir inclui saídas para essa ação de etapa.

Nome da saída	Description	Tipo
executionArn	O ARN da execução da máquina de estado.	String
saída	A saída da execução da máquina de estado.	String

Permissões do IAM necessárias

Sua função de execução personalizada deve ter as seguintes permissões para usar essa ação de etapa:

Permitir ações

- `states:StartExecution`
- `states:DescribeExecution`

Especificar recursos

- `arn:aws:states:us-west-2:111122223333:stateMachine:state-machine-name`
- `arn:aws:states:us-west-2:111122223333:execution:state-machine-name:*`

Exemplo

Especifique a ação de etapa no documento de fluxo de trabalho.

```
- name: ValidateImageCompliance
  action: ExecuteStateMachine
  timeoutSeconds: 3600
  onFailure: Abort
  inputs:
    stateMachineArn: arn:aws:states:us-
west-2:111122223333:stateMachine:ImageComplianceValidation
    input: |
      {
```

```
        "imageId": "{{ $.stepOutputs.CreateImageFromInstance.imageId }}",
        "region": "us-west-2",
        "complianceLevel": "high",
        "requiredScans": ["cve", "benchmark", "configuration"]
      }
```

Use a saída do valor da ação da etapa no documento do fluxo de trabalho.

```
$.stepOutputs.ValidateImageCompliance.executionArn
```

LaunchInstance

Essa ação de etapa inicia uma instância na sua Conta da AWS e espera até que o agente do Systems Manager esteja em execução na instância antes de passar para a próxima etapa. A ação de inicialização usa configurações de sua fórmula e recursos de configuração de infraestrutura associados à sua imagem. Por exemplo, o tipo de instância a ser executada vem da configuração da infraestrutura. A saída é o ID da instância que foi executada.

A entrada `waitFor` configura a condição que satisfaz o requisito de conclusão da etapa.

Tempo limite padrão: 75 minutos

Tempo limite máximo: 720 minutos

Reversão: para instâncias de construção, a reversão executa a ação que você configurou em seu recurso de configuração de infraestrutura. Por padrão, as instâncias de compilação são encerradas se a criação da imagem falhar. No entanto, há uma definição na configuração da infraestrutura para manter a instância de compilação para solução de problemas.

Entradas: a tabela a seguir inclui entradas compatíveis com essa ação de etapa.

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
imageIdOverride	A imagem a ser usada para iniciar a instância	String	Não	Estágio de construção: imagem base da receita da imagem	Deve ser uma ID de AMI válida

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
				Estágio de teste: saída da AMI do estágio de construção	
instanceTypesOverride	O Image Builder testa cada tipo de instância na lista até encontrar uma que seja executada com sucesso	Lista de strings	Não	Tipos de instância especificados em sua configuração de infraestrutura	Devem ser tipos de instância válidos
waitFor	A condição a ser aguardada antes de concluir a etapa do fluxo de trabalho e passar para a próxima etapa	String	Sim		O Image Builder é compatível com `ssmAgent`.

Saídas: a tabela a seguir inclui saídas para essa ação de etapa.

Nome da saída	Description	Tipo
instanceId	O ID de instância da instância que foi executada.	String

Exemplo

Especifique a ação de etapa no documento de fluxo de trabalho.

```
- name: LaunchStep
  action: LaunchInstance
  onFailure: Abort
  inputs:
    waitFor: ssmAgent
```

Use a saída da ação da etapa no documento do fluxo de trabalho.

```
$.stepOutputs.LaunchStep.instanceId
```

ModifyImageAttributes

Essa ação de etapa modifica atributos de distribuídos AMIs, como permissões de execução e outros atributos da AMI. Ele opera com base nas AMIs que foram distribuídas para regiões e contas alvo.

Tempo limite padrão: 120 minutos

Tempo limite máximo: 180 minutos

Reversão: não há reversão para essa ação de etapa.

Entradas: a tabela a seguir inclui entradas compatíveis com essa ação de etapa.

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
region	A região da imagem.	String	Sim		
Permissão de lançamento			Não		
Permissão de lançamento: IDs de usuário	O usuário IDs a ser modificado nas	String	Não		

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
	permissões de lançamento da imagem.				
Permissão de lançamento: grupos de usuários	Os grupos de usuários a serem modificados nas permissões de lançamento da imagem.	String	Não		
Permissão de lançamento: Organization ARNS	A AWS organização ARNs a ser modificada nas permissões de lançamento da imagem.	String	Não		
Permissão de lançamento: organizationalUnitArns	A Unidade AWS Organizacional ARNs a ser modificada nas permissões de lançamento da imagem.	String	Não		

Saídas: a tabela a seguir inclui saídas para essa ação de etapa.

Nome da saída	Description	Tipo
Imagens modificadas	Uma lista de imagens modificadas	Array
Imagens modificadas: ID da conta	O ID da conta de destino da imagem distribuída.	String
Imagens modificadas: Nome	O nome da AMI.	String
Imagens modificadas: Amiid	O ID da AMI da imagem distribuída.	String
Imagens modificadas: data de início	Hora UTC quando a distribui ção começou.	String
Imagens modificadas: data de parada	Hora UTC quando a distribui ção foi concluída.	String
Imagens modificadas: etapa	A etapa na qual a distribuição foi interrompida.	Concluído| ModifyAmiRunning
Imagens modificadas: Região	A AWS região da imagem.	String
Imagens modificadas: status	Status de distribuição.	Concluído | Falha|Cancelado| TimedOut
Imagens modificadas: mensagem de erro	Mensagem de erro, se houver.	String

Exemplo

Especifique a ação de etapa no documento de fluxo de trabalho.

```
- name: ModifyImageAttributes
  action: ModifyImageAttributes
  onFailure: Abort
  inputs:
    distributedImages.$: $.stepOutputs.DistributeImageStep.distributedImages
```

Use a saída do valor da ação da etapa no documento do fluxo de trabalho.

```
$.stepOutputs.ModifyImageAttributesStep.modifiedImages
```

RegisterImage

Essa ação de etapa registra uma nova Amazon Machine Image (AMI) usando a EC2 RegisterImage API da Amazon. Ele permite que você crie uma AMI a partir de um snapshot existente ou de um conjunto de snapshots, especificando vários atributos de imagem.

Tempo limite padrão: 540 minutos

Tempo limite máximo: 720 minutos

Reversão: não há reversão para essa ação de etapa.

Entradas: a tabela a seguir inclui entradas compatíveis com essa ação de etapa.

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
arquitetura	A arquitetura da AMI.	String	Não		Valores válidos: i386, x86_64, arm64, x86_64_mac, arm64_mac
blockDeviceMapping	As entradas de mapeamento de dispositivos de bloco para a AMI.	Array	Não		
Modo de inicialização	O modo de inicialização da AMI.	String	Não		Valores válidos: legacy-bios, uefi, uefi-preferred
descrição	Uma descrição para a AMI.	String	Não		

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
enaSupport	Se a rede aprimorada com ENA está habilitada.	Booleano	Não		
Localização da imagem	A localização do manifesto da AMI.	String	Não		Necessário para suporte de S3 AMIs
Suporte ao IMDS	O nível de IMDSv2 suporte.	String	Não		Valores válidos: v2.0

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
includeSnapshotTags	Se deve incluir tags do primeiro instantâneo definido no mapeamento do dispositivo de bloco.	Booleano	Não	FALSE	Quando definidas como verdadeiras, as tags são incluídas da seguinte forma: • As tags do primeiro volume `SnapshotId` do EBS na `blockDeviceMapping` lista que contém a `SnapshotId` são mescladas com as tags de saída da AMI. • As tags AMI de saída têm precedência sobre as tags de snapshot com a mesma chave. • AWS tags reservadas (aquelas com chaves começando com `aws:`) são excluídas automaticamente. • Se vários volumes do EBS com

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
					`SnapshotId` forem definidos, somente as tags do primeiro volume do EBS na lista que contém um `SnapshotId` serão incluídas.
ID do kernel	O ID do kernel a ser usado.	String	Não		
RAM é Kid	O ID do disco RAM a ser usado.	String	Não		
rootDeviceName	O nome do dispositivo raiz.	String	Não		Exemplo: /dev/sda1
sriovNetSupport	Rede aprimorada com a interface Intel 82599 VF.	String	Não		
Suporte TPM	Suporte à versão TPM.	String	Não		Valores válidos: v2.0
Dados da UEFI	Dados UEFI codificados em Base64.	String	Não		
Tipo de virtualização	O tipo de virtualização.	String	Não		Valores válidos: hvm, paravirtual

Saídas: a tabela a seguir inclui saídas para essa ação de etapa.

Nome da saída	Description	Tipo
imageId	O ID da AMI da imagem registrada.	String

Permissões do IAM necessárias

Sua função de execução personalizada deve ter as seguintes permissões para usar essa ação de etapa:

Permitir ações

- `ec2:DescribeSnapshots`
- `ec2:CreateTags`

Exemplo

Especifique a ação de etapa no documento de fluxo de trabalho.

```
- name: RegisterNewImage
  action: RegisterImage
  onFailure: Abort
  inputs:
    architecture: "x86_64"
    bootMode: "uefi"
    blockDeviceMapping:
      - DeviceName: "/dev/sda1"
        Ebs:
          SnapshotId: "snap-1234567890abcdef0"
          VolumeSize: 100
          VolumeType: "gp3"
    rootDeviceName: "/dev/sda1"
    virtualizationType: "hvm"
```

Use a saída do valor da ação da etapa no documento do fluxo de trabalho.

```
$.stepOutputs.RegisterNewImage.imageId
```

Exemplo com uma SnapshotId de outra etapa e tags de snapshot incluídas na AMI gerada

```
- name: CreateSnapshot
  action: RunCommand
  onFailure: Abort
  inputs:
    instanceId: "i-1234567890abcdef0"
    documentName: "AWS-RunShellScript"
    parameters:
      commands:
        - "aws ec2 create-snapshot --volume-id vol-1234567890abcdef0 --description
 'Snapshot for AMI' --query 'SnapshotId' --output text"

- name: RegisterImageFromSnapshot
  action: RegisterImage
  onFailure: Abort
  inputs:
    architecture: "x86_64"
    bootMode: "uefi"
    blockDeviceMapping:
      - DeviceName: "/dev/sda1"
        Ebs:
          SnapshotId.$: "$.stepOutputs.CreateSnapshot.output[0]"
          VolumeSize: 100
          VolumeType: "gp3"
    includeSnapshotTags: true
    rootDeviceName: "/dev/sda1"
    virtualizationType: "hvm"
```

RunCommand

Essa ação de etapa executa um documento de comando para seu fluxo de trabalho. O Image Builder usa o sendCommand na API do Systems Manager para executá-lo em seu nome. Para obter mais informações, consulte Comando Run do AWS Systems Manager.

Tempo limite padrão: 720 minutos

Tempo limite máximo: 720 minutos

Reversão: não há reversão para essa ação de etapa.

Entradas: a tabela a seguir inclui entradas compatíveis com essa ação de etapa.

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
instanceId	O ID da instância na qual executar o documento de comando.	String	Sim		Deve ser o ID da instância de saída da etapa do fluxo de trabalho que iniciou a instância para esse fluxo de trabalho.
documentName	O nome do documento de comando do Systems Manager a ser executado.	String	Sim		
parameters	Uma lista de pares chave e valor para qualquer parâmetro exigido pelo documento de comando.	dictionary<string, list<string>>	Condicional		
documentVersion	A versão do documento de comando a ser executada.	String	Não	$DEFAULT	

Saídas: a tabela a seguir inclui saídas para essa ação de etapa.

Nome da saída	Description	Tipo
runCommandId	O ID do sendCommand do Systems Manager que executou o documento de comando na instância.	String
status	O status retornado do sendCommand do Systems Manager.	String
saída	A saída retornada do sendCommand do Systems Manager.	Lista de strings

Exemplo

Especifique a ação de etapa no documento de fluxo de trabalho.

```
- name: RunCommandDoc
  action: RunCommand
  onFailure: Abort
  inputs:
    documentName: SampleDocument
    parameters:
        osPlatform:
          - "linux"
    instanceId.$: $.stepOutputs.LaunchStep.instanceId
```

Use a saída do valor da ação da etapa no documento do fluxo de trabalho.

```
$.stepOutputs.RunCommandDoc.status
```

RunSysPrep

Essa ação de etapa usa o sendCommand na API do Systems Manager para executar o documento `AWSEC2-RunSysprep` para instâncias do Windows antes que a instância de compilação seja

encerrada para o snapshot. Essas ações seguem as AWS melhores práticas para endurecer e limpar a imagem.

Tempo limite padrão: 60 minutos

Tempo limite máximo: 720 minutos

Reversão: não há reversão para essa ação de etapa.

Entradas: a tabela a seguir inclui entradas compatíveis com essa ação de etapa.

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
instanceId	O ID da instância na qual executar o documento `AWSEC2-RunSysprep` .	String	Sim		Deve ser o ID da instância de saída da etapa do fluxo de trabalho que iniciou a instância para esse fluxo de trabalho.

Saídas: a tabela a seguir inclui saídas para essa ação de etapa.

Nome da saída	Description	Tipo
runCommandId	O ID do sendCommand do Systems Manager que executou o documento `AWSEC2-RunSysprep` na instância.	String

Nome da saída	Description	Tipo
status	O status retornado do sendCommand do Systems Manager.	String
saída	A saída retornada do sendCommand do Systems Manager.	String

Exemplo

Especifique a ação de etapa no documento de fluxo de trabalho.

```
- name: RunSysprep
  action: RunSysPrep
  onFailure: Abort
  inputs:
      instanceId.$: $.stepOutputs.LaunchStep.instanceId
```

Use a saída do valor da ação da etapa no documento do fluxo de trabalho.

```
$.stepOutputs.RunSysprep.status
```

SanitizeInstance

Essa ação de etapa executa o script de limpeza recomendado para instâncias Linux antes que a instância de compilação seja encerrada para o snapshot. O script de limpeza ajuda a garantir que a imagem final siga as práticas recomendadas de segurança e a remoção de qualquer artefato ou configuração de compilação que não deva ser transferida para seu snapshot. Para obter mais informações sobre o script, consulte Limpeza necessária após a construção. Essa ação não se aplica a imagens de contêiner.

O Image Builder usa o sendCommand na API do Systems Manager para executar esse script. Para obter mais informações, consulte Comando Run do AWS Systems Manager.

Tempo limite padrão: 60 minutos

Tempo limite máximo: 720 minutos

Reversão: não há reversão para essa ação de etapa.

Entradas: a tabela a seguir inclui entradas compatíveis com essa ação de etapa.

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
instanceId	O ID da instância a passar por limpeza.	String	Sim		Deve ser o ID da instância de saída da etapa do fluxo de trabalho que iniciou a instância para esse fluxo de trabalho.

Saídas: a tabela a seguir inclui saídas para essa ação de etapa.

Nome da saída	Description	Tipo
runCommandId	O ID do sendCommand do Systems Manager que executou o script de limpeza na instância.	String
status	O status retornado do sendCommand do Systems Manager.	String
saída	A saída retornada do sendCommand do Systems Manager.	String

Exemplo

Especifique a ação de etapa no documento de fluxo de trabalho.

```
- name: SanitizeStep
  action: SanitizeInstance
  onFailure: Abort
  inputs:
      instanceId: $.stepOutputs.LaunchStep.instanceId
```

Use a saída do valor da ação da etapa no documento do fluxo de trabalho.

```
$.stepOutputs.SanitizeStep.status
```

TerminateInstance

Essa ação de etapa encerra a instância com o ID da instância que é transmitido como entrada.

Tempo limite padrão: 30 minutos

Tempo limite máximo: 720 minutos

Reversão: não há reversão para essa ação de etapa.

Entradas: a tabela a seguir inclui entradas compatíveis com essa ação de etapa.

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
instanceId	Forneça o ID da instância a ser encerrada .	String	Sim		

Saídas: não há saídas para essa ação de etapa.

Exemplo

Especifique a ação de etapa no documento de fluxo de trabalho.

```
- name: TerminateInstance
  action: TerminateInstance
  onFailure: Continue
```

```
  inputs:
      instanceId.$: i-1234567890abcdef0
```

WaitForAction

Essa ação de etapa pausa o fluxo de trabalho em execução e aguarda o recebimento de uma ação externa da ação SendWorkflowStepAction da API do Image Builder. Esta etapa publica um EventBridge evento em seu ônibus de EventBridge eventos padrão com o tipo de `EC2 Image Builder Workflow Step Waiting` detalhe. A etapa também pode enviar uma notificação do SNS se você fornecer um ARN de tópico do SNS ou invocar uma função do Lambda de forma assíncrona se você fornecer um nome de função do Lambda.

Tempo limite padrão: 3 dias

Tempo limite máximo: 7 dias

Reversão: não há reversão para essa ação de etapa.

Entradas: a tabela a seguir inclui entradas compatíveis com essa ação de etapa.

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
snsTopicArn	Um ARN opcional de tópico do SNS para enviar uma notificação quando a etapa do fluxo de trabalho estiver pendente.	String	Não		
lambdaFunctionName	Um nome opcional ou ARN	String	Não		

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
	da função Lambda a ser invocada de forma assíncrona quando a etapa do fluxo de trabalho estiver pendente.				

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
payload	Cadeia de caracteres JSON usada como mensagem para SNS e carga útil para Lambda. Se fornecida, uma carga personalizada é agrupada por padrãomessage/payload, used for SNS and Lambda respectively. If not provided, generates default message/payload.	String	Não		Deve ser uma string JSON válida, no máximo 16 KiB

Saídas: a tabela a seguir inclui saídas para essa ação de etapa.

Nome da saída	Description	Tipo
action	A ação retornada pela ação de API SendWorkflowStepAction.	String (`RESUME` ou `STOP`)

Nome da saída	Description	Tipo
reason	O motivo da ação retornada.	String

Permissões do IAM necessárias

Sua função de execução personalizada deve ter as seguintes permissões para usar essa ação de etapa:

Permitir ações

- `lambda:InvokeFunction`

Especificar recursos

- `arn:aws:lambda:us-west-2:111122223333:function:function-name`
- `arn:aws:lambda:us-west-2:111122223333:function:*`

Exemplo

Especifique a ação da etapa no documento do fluxo de trabalho com a notificação do SNS.

```
- name: SendEventAndWait
  action: WaitForAction
  onFailure: Abort
  inputs:
    snsTopicArn: arn:aws:sns:us-west-2:111122223333:ExampleTopic
```

Especifique a ação da etapa no documento do fluxo de trabalho com a invocação da função Lambda.

```
- name: SendEventAndWaitWithLambda
  action: WaitForAction
  onFailure: Abort
  inputs:
    lambdaFunctionName: ExampleFunction
    payload: |
      {
        "imageId": "{{ $.stepOutputs.CreateImageFromInstance.imageId }}",
        "region": "us-west-2"
```

```
    }
```

Use a saída do valor da ação da etapa no documento do fluxo de trabalho.

```
$.stepOutputs.SendEventAndWait.reason
```

WaitForSSMAgent

Essa ação de etapa espera que uma EC2 instância se torne gerenciável AWS Systems Manager após os períodos esperados de falta de resposta. É particularmente importante para fluxos de trabalho com interrupções de instância conhecidas, como reinicializações do sistema, atualizações do sistema operacional ou operações específicas da plataforma que desconectam temporariamente a instância do SSM. O Image Builder monitora a instância até que ela recupere a conectividade SSM ou atinja o tempo limite.

Tempo limite padrão: 60 minutos

Tempo limite máximo: 180 minutos

Reversão: não há reversão para essa ação de etapa.

Entradas: a tabela a seguir inclui entradas compatíveis com essa ação de etapa.

Nome da entrada	Description	Tipo	Obrigatório	Padrão	Restrições
instanceId	O ID da instância para monitorar a conectividade do SSM.	String	Sim		Deve ser um ID de EC2 instância válido

Saídas: a tabela a seguir inclui saídas para essa ação de etapa.

Nome da saída	Description	Tipo
status	Status da conexão do SSM Agent.	String

Exemplo

Especifique a ação de etapa no documento de fluxo de trabalho.

```
- name: WaitForInstanceAfterReboot
  action: WaitForSSMAgent
  onFailure: Abort
  timeoutInSeconds: 900
  inputs:
    instanceId.$: $.stepOutputs.LaunchStep.instanceId
```

Use a saída do valor da ação da etapa no documento do fluxo de trabalho.

```
$.stepOutputs.WaitForInstanceAfterReboot.Status
```

Use variáveis dinâmicas em seu documento de fluxo de trabalho

Você pode usar variáveis dinâmicas em seus documentos de fluxo de trabalho para representar valores que variam em runtime para seu processo de criação de imagem. A interpolação de strings para variáveis dinâmicas permite incorporar JSONPath expressões em conteúdo estruturado, como cadeias de caracteres JSON. Isso é particularmente útil quando você precisa passar valores de tempo de execução em cargas complexas para ações escalonadas como ou. `ExecuteStateMachine` `WaitForAction`

Para usar a interpolação de strings para variáveis dinâmicas, coloque JSONPath expressões em colchetes duplos `"{{...}}"` dentro do conteúdo da string. Somente JSONPath expressões entre chaves duplas são processadas como variáveis. Qualquer JSONPath expressão que não esteja entre chaves duplas é tratada como conteúdo literal de seqüência de caracteres.

JSONPath sintaxe variável de fluxo de trabalho dinâmico

```
$.<document structure>.[<step name>.]<variable name>
```

Os valores das variáveis dinâmicas são representados como JSONPath seletores com nós estruturais que identificam exclusivamente a variável alvo. O primeiro nó após a raiz ($) se refere à estrutura do documento do fluxo de trabalho, como `stepOutputs`, ou no caso das variáveis de sistema do Image Builder, a `imageBuilder`. A lista a seguir contém nós de estrutura JSONPath de documentos de fluxo de trabalho compatíveis.

Nós de estrutura do documento

- parameters: os parâmetros do fluxo de trabalho
- stepOutputs: saídas de uma etapa no mesmo documento de fluxo de trabalho
- workflowOutputs: saídas de um documento de fluxo de trabalho que já foi executado
- imagebuilder: variáveis de sistema do Image Builder

Os nós da estrutura de documento `parameters` e `stepOutputs` incluem um nó opcional para o nome da etapa. Isso ajuda a garantir nomes exclusivos de variáveis em todas as etapas.

O nó final no JSONPath é o nome da variável de destino, como`instanceId`.

Cada etapa pode se referir à saída de qualquer ação da etapa anterior com essas variáveis JSONPath dinâmicas. Isso também é conhecido como encadeamento ou referência. Você pode usar a variável dinâmica a seguir para fazer referência à saída de uma ação da etapa anterior.

```
$.stepOutputs.step-name.output-name
```

> **Important**
>
> Quando um parâmetro de entrada se refere a uma variável dinâmica, o indicador de encadeamento (`.$`) deve ser anexado ao final do nome do parâmetro.

Exemplo 1: indicador de encadeamento de parâmetros de entrada

O exemplo a seguir mostra um parâmetro de entrada que usa interpolação de strings para resolver uma variável dinâmica no valor do parâmetro em tempo de execução.

```
  - name: ApplyTestComponents
    action: ExecuteComponents
    onFailure: Abort
    inputs:
```

```
      instanceId.$: "$.stepOutputs.LaunchTestInstance.instanceId"
```

Exemplo 2: interpolação de strings em variáveis dinâmicas

O exemplo a seguir demonstra como as variáveis dinâmicas usam a interpolação de strings para determinar valores em tempo de execução.

```
- name: ValidateImageConfiguration
  action: ExecuteStateMachine
  inputs:
    stateMachineArn: arn:aws:states:us-east-1:111122223333:stateMachine:ImageValidation
    input: |
      {
        "imageId": "{{ $.stepOutputs.CreateImageFromInstance.imageId }}",
        "region": "us-east-1",
        "buildDate": "{{ $.imagebuilder.dateTime }}",
        "instanceType": "{{ $.stepOutputs.LaunchStep.instanceType }}"
      }
```

Neste exemplo, as JSONPath expressões encapsuladas em chaves duplas são resolvidas em tempo de execução:

- `{{ $.stepOutputs.CreateImageFromInstance.imageId }}`- Resolve para o ID real da imagem a partir da etapa CreateImageFromInstance
- `{{ $.imagebuilder.dateTime }}`- Resolve para o timestamp de compilação atual. Consulte Usar variáveis de sistema do Image Builder para obter uma lista das variáveis de sistema do Image Builder que você pode usar.
- `{{ $.stepOutputs.LaunchStep.instanceType }}`- Resolve para o tipo de instância usado no LaunchStep

As sequências literais, como por exemplo, `"region": "us-east-1"` permanecem inalteradas.

> **Note**
>
> A interpolação de strings funciona com qualquer conteúdo de string em seu documento de fluxo de trabalho, incluindo cadeias de caracteres de várias linhas usando o operador YAML pipe (). | O requisito de colchete curvo atua como um mecanismo de escape para distinguir claramente entre JSONPath variáveis e conteúdo de texto literal.

Usar variáveis de sistema do Image Builder

O Image Builder fornece as seguintes variáveis de sistema que você pode usar em seu documento de fluxo de trabalho:

Nome da variável	Description	Tipo	Valor de exemplo
cloudWatchLogGrupo	O nome do grupo CloudWatch Logs para registros de saída. Formato: `/aws/imagebuilder/` *`<recipe-name>`*	String	`/aws/imagebuilder/` *`sampleImageRecipe`*
cloudWatchLogStream	O nome do fluxo de CloudWatch registros para registros de saída.	String	*`1.0.0/1`*
collectImageMetadata	A configuração que orienta o Image Builder a coletar ou não os metadados da instância.	Booleano	`true` | `false`
collectImageScanConclusões	O valor atual da configuração que permite ao Image Builder coletar resultados de varredura de imagens.	Booleano	`true` | `false`

Nome da variável	Description	Tipo	Valor de exemplo
imageBuildNumber	O número de versão da compilação da imagem.	Inteiro	*1*
imageId	O ID da AMI da imagem de base.	String	*ami-12345 67890abcdef1*
imageName	O nome da imagem.	String	*sampleImage*
imageType	O tipo de saída da imagem.	String	`AMI` | `Docker`
imageVersionNumber	O número de versão da imagem.	String	*1.0.0*
instanceProfileName	O nome da função do perfil de instância que o Image Builder usa para iniciar, criar e testar instâncias.	String	*SampleIma geBuilder InstanceP rofileRole*
platform	A plataforma do sistema operacional da imagem que foi compilada.	String	`Linux` | `Windows` | `MacOS`
s3Logs	Um objeto JSON que contém a configura ção dos logs do S3 que o Image Builder grava.	Objeto JSON	{'S3Logs': {'s3 BucketName ':' ', 's3*sample-bu cket* ': KeyPrefix '}} *ib-logs*

Nome da variável	Description	Tipo	Valor de exemplo
securityGroups	O grupo de segurança IDs que se aplica para criar e testar instâncias.	List [String]	*[sg-12345 67890abcd ef1, sg-111122 22333344445]*
sourceImageARN	O nome do recurso da Amazon (ARN) do recurso de imagem do Image Builder que o fluxo de trabalho usa para os estágios de compilação e teste.	String	arn: aws:image builder: ::image//*us-east-1 111122223 333 sampleIma ge 1.0.0/1*
subnetId	O ID da sub-rede na qual inicializar as instâncias de compilação e teste.	String	*subnet-12 34567890a bcdef1*
terminateInstanceOnFalha	O valor atual da configuração que direciona o Image Builder a encerrar a instância em caso de falha ou mantê-la para solução de problemas.	Booleano	`true` | `false`
workflowPhase	O estágio atual inicializado para a execução do fluxo de trabalho.	String	`Build` | `Test`

Nome da variável	Description	Tipo	Valor de exemplo
workingDirectory	O caminho para o diretório de trabalho.	String	`/tmp`

Usar declarações condicionais nas etapas do seu fluxo de trabalho

As declarações condicionais começam com o atributo `if` do documento de declaração. O objetivo final da declaração `if` é determinar se a ação da etapa deve ser executada ou ignorada. Se a declaração `if` for resolvida como `true`, a ação da etapa será executada. Se ela for resolvida para `false`, o Image Builder vai ignorar a ação da etapa e registrar o status `SKIPPED` da etapa no log.

A declaração `if` é compatível com instruções ramificadas (`and`, `or`) e modificadores condicionais (`not`). Ela também é compatível com os operadores de comparação a seguir que realizam comparações de valores (igual, menor que, maior que) com base nos tipos de dados que ela compara (string ou número).

Operadores de comparação compatíveis

- `booleanEquals`
- `numberEquals`
- `numberGreaterThan`
- `numberGreaterThanEquals`
- `numberLessThan`
- `numberLessThanEquals`
- `stringEquals`

Regras para declarações ramificadas e modificadores condicionais

As regras a seguir são aplicáveis a declarações ramificadas (`and`, `or`) e modificadores condicionais (`not`).

- Declarações ramificadas e modificadores condicionais devem aparecer individualmente em uma linha.
- As declarações ramificadas e os modificadores condicionais devem seguir as regras de nível.

- Só pode haver uma declaração no nível principal.
- Cada ramificação ou modificador secundário inicia um novo nível.

  Para obter mais informações sobre os níveis, consulte Níveis aninhados em declarações condicionais.
- Cada declaração ramificada deve ter pelo menos uma declaração condicional secundária, mas não mais do que dez.
- Os modificadores condicionais operam em apenas uma declaração condicional secundária.

Níveis aninhados em declarações condicionais

As declarações condicionais operam em vários níveis em uma seção própria. Por exemplo, o atributo de declaração `if` aparece no mesmo nível do nome da etapa e da ação em seu documento de fluxo de trabalho. Essa é a base da declaração condicional.

Você pode especificar até quatro níveis de declarações condicionais, mas somente uma declaração poderá aparecer no nível principal. Todas as outras declarações ramificadas, modificadores condicionais ou operadores condicionais serão recuados a partir desse nível, um recuo por nível.

O esquema a seguir mostra o número máximo de níveis aninhados para uma declaração condicional.

```
base:
  parent:
    - child (level 2)
        - child (level 3)
            child (level 4)
```

Atributo `if`

O atributo `if` especifica a declaração condicional como um atributo do documento. Esse é o nível zero.

Nível principal

Esse é o primeiro nível de aninhamento de declarações condicionais. Só pode haver uma declaração nesse nível. Se você não precisar de ramificações ou modificadores, isso pode ser um operador condicional sem declarações secundárias. Esse nível não usa notação de hífen, exceto para operadores condicionais.

Níveis secundários

Os níveis dois a quatro são considerados níveis secundários. As declarações secundárias podem incluir instruções ramificadas, modificadores condicionais ou operadores condicionais.

Exemplo: níveis aninhados

O exemplo a seguir mostra o número máximo de níveis em uma declaração condicional.

```
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"
```

Regras de aninhamento

- Cada ramificação ou modificador no nível secundário inicia um novo nível.
- Cada nível é indentado.
- É possível haver até quatro níveis, incluindo uma declaração, modificador ou operador no nível principal e até três níveis adicionais.

Exemplos de declaração condicional

Esse grupo de exemplos mostra diversos aspectos das declarações condicionais.

Ramificação: and

A instrução de ramificação `and` opera em uma lista de expressões que são secundárias à ramificação e que devem todas ser avaliadas como `true`. O Image Builder avalia as expressões na ordem em que elas aparecem na lista. Se alguma expressão for avaliada como `false`, o processamento será interrompido e a ramificação será considerada `false`.

O exemplo a seguir é avaliado como `true`, porque ambas as expressões são avaliadas como `true`.

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

Ramificação: or

A instrução de ramificação `or` opera em uma lista de expressões que são secundárias à ramificação e que devem ter ao menos uma avaliada como `true`. O Image Builder avalia as expressões na ordem em que elas aparecem na lista. Se alguma expressão for avaliada como `true`, o processamento será interrompido e a ramificação será considerada `true`.

Mesmo que a primeira expressão seja `false`, o exemplo a seguir é avaliado como `true`.

```
if:
  or:
    - stringEquals: 'test_string'
      value: 'test_string_not_equal'
    - numberEquals: 1
      value: 1
```

Modificador condicional: not

O modificador condicional `not` nega as declarações condicionais que são secundárias à ramificação.

O exemplo a seguir é avaliado `true` quando o modificador `not` negar a declaração condicional `stringEquals`.

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

Declaração condicional: booleanEquals

O operador de comparação `booleanEquals` compara os valores boolianos e responde com verdadeiro quando os valores boolianos têm correspondência exata.

O exemplo a seguir determina se `collectImageScanFindings` está habilitado.

```
if:
  - booleanEquals: true
    value: '$.imagebuilder.collectImageScanFindings'
```

Declaração condicional: stringEquals

O operador de comparação `stringEquals` compara duas strings e responde com verdadeiro quando as strings são uma correspondência exata. Se um dos valores não for uma string, o Image Builder o converterá em uma string antes da comparação.

O exemplo a seguir compara a variável de sistema da plataforma para determinar se o fluxo de trabalho está sendo executado em uma plataforma Linux.

```
if:
  - stringEquals: 'Linux'
    value: '$.imagebuilder.Platform'
```

Declaração condicional: numberEquals

O operador de comparação `numberEquals` compara dois números e responde com verdadeiro quando os números são iguais. Os números a serem comparados devem ter um dos seguintes formatos.

- Inteiro
- Float
- Uma string que corresponde ao seguinte padrão de regex: `^-?[0-9]+(\.)?[0-9]+$`.

No exemplo a seguir, todas as comparações são avaliadas como `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
```

Gerenciar a criação de imagens personalizadas no Image Builder por meio de um processo de pipeline repetível

Os pipelines de imagem do Image Builder fornecem uma estrutura de automação para criar e manter imagens personalizadas AMIs e de contêiner. Os pipelines oferecem as seguintes funcionalidades:

- Monte a imagem base, crie e teste componentes para personalização de imagens, configuração de infraestrutura e configurações de distribuição.
- Facilitar o agendamento de processos de manutenção automatizados usando o `Schedule builder` no assistente do console ou inserindo expressões cron para atualizações recorrentes de suas imagens.
- Ative a detecção de alterações na imagem base e nos componentes, para ignorar automaticamente as compilações programadas quando não houver alterações.
- Habilite a automação baseada em regras por meio da Amazon. EventBridge

> **Note**
>
> Para obter mais informações sobre o uso da EventBridge API para visualizar ou alterar regras, consulte a Amazon EventBridge API Reference. Para obter mais informações sobre o uso de EventBridge events comandos no AWS CLI para visualizar ou alterar regras, consulte eventos na Referência de AWS CLI Comandos.

Tópicos

- Definir as configurações de execução do pipeline para pipelines de imagem
- Indicar e visualizar detalhes do pipeline
- Criar e atualizar pipelines de imagens de AMI
- Criar e atualizar pipelines de imagens de contêiner
- Configurar fluxos de trabalho do pipeline de imagens no Image Builder
- Use EventBridge regras com pipelines do Image Builder

Definir as configurações de execução do pipeline para pipelines de imagem

Você pode escolher entre as seguintes opções para programar a execução do pipeline:

Construtor de horários

Use o Schedule Builder para configurar a execução automática e recorrente do pipeline. Você define quando e com que frequência seu pipeline deve ser executado (dia, hora e frequência). A programação padrão é toda semana com base no dia e na hora em que a programação foi criada (UTC).

Expressão Cron

Execute automaticamente o pipeline com uma expressão cron que especifica o cronograma. Para obter mais informações sobre a sintaxe cron usada pelo Image Builder, consulte. Usar expressões cron no Image Builder

Manual

O pipeline não funciona de acordo com um cronograma. No console, escolha Executar pipeline no menu Ações para executar o pipeline. A partir do AWS CLI, você pode correr`start-image-pipeline-execution`.

Configurações de dependência

Para compilações agendadas, você pode escolher se quer sempre executar dentro do cronograma ou ignorar a execução do pipeline, a menos que haja atualizações de dependência, como uma alteração na imagem base ou em um componente usado na receita.

Desativar automaticamente um pipeline com defeito

Para pipelines de imagem executados em um cronograma, você pode configurar o número máximo de falhas consecutivas a serem permitidas (até`10`) antes que o Image Builder desative automaticamente o pipeline.

Configurações de desativação automática

O Image Builder rastreia o número de falhas consecutivas nas execuções programadas do pipeline e realiza uma das seguintes ações sempre que é executado de acordo com um cronograma:

- Se a execução do pipeline for bem-sucedida, o número de falhas consecutivas será redefinido para zero.
- Se a execução do pipeline falhar, o Image Builder incrementa o número de falhas consecutivas. Se a contagem de falhas exceder o limite definido no`AutoDisablePolicy`, o Image Builder desativará o pipeline.

A contagem de falhas consecutivas também é redefinida para zero nas seguintes condições:

- O pipeline é executado manualmente e é bem-sucedido.
- A configuração do pipeline é atualizada.

Se a tubulação funcionar manualmente e falhar, a contagem permanecerá a mesma. A próxima execução programada continua aumentando de onde parou antes.

Configurar o registro do pipeline

Ao criar ou atualizar um pipeline de imagens, você pode configurar grupos de CloudWatch registros personalizados para criação de imagens e registros de pipeline. Certifique-se de que sua função de execução de pipeline personalizada tenha as seguintes permissões para criar e acessar os recursos do grupo de registros.

- registros: CreateLogGroup
- registros: CreateLogStream
- registros: PutLogEvents

Grupos de registros personalizados

Para usar grupos de registros personalizados para criação de imagens ou execução de pipeline, primeiro crie o grupo de CloudWatch registros em Registros. Para obter mais informações, consulte Criar um grupo de registros no Guia do usuário do Amazon CloudWatch Logs. Para obter orientação adicional sobre os requisitos de nomenclatura de grupos de registros, consulte CreateLogGroupa Referência da API Amazon CloudWatch Logs.

Console

Especifique o grupo de registros de imagem ou o grupo de registros de pipeline para seu pipeline na seção Configuração de registro em Configurações avançadas.

CLI

Especifique os seguintes campos no `logging-configuration` objeto se você usar um objeto JSON para configuração:

- `imageLogGroupName`
- `pipelineLogGroupName`

Para especificar todos os parâmetros diretamente na linha de comando, consulte create-image-pipelinea Referência de AWS CLI Comandos.

O Image Builder usa os seguintes grupos de registros padrão se você não especificar um grupo de registros personalizado:

Registros de criação de imagens

O Image Builder grava registros de compilação no seguinte grupo e stream de CloudWatch registros do Image Builder:

LogGroup: `/aws/imagebuilder/`*`ImageName`*

LogStream (x.x.x/x): *`ImageVersion/ImageBuildVersion`*

Registros de execução do pipeline

O Image Builder grava registros de execução do pipeline no seguinte grupo e stream de CloudWatch registros do Image Builder:

LogGroup: `/aws/imagebuilder/pipeline/`*`pipeline-name`*

LogStream: *`2025/09/01`* (a data de execução do pipeline em YYYY/MM/DD formato)

Cada registro do pipeline é anexado ao stream desse dia.

Execute um pipeline de imagens manualmente

Se você escolher a opção de agendamento manual para seu pipeline, ela só será executada quando você iniciar manualmente a compilação. Se você escolher uma das opções de agendamento automático, também poderá executá-la manualmente, entre as execuções programadas

regularmente. Por exemplo, se você tem um pipeline que normalmente é executado uma vez por mês, mas precisa incorporar uma atualização em um de seus componentes duas semanas após a execução anterior, você pode optar por executar o pipeline manualmente.

Console

Para executar seu pipeline na página de detalhes do pipeline no console do Image Builder, escolha Executar pipeline no menu Ações na parte superior da página. Uma mensagem de status é exibida na parte superior da página para notificá-lo de que seu pipeline foi iniciado ou se há um erro.

1. No canto superior esquerdo da página de detalhes do pipeline, escolha Executar pipeline no menu Ações.
2. Você pode ver o status atual do seu pipeline na guia Imagens de saída, na coluna Status.

AWS CLI

O exemplo a seguir mostra como usar o comando start-image-pipeline-execution no AWS CLI para iniciar um pipeline de imagem manualmente. Quando você executa esse comando, o pipeline cria e distribui uma nova imagem.

```
aws imagebuilder start-image-pipeline-execution --image-pipeline-arn
 arn:aws:imagebuilder:us-west-2:111122223333:image-pipeline/my-example-pipeline
```

Para ver quais recursos são criados quando o pipeline de compilação é executado, consulte Recursos criados.

Usar expressões cron no Image Builder

Use expressões cron no EC2 Image Builder para configurar uma janela de tempo para atualizar sua imagem com atualizações que se aplicam à imagem base e aos componentes do seu pipeline. A janela de tempo para a atualização do pipeline começa com a hora definida na expressão cron. Você pode definir a hora em sua expressão cron até o minuto. A construção do seu pipeline pode ser executada durante ou após o horário de início.

Às vezes, pode levar alguns segundos ou até um minuto para que sua compilação comece a ser executada.

> **Note**
>
> As expressões cron usam o fuso horário Tempo Universal Coordenado (UTC) por padrão, ou você pode especificar o fuso horário. Para obter mais informações sobre o horário UTC e encontrar a diferença do seu fuso horário, consulte Time Zone Abbreviations, Worldwide List (Abreviações de fuso horário: lista mundial).

Valores compatíveis com expressões cron no Image Builder

EC2 O Image Builder usa um formato cron que consiste em seis campos obrigatórios. Cada um é separado dos outros por um espaço intermediário, sem espaços à esquerda ou à direita:

```
<Minute> <Hour> <Day> <Month> <Day of the week> <Year>
```

A tabela a seguir mostra os valores compatíveis com as entradas cron necessárias.

Valores compatíveis com expressões cron

Campo	Valores	Curingas
Minuto	`0-59`	`, - * /`
Hora	`0-23`	`, - * /`
Dia	`1-31`	`, - * ? / L W`
Mês	`1-12` ou `jan-dec`	`, - * /`
Dia da semana	`1-7` ou `sun-sat`	`, - * ? L #`
Ano	`1970-2199`	`, - * /`

Curingas

A tabela a seguir descreve como o Image Builder usa curingas para expressões cron. Lembre-se de que pode levar até um minuto após o horário especificado para que a compilação seja iniciada.

Curingas compatíveis com expressões cron

Curinga	Descrição
,	A , (vírgula) curinga inclui valores adicionais. No campo mês, `jan,feb,mar` inclui janeiro, fevereiro e março.
-	O - (traço) curinga especifica intervalos. Em dia do campo mês, `1-15` inclui dias 1 a 15 do mês especificado.
*	O * (asterisco) curinga inclui todos os valores válidos para o campo.
?	O curinga `?` (interrogação) especifica que o valor do campo depende de outra configura ção. No caso do Dia e dos Day-of-week campos, quando um é especificado ou inclui todos os valores possíveis (`*`), o outro deve ser `?` a. Não é possível especificar ambos. Por exemplo, se você inserir a `7` no campo Dia (executar a construção no sétimo dia do mês), a Day-of-week posição deverá conter `?` a.
/	A / (barra) curinga especifica incrementos. Por exemplo, se você quiser que sua compilação seja executada a cada dois dias, insira `*/2` o campo dia.
L	O curinga L em qualquer um dos campos do dia especifica o último dia: 28 a 31 para o dia do mês, dependendo do mês, ou domingo, para o dia da semana.
W	O curinga W no campo Day-of-month especific a um dia da semana. No Day-of-month campo, se você inserir um número antes do`W`, isso significa que você deseja segmentar o dia

Curinga	Descrição
	da semana mais próximo desse dia. Por exemplo, se você especificar`3W`, deseja que sua compilação seja executada no dia da semana mais próximo ao terceiro dia do mês.
#	O # (hash) é permitido somente para o campo do dia da semana e deve ser seguido por um número entre 1 e 5. O número especific a quais semanas em um determinado mês se aplicam à execução da compilação. Por exemplo, se você quiser que sua compilação seja executada na segunda sexta-feira de cada mês, use `fri#2` para o campo dia da semana.

Restrições

- Não é possível especificar os campos Day-of-month e Day-of-week na mesma expressão cron. Se você especificar um valor ou `*` em um dos campos, deverá usar um `?` (ponto de interrogação) no outro.
- Não há suporte para expressões Cron que causam taxas mais rápidas que um minuto.

Exemplos de expressões cron no Image Builder

As expressões cron são inseridas de forma diferente no console do Image Builder e na API ou na CLI. Para ver exemplos, escolha a guia que se aplica a você.

Image Builder console

Os exemplos a seguir mostram expressões cron que você pode inserir no console para sua agenda de compilação. O horário UTC é especificado usando um relógio de 24 horas.

Executada diariamente às 10h (UTC)

```
0 10 * * ? *
```

Corra diariamente às 12h15 (UTC)

```
15 12 * * ? *
```

Executar diariamente à meia-noite (UTC)

```
0 0 * * ? *
```

Executada às 10h (UTC) todas as manhãs dos dias da semana

```
0 10 ? * 2-6 *
```

Executada às 18h (UTC) todas as noites da semana

```
0 18 ? * mon-fri *
```

Executar às 8h (UTC) todo o primeiro dia do mês

```
0 8 1 * ? *
```

Execute na segunda terça-feira de cada mês às 22h30 (UTC)

```
30 22 ? * tue#2 *
```

> **Tip**
>
> Se você não quiser que seu trabalho de pipeline se estenda até o dia seguinte enquanto estiver em execução, lembre-se de levar em consideração o tempo de sua compilação ao especificar o horário de início.

API/CLI

Os exemplos a seguir mostram expressões cron que você pode inserir usando comandos CLI ou solicitações de API para sua agenda de compilação. Somente a expressão cron é mostrada.

Executada diariamente às 10h (UTC)

```
cron(0 10 * * ? *)
```

Corra diariamente às 12h15 (UTC)

```
cron(15 12 * * ? *)
```

Executar diariamente à meia-noite (UTC)

```
cron(0 0 * * ? *)
```

Executada às 10h (UTC) todas as manhãs dos dias da semana

```
cron(0 10 ? * 2-6 *)
```

Executada às 18h (UTC) todas as noites da semana

```
cron(0 18 ? * mon-fri *)
```

Executar às 8h (UTC) todo o primeiro dia do mês

```
cron(0 8 1 * ? *)
```

Execute na segunda terça-feira de cada mês às 22h30 (UTC)

```
cron(30 22 ? * tue#2 *)
```

> **Tip**
>
> Se você não quiser que seu trabalho de pipeline se estenda até o dia seguinte enquanto estiver em execução, lembre-se de levar em consideração o tempo de sua compilação ao especificar o horário de início.

Expressões rate no Image Builder

Uma expressão rate começa quando você cria a regra de evento programado e, em seguida, a executa em sua programação definida.

As expressões rate têm dois campos obrigatórios. Os campos são separados por um espaço em branco.

Sintaxe

```
rate(value unit)
```

valor

Um número positivo.

unidade

A unidade de tempo. Diferentes unidades são necessárias para valores de 1, como `minute`, e valores acima de 1, como `minutes`.

Valores válidos: minuto | minutos | hora | horas | dia | dias

Restrições

Se o valor for igual a 1, a unidade deverá ser singular. Da mesma forma, para valores maiores do que 1, a unidade deve ser plural. Por exemplo, `rate(1 hours)` e `rate(5 hour)` não são válidos, mas `rate(1 hour)` e `rate(5 hours)` são.

Indicar e visualizar detalhes do pipeline

Esta seção descreve as várias maneiras pelas quais você pode encontrar informações e visualizar detalhes dos pipelines de EC2 imagem do Image Builder.

Detalhes do pipeline

- Listar pipelines de imagens do AWS CLI
- Obtenha detalhes do pipeline de imagens no AWS CLI

Listar pipelines de imagens do AWS CLI

O exemplo a seguir mostra como usar o list-image-pipelines comando no AWS CLI para listar todos os seus pipelines de imagem.

```
aws imagebuilder list-image-pipelines
```

Obtenha detalhes do pipeline de imagens no AWS CLI

O exemplo a seguir mostra como usar o get-image-pipeline comando no AWS CLI para obter os detalhes sobre um pipeline de imagem por meio de seu ARN.

```
aws imagebuilder get-image-pipeline --image-pipeline-arn arn:aws:imagebuilder:us-
west-2:123456789012:image-pipeline/my-example-pipeline
```

Criar e atualizar pipelines de imagens de AMI

Você pode instalar, configurar e gerenciar pipelines de imagem da AMI a partir do console do Image Builder, por meio da API Image Builder SDKs ou do AWS CLI. No console, você pode usar o assistente de console Create Image Pipeline para guiá-lo pelas etapas a seguir.

- Especifique os detalhes do pipeline, como nome, descrição e tags de recursos.
- Configure o cronograma do pipeline e os padrões de registro. Para a execução programada do pipeline, você pode definir o número de falhas consecutivas que são permitidas antes que o Image Builder desative o pipeline.
- Selecione uma receita de imagem da AMI que inclua uma imagem base das imagens gerenciadas pela Amazon de início rápido, imagens que você criou ou que foram compartilhadas com você ou imagens que você assina por meio do. AWS Marketplace A receita também inclui componentes que realizam as seguintes tarefas nas EC2 instâncias que o Image Builder usa para criar sua imagem:
  - Adicionar e remover software
  - Personalizar configurações e scripts
  - Executar os testes selecionados
- Especifique fluxos de trabalho para configurar as etapas de compilação e teste de imagens executadas por seu pipeline.
- Defina a configuração da infraestrutura para seu pipeline com configurações padrão ou configurações que você mesmo define. A configuração inclui o tipo de instância e o par de chaves a serem usados em suas configurações de imagem, segurança e rede, armazenamento de log e configurações de solução de problemas e notificações do SNS.

  Esta é uma etapa opcional. Se você não definir a configuração sozinho, o Image Builder vai usar configurações padrão para sua configuração de infraestrutura.
- Defina as configurações de distribuição para entregar suas imagens às regiões e contas de destino da AWS . Você pode especificar uma chave KMS para criptografia, configurar o compartilhamento da AMI ou a configuração da licença ou configurar um modelo de execução para a AMIs que você distribui.

  Esta é uma etapa opcional. Se você não definir uma configuração, o Image Builder usará a nomenclatura padrão para sua AMI de saída e distribuirá a AMI para a região de origem. A região de origem é a região na qual você executa o pipeline.

Para obter mais informações e um step-by-step tutorial sobre como usar o assistente de console Create image pipeline com valores padrão, quando fornecidos, consulteTutorial: criar um pipeline de imagens com a AMI de saída por meio do assistente do console do Image Builder.

Conteúdo

- Crie um pipeline de imagens da AMI a partir do AWS CLI
- Atualizar pipelines de imagens de AMI com o console
- Atualize os pipelines de imagens da AMI a partir do AWS CLI

Crie um pipeline de imagens da AMI a partir do AWS CLI

Para criar um pipeline de imagem a partir do AWS CLI, execute o create-image-pipeline comando com as opções de configuração que se aplicam ao seu pipeline. Você tem a opção de criar um arquivo JSON que contém toda a configuração do pipeline ou especificar a configuração em tempo de execução. Esta seção usa o método do arquivo de configuração JSON para simplificar o comando.

A frequência com que seu pipeline cria uma nova imagem para incorporar quaisquer atualizações pendentes da imagem base e dos componentes depende do `schedule` que você configurou. Cada `schedule` tem os atributos a seguir:

- `scheduleExpression`: define o cronograma de execução do pipeline para avaliar `pipelineExecutionStartCondition` e determinar se ele deve iniciar uma compilação. A programação é configurada com expressões cron. Para obter mais informações sobre como formatar uma expressão cron no Image Builder, consulte Usar expressões cron no Image Builder.
- `pipelineExecutionStartCondition`: determina se seu pipeline deve iniciar a compilação. Os valores válidos são:
  - `EXPRESSION_MATCH_ONLY`: seu pipeline compilará uma imagem sempre que a expressão cron corresponder à hora atual.
  - `EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE`: seu pipeline não iniciará uma nova compilação de imagem, a menos que haja alterações pendentes em sua imagem base ou em seus componentes.

Quando você executa o create-image-pipeline comando no AWS CLI, muitos dos recursos de configuração são opcionais. No entanto, alguns dos recursos têm requisitos condicionais,

dependendo do tipo de imagem que o pipeline cria. Os seguintes identificadores de recursos são necessários para os pipelines de imagem da AMI:

- ARN da fórmula da imagem
- ARN de configuração de infraestrutura

Exemplo: criar uma imagem do Windows 2019

Este exemplo configura um pipeline que está programado para ser executado uma vez por semana no domingo. O arquivo de configuração mostrado na primeira etapa usa os recursos existentes para a receita de imagem, infraestrutura e configuração de distribuição, junto com outras configurações para criar uma imagem do Windows 2019.

1. Crie um arquivo de configuração (opcional)

   Este exemplo usa um arquivo de configuração chamado `create-image-pipeline.json` para definir as configurações em um só lugar. Como alternativa, você pode usar as opções da linha de comando ao executar o comando para especificar todos os detalhes que são mostrados aqui no arquivo de configuração.

```
{
 "name": "ExampleWindows2019Pipeline",
 "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**
>
> - É necessário incluir a notação `file://` no início do caminho do arquivo JSON.
> - O caminho para o arquivo JSON deve seguir a convenção apropriada para o sistema operacional de base no qual você está executando o comando. Por exemplo, o Windows usa a barra invertida (\) para se referir ao caminho do diretório, enquanto o Linux e o macOS usam a barra (/).

2. Execute o comando para criar uma imagem

   Este exemplo usa o arquivo de configuração criado na primeira etapa como entrada para o `create-image-pipeline` comando. Como alternativa, você pode especificar as configurações e os recursos do seu pipeline diretamente ao executar o comando. Para obter mais informações, consulte create-image-pipeline na AWS CLI Referência.

   ```
   aws imagebuilder create-image-pipeline --cli-input-json file://create-image-
   pipeline.json
   ```

Atualizar pipelines de imagens de AMI com o console

Depois de criar um pipeline de imagem do Image Builder para sua imagem da AMI, você pode fazer alterações na configuração da infraestrutura e nas configurações de distribuição no console do Image Builder.

Para atualizar um pipeline de imagens com uma nova fórmula de imagem, você deve usar o AWS CLI. Para obter mais informações, consulte Atualize os pipelines de imagens da AMI a partir do AWS CLI neste guia.

Escolha um pipeline existente do Image Builder

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.

2. Para ver uma lista dos pipelines de imagem criados em sua conta, escolha Pipelines de imagem no painel de navegação.

> **Note**
>
> A lista de pipelines de imagem inclui um indicador do tipo de imagem de saída criada pelo pipeline: AMI ou Docker.

3. Para ver detalhes ou editar um pipeline, escolha o link do nome do pipeline. Isto abre a visão detalhada do pipeline.

> **Note**
>
> Você também pode selecionar a caixa ao lado de Nome do pipeline, depois escolher Exibir detalhes.

Detalhes do pipeline

A página de detalhes do pipeline inclui as seguintes seções:

Resumo

A seção na parte superior da página resume os principais detalhes do pipeline que são visíveis com qualquer uma das guias de detalhes abertas. Os detalhes exibidos nesta seção são editáveis somente nas respectivas guias de detalhes.

Guias de detalhes

- Imagens de saída: mostra as imagens de saída que o pipeline produziu.
- Fórmula de imagem: mostra detalhes da fórmula. Após criar uma fórmula, você não poderá editá-la. Você deve criar uma nova versão da fórmula na página fórmulas de imagens no console do Image Builder ou usando os comandos do Image Builder no AWS CLI. Para obter mais informações, consulte Gerenciar fórmulas no Image Builder.
- Configuração da infraestrutura: mostra informações editáveis para configurar sua infraestrutura de pipeline de compilação.
- Configurações de distribuição: mostra informações editáveis para a distribuição da AMI.
- EventBridge regras — Para o Event Bus selecionado, mostra EventBridge as regras que têm como alvo o pipeline atual. Inclui as ações Criar barramento de eventos e Criar regras vinculadas ao

EventBridge console. Para obter mais informações sobre esta guia, consulte EventBridge Regras de uso.

Edite a configuração da infraestrutura para seu pipeline

A configuração da infraestrutura inclui os seguintes detalhes que você pode editar depois de criar o pipeline:

- A descrição da configuração da infraestrutura.
- Associe o perfil do IAM para associar ao perfil de instância.
- AWS infraestrutura, incluindo o tipo de instância e um tópico de SNS para notificações.
- VPC, sub-rede e grupos de segurança.
- Configurações de solução de problemas, incluindo Encerrar instância em caso de falha, o par de chaves para conexão e um local opcional do bucket S3 para os logs da instância.

Para editar a configuração da infraestrutura na página de detalhes do pipeline, siga estas etapas:

1. Escolha a guia Configuração da infraestrutura.
2. Escolha Editar no canto superior direito do painel Detalhes da configuração.
3. Quando quiser salvar as atualizações feitas na configuração da infraestrutura, escolha Salvar alterações.

Edite as configurações de distribuição do seu pipeline

As configurações de distribuição incluem os seguintes detalhes que você pode editar depois de criar o pipeline:

- A descrição desta configuração de distribuição.
- Configurações de região para as regiões em que você distribui sua imagem. A Região 1 usa como padrão a Região onde o pipeline foi criado. Você pode adicionar regiões para distribuição com o botão Adicionar região e remover todas as regiões, exceto a região 1.

  As configurações de região incluem:

  - Região de destino
  - O nome da AMI de saída

- Inicie permissões e contas com as quais compartilhá-las
- Licenças associadas (configurações de licença associada)

> **Note**
>
> As configurações do License Manager não serão replicadas entre AWS regiões que devem ser ativadas em sua conta, por exemplo, entre as regiões `ap-east-1` (Hong Kong) e `me-south-1` (Bahrein).

Para editar suas configurações de distribuição na página de detalhes do pipeline, siga estas etapas:

1. Escolha a guia Configurações de distribuição.
2. Escolha Editar no canto superior direito do painel Detalhes da distribuição.
3. Quando estiver pronto para salvar suas atualizações, escolha Salvar alterações.

Editar o cronograma de compilação do seu pipeline

A página Editar pipeline inclui os seguintes detalhes que você pode editar depois de criar o pipeline:

- A descrição do seu pipeline.
- Coleta avançada de metadados. Isto é ativado por padrão. Para desativar, desmarque a caixa de seleção Ativar coleta avançada de metadados.
- O cronograma de compilação do seu pipeline. Você pode alterar suas opções de agendamento e todas as configurações aqui.

Para editar seu pipeline na página de detalhes do pipeline, siga estas etapas:

1. No canto superior direito da página de detalhes do pipeline, escolha Ações e, em seguida, Editar pipeline.
2. Quando estiver pronto para salvar suas atualizações, escolha Salvar alterações.

> **Note**
>
> Para obter mais informações sobre o cronograma da sua compilação usando expressões cron, consulte Usar expressões cron no Image Builder.

Atualize os pipelines de imagens da AMI a partir do AWS CLI

Você pode atualizar um pipeline de imagens de AMI usando um arquivo JSON como entrada para o comando update-image-pipeline na AWS CLI. Para configurar o arquivo JSON, você deve ter Amazon Resource Names (ARNs) para referenciar os seguintes recursos existentes:

- Pipeline de imagens a ser atualizado
- Fórmula da imagem
- Configuração de infraestrutura
- Configurações de distribuição

Você pode atualizar um pipeline de imagem da AMI com o update-image-pipeline comando da AWS CLI seguinte forma:

> **Note**
>
> UpdateImagePipeline não oferece suporte a atualizações seletivas para o pipeline. Você deve especificar todas as propriedades necessárias na solicitação de atualização, não apenas as propriedades que foram alteradas.

1. Criar um arquivo JSON de entrada da CLI

   Use a sua ferramenta favorita de edição de arquivos para criar um arquivo JSON com as chaves a seguir, além de valores válidos para seu ambiente. Este exemplo usa um arquivo denominado `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**
>
> - É necessário incluir a notação `file://` no início do caminho do arquivo JSON.
> - O caminho para o arquivo JSON deve seguir a convenção apropriada para o sistema operacional de base no qual você está executando o comando. Por exemplo, o Windows usa a barra invertida (\) para se referir ao caminho do diretório, enquanto o Linux e o macOS usam a barra (/).

2. Execute o seguinte comando, usando o arquivo que você criou como entrada.

```
aws imagebuilder update-image-pipeline --cli-input-json file://update-image-
pipeline.json
```

Criar e atualizar pipelines de imagens de contêiner

Você pode instalar, configurar e gerenciar pipelines de imagens de contêiner usando o console do Image Builder, a API do Image Builder ou os comandos imagebuilder na AWS CLI. O assistente do console Create Image Pipeline fornece artefatos iniciais e orienta você nas etapas para:

- Selecionar uma imagem base a partir dos repositórios de imagens gerenciadas de início rápido, Amazon ECR ou Docker Hub
- Adicionar e remover software
- Personalizar configurações e scripts
- Executar os testes selecionados
- Criar um Dockerfile usando variáveis de tempo de compilação pré-configuradas.
- Distribua imagens para AWS regiões

Para obter mais informações e um step-by-step tutorial sobre como usar o assistente de console Create image pipeline, consulteTutorial: criar um pipeline de imagens com a imagem de contêiner do Docker de saída por meio do assistente do console do Image Builder.

Conteúdo

- Crie um pipeline de imagens de contêiner a partir do AWS CLI
- Atualizar um pipeline de imagens de contêiner com o console
- Atualize os pipelines de imagens do contêiner a partir do AWS CLI

Crie um pipeline de imagens de contêiner a partir do AWS CLI

Para criar um pipeline de imagem a partir do AWS CLI, execute o create-image-pipeline comando com as opções de configuração que se aplicam ao seu pipeline. Você tem a opção de criar um arquivo JSON que contém toda a configuração do pipeline ou especificar a configuração em tempo de execução. Esta seção usa o método do arquivo de configuração JSON para simplificar o comando.

A frequência com que seu pipeline cria uma nova imagem para incorporar quaisquer atualizações pendentes da imagem base e dos componentes depende do `schedule` que você configurou. Cada `schedule` tem os atributos a seguir:

- `scheduleExpression`: define o cronograma de execução do pipeline para avaliar `pipelineExecutionStartCondition` e determinar se ele deve iniciar uma compilação. A programação é configurada com expressões cron. Para obter mais informações sobre como formatar uma expressão cron no Image Builder, consulte Usar expressões cron no Image Builder.
- `pipelineExecutionStartCondition`: determina se seu pipeline deve iniciar a compilação. Os valores válidos são:

- `EXPRESSION_MATCH_ONLY`: seu pipeline compilará uma imagem sempre que a expressão cron corresponder à hora atual.
- `EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE`: seu pipeline não iniciará uma nova compilação de imagem, a menos que haja alterações pendentes em sua imagem base ou em seus componentes.

Quando você executa o create-image-pipeline comando no AWS CLI, muitos dos recursos de configuração são opcionais. No entanto, alguns dos recursos têm requisitos condicionais, dependendo do tipo de imagem que o pipeline cria. Os seguintes recursos são necessários para os pipelines de imagens de contêiner:

- ARN da fórmula do contêiner
- ARN de configuração de infraestrutura

Se você não incluir um recurso de configuração de distribuição ao executar o comando create-image-pipeline, a imagem de saída será armazenada no repositório ECR que você especifica como repositório de destino em sua fórmula do contêiner na região em que você executa o comando. Se você incluir um recurso de configuração de distribuição para seu pipeline, o repositório de destino que você especificou para a primeira região na distribuição será usado.

1. Criar um arquivo JSON de entrada da CLI

   Use a sua ferramenta favorita de edição de arquivos para criar um arquivo JSON com as chaves a seguir, além de valores válidos para seu ambiente. Este exemplo usa um arquivo denominado `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**
>
> - É necessário incluir a notação `file://` no início do caminho do arquivo JSON.
> - O caminho para o arquivo JSON deve seguir a convenção apropriada para o sistema operacional de base no qual você está executando o comando. Por exemplo, o Windows usa a barra invertida (\) para se referir ao caminho do diretório, enquanto o Linux e o macOS usam a barra (/).

2. Execute o seguinte comando, usando o arquivo que você criou como entrada.

```
aws imagebuilder create-image-pipeline --cli-input-json file://create-image-
pipeline.json
```

Atualizar um pipeline de imagens de contêiner com o console

Depois de criar um pipeline de imagem de contêiner do Image Builder para sua imagem do Docker, você pode fazer alterações na configuração da infraestrutura e nas configurações de distribuição no console do Image Builder.

Para atualizar um pipeline de imagem de contêiner com uma nova fórmula de contêiner, você deve usar a AWS CLI. Para obter mais informações, consulte Atualize os pipelines de imagens do contêiner a partir do AWS CLI neste guia.

Escolha um pipeline de imagem do Docker existente do Image Builder

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.

2. Para ver uma lista dos pipelines de imagem criados em sua conta, escolha Pipelines de imagem no painel de navegação.

> **Note**
>
> A lista de pipelines de imagem inclui um indicador do tipo de imagem de saída criada pelo pipeline: AMI ou Docker.

3. Para ver detalhes ou editar um pipeline, escolha o link do nome do pipeline. Isto abre a visão detalhada do pipeline.

> **Note**
>
> Você também pode selecionar a caixa ao lado de Nome do pipeline, depois escolher Exibir detalhes.

Detalhes do pipeline

A página de detalhes do pipeline do EC2 Image Builder inclui as seguintes seções:

Resumo

A seção na parte superior da página resume os principais detalhes do pipeline que são visíveis com qualquer uma das guias de detalhes abertas. Os detalhes exibidos nesta seção são editáveis somente nas respectivas guias de detalhes.

Guias de detalhes

- Imagens de saída: mostra as imagens de saída que o pipeline produziu.
- Fórmula do contêiner: mostra os detalhes da fórmula. Após criar uma fórmula, você não poderá editá-la. Você deve criar uma nova versão da fórmula na página Fórmulas do contêiner. Para obter mais informações, consulte Criar uma nova versão de fórmula de contêiner.
- Configuração da infraestrutura: mostra informações editáveis para configurar sua infraestrutura de pipeline de compilação.
- Configurações de distribuição: mostra informações editáveis para distribuição da imagem do Docker.
- EventBridge regras — Para o Event Bus selecionado, mostra EventBridge as regras que têm como alvo o pipeline atual. Inclui as ações Criar barramento de eventos e Criar regras vinculadas ao

EventBridge console. Para obter mais informações sobre esta guia, consulte EventBridge Regras de uso.

Edite a configuração da infraestrutura para seu pipeline

A configuração da infraestrutura inclui os seguintes detalhes que você pode editar depois de criar o pipeline:

- A descrição da sua configuração da infraestrutura.
- Associe o perfil do IAM para associar ao perfil de instância.
- AWS infraestrutura, incluindo o tipo de instância e um tópico de SNS para notificações.
- VPC, sub-rede e grupos de segurança.
- Configurações de solução de problemas, incluindo Encerrar instância em caso de falha, o par de chaves para conexão e um local opcional do bucket S3 para os logs da instância.

Para editar a configuração da infraestrutura na página de detalhes do pipeline, siga estas etapas:

1. Escolha a guia Configuração da infraestrutura.
2. Escolha Editar no canto superior direito do painel Detalhes da configuração.
3. Quando quiser salvar as atualizações feitas na configuração da infraestrutura, escolha Salvar alterações.

Edite as configurações de distribuição do seu pipeline

As configurações de distribuição incluem os seguintes detalhes que você pode editar depois de criar o pipeline:

- A descrição de suas configurações de distribuição.
- Configurações de região para as regiões em que você distribui sua imagem. A Região 1 usa como padrão a Região onde o pipeline foi criado. Você pode adicionar regiões para distribuição com o botão Adicionar região e remover todas as regiões, exceto a região 1.

  As configurações de região incluem:

  - Região de destino
  - O serviço é padronizado como “ECR” e não é editável.

- Nome do repositório: o nome do seu repositório de destino (sem incluir a localização do Amazon ECR). Por exemplo, o nome do repositório com a localização seria semelhante ao seguinte padrão:

  `<account-id>.dkr.ecr.<region>.amazonaws.com/<repository-name>`

  > **Note**
  >
  > Se você alterar o nome do repositório, somente as imagens criadas após a alteração do nome serão adicionadas ao novo nome. Todas as imagens anteriores que seu pipeline criou permanecem no repositório original.

Para editar suas configurações de distribuição na página de detalhes do pipeline, siga estas etapas:

1. Escolha a guia Configurações de distribuição.
2. Escolha Editar no canto superior direito do painel Detalhes da distribuição.
3. Quando estiver pronto para salvar as atualizações feitas nas configurações de distribuição, escolha Salvar alterações.

Editar o cronograma de compilação do seu pipeline

A página Editar pipeline inclui os seguintes detalhes que você pode editar depois de criar o pipeline:

- A descrição do seu pipeline.
- Coleta avançada de metadados. Isto é ativado por padrão. Para desativar, desmarque a caixa de seleção Ativar coleta avançada de metadados.
- O cronograma de compilação do seu pipeline. Você pode alterar suas opções de cronograma e todas as configurações nesta seção.

Para editar seu pipeline na página de detalhes do pipeline, siga estas etapas:

1. No canto superior direito da página de detalhes do pipeline, escolha Ações e, em seguida, Editar pipeline.
2. Quando estiver pronto para salvar suas atualizações, escolha Salvar alterações.

> **Note**
>
> Para obter mais informações sobre o cronograma da sua compilação usando expressões cron, consulte Usar expressões cron no Image Builder.

Atualize os pipelines de imagens do contêiner a partir do AWS CLI

Você pode atualizar um pipeline de imagens de contêiner usando um arquivo JSON como entrada para o comando update-image-pipeline na AWS CLI. Para configurar o arquivo JSON, você deve ter Amazon Resource Names (ARNs) para referenciar os seguintes recursos existentes:

- Pipeline de imagens a ser atualizado
- Fórmula do contêiner
- Configuração de infraestrutura
- Configurações de distribuição (se incluídas no pipeline atual)

> **Note**
>
> Se o recurso de configurações de distribuição estiver incluído, o repositório ECR especificado como repositório de destino nas configurações de distribuição da região em que o comando é executado (Região 1) terá precedência sobre o repositório de destino especificado na fórmula do contêiner.

Siga estas etapas para atualizar um pipeline de imagem de contêiner usando o comando update-image-pipeline na AWS CLI:

> **Note**
>
> UpdateImagePipeline não oferece suporte a atualizações seletivas para o pipeline. Você deve especificar todas as propriedades necessárias na solicitação de atualização, não apenas as propriedades que foram alteradas.

1. Criar um arquivo JSON de entrada da CLI

   Use a sua ferramenta favorita de edição de arquivos para criar um arquivo JSON com as chaves a seguir, além de valores válidos para seu ambiente. Este exemplo usa um arquivo denominado `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**
   >
   > - É necessário incluir a notação `file://` no início do caminho do arquivo JSON.
   > - O caminho para o arquivo JSON deve seguir a convenção apropriada para o sistema operacional de base no qual você está executando o comando. Por exemplo, o Windows usa a barra invertida (\) para se referir ao caminho do diretório, enquanto o Linux e o macOS usam a barra (/).

2. Execute o seguinte comando, usando o arquivo que você criou como entrada.

```
aws imagebuilder update-image-pipeline --cli-input-json file://update-image-pipeline.json
```

Configurar fluxos de trabalho do pipeline de imagens no Image Builder

Com fluxos de trabalho de imagem, você pode personalizar os fluxos de trabalho que seu pipeline executa para compilar e testar imagens de acordo com suas necessidades. Os fluxos de trabalho que você define são executados dentro do contexto do framework de fluxo de trabalho do Image Builder. Para obter mais informações sobre os estágios que compõem o framework de fluxo de trabalho, consulte Gerenciar fluxos de trabalho de compilação e teste para imagens do Image Builder.

Criar fluxos de trabalho

Crie fluxos de trabalho executados durante o estágio `Build` do framework do fluxo de trabalho. Você só pode especificar um fluxo de trabalho de compilação para seu pipeline. Como alternativa, você pode ignorar totalmente a compilação para configurar um pipeline somente para testes.

Testar fluxos de trabalho

Teste fluxos de trabalho executados durante o estágio `Test` do framework do fluxo de trabalho. Você pode especificar até dez fluxos de trabalho de teste para seu pipeline. Também é possível ignorar totalmente os testes se quiser apenas que seu pipeline seja compilado.

Definir grupos de teste para fluxos de trabalho de teste

Os fluxos de trabalho de teste são definidos nos grupos de teste. Você pode executar até dez fluxos de trabalho de teste para seu pipeline. Você decide se deseja executar os fluxos de trabalho de teste em uma ordem específica ou executar o maior número possível ao mesmo tempo. A forma como eles são executados dependerá de como você definir seus grupos de teste. Os cenários a seguir demonstram várias maneiras de definir seus fluxos de trabalho de teste.

> **Note**
>
> Se você usar o console para criar fluxos de trabalho, recomendamos que reserve um tempo para planejar como deseja executar seus fluxos de trabalho de teste antes de defini-los. No console, você pode adicionar ou remover fluxos de trabalho e grupos de teste, mas não pode reordená-los.

Cenário 1: executar um fluxo de trabalho de teste por vez

Para executar todos os seus fluxos de trabalho de teste um por um, você pode configurar até dez grupos de teste, cada um com um único fluxo de trabalho de teste. Os grupos de teste são executados um por vez, na ordem em que você os adicionar ao seu pipeline. Essa é uma forma de garantir que seus fluxos de trabalho de teste sejam executados um por vez um uma ordem específica.

Cenário 2: executar vários fluxos de trabalho de teste ao mesmo tempo

Se a ordem não importar e você quiser executar o maior número possível de fluxos de trabalho de teste ao mesmo tempo, é possível configurar um único grupo de teste e colocar o número máximo de fluxos de trabalho de teste nele. O Image Builder iniciará até cinco fluxos de trabalho de teste ao mesmo tempo e iniciará fluxos de trabalho de teste adicionais conforme os outros forem concluídos. Se seu objetivo for executar seus fluxos de trabalho de teste com o máximo de rapidez possível, essa é uma maneira de fazer isso.

Cenário 3: misturar e combinar

Se você tiver um cenário misto, com alguns fluxos de trabalho de teste que possam ser executados ao mesmo tempo e alguns que devam ser executados um por vez, será possível configurar seus grupos de teste para atingir esse objetivo. O único limite de como você configura seus grupos de teste é o número máximo de fluxos de trabalho de teste que podem ser executados em seu pipeline.

Definir parâmetros de fluxo de trabalho em um pipeline do Image Builder com o console

Os parâmetros do fluxo de trabalho funcionam da mesma forma para criar fluxos de trabalho e testar fluxos de trabalho. Ao criar ou atualizar um pipeline, você seleciona compilar e testar os fluxos de trabalho que deseja incluir. Se você tiver definido parâmetros no documento do fluxo de trabalho

para um fluxo de trabalho selecionado, o Image Builder os exibirá no painel Parâmetros. O painel ficará oculto para fluxos de trabalho que não tenham parâmetros definidos.

Cada parâmetro exibirá os seguintes atributos que seu documento de fluxo de trabalho definiu:

- Nome do parâmetro (não editável): o nome do parâmetro.
- Tipo (não editável): o tipo de dados para o valor do parâmetro.
- Valor: o valor do parâmetro. Você pode editar o valor do parâmetro a fim de configurá-lo para seu funil.

Especificar o perfil de serviço do IAM que o Image Builder usará para executar ações de fluxo de trabalho

Para executar fluxos de trabalho de imagem, o Image Builder precisa de permissão para realizar ações de fluxo de trabalho. Veja a seguir como especificar um perfil vinculado a serviço AWSServiceRoleForImageBuilder ou especificar seu próprio perfil personalizado para acesso ao serviço.

- Console: na Etapa 3: definir o processo de criação de imagem do assistente de pipeline, selecione o perfil vinculado a serviço ou seu próprio perfil personalizado na lista de perfis do IAM no painel Acesso ao serviço.
- API Image Builder — Na solicitação de CreateImageação, especifique a função vinculada ao serviço ou sua própria função personalizada como o valor do parâmetro. `executionRole`

Para saber mais sobre como criar uma função de serviço, consulte Criação de uma função para delegar permissões a um AWS serviço no Guia do AWS Identity and Access Management usuário.

Use EventBridge regras com pipelines do Image Builder

Eventos de uma ampla variedade de serviços AWS e de parceiros são transmitidos para os ônibus de EventBridge eventos da Amazon quase em tempo real. Você também pode gerar eventos personalizados e enviar eventos de seus próprios aplicativos para EventBridge o. Os barramentos de eventos usam regras para determinar para onde rotear os dados do evento.

Os pipelines do Image Builder estão disponíveis como destinos de EventBridge regras, o que significa que você pode executar um pipeline do Image Builder com base nas regras criadas para responder a eventos no barramento ou em uma programação.

Para obter um resumo dos eventos gerados pelo sistema para os quais o Image Builder envia EventBridge, consulteMensagens de eventos que o Image Builder envia.

> **Note**
>
> Os barramentos de eventos são específicos para uma região. A regra e o alvo devem estar localizados na mesma região.

Conteúdo

- EventBridge termos
- Veja EventBridge as regras para seu pipeline do Image Builder
- Use EventBridge regras para agendar a construção de um pipeline

EventBridge termos

Esta seção contém um resumo dos termos para ajudar você a entender como se EventBridge integra aos seus pipelines do Image Builder.

Event

Descreve uma alteração em um ambiente que pode afetar um ou mais recursos do aplicativo. O ambiente pode ser um AWS ambiente, um serviço ou aplicativo de parceiro SaaS ou um de seus aplicativos ou serviços. Também é possível configurar eventos agendados em uma linha do tempo.

Barramento de eventos

Um pipeline que recebe dados de eventos de aplicativos e serviços.

Origem

O serviço ou aplicativo que enviou o evento para o ônibus do evento.

Alvo

Um recurso ou endpoint que é EventBridge invocado quando corresponde a uma regra, entregando dados do evento para o destino.

Regra

Uma regra corresponde a eventos de entrada e encaminha-os para os destinos para processamento. Uma única regra pode enviar um evento para vários destinos, que podem ser executados paralelamente. As regras são baseadas em um padrão de evento ou em uma programação.

Padrão

Um padrão de evento define a estrutura do evento e os campos aos quais uma regra corresponde para iniciar a ação de destino.

Programação

As regras de agendamento executam uma ação em um cronograma, como executar um pipeline do Image Builder para atualizar uma imagem trimestralmente. Há dois tipos de expressões de agendamento:

- Expressões Cron: combine critérios de agendamento específicos usando a sintaxe cron que pode delinear critérios simples; por exemplo, execução semanal em um dia específico. Você também pode estabelecer critérios mais complexos, como correr trimestralmente no quinto dia do mês, entre 2h e 4h.
- Expressões de taxa: especifique um intervalo regular quando o alvo é invocado, como a cada 12 horas.

Veja EventBridge as regras para seu pipeline do Image Builder

A guia de EventBridge regras na página de detalhes dos pipelines de imagem do Image Builder exibe os barramentos de EventBridge eventos aos quais sua conta tem acesso e as regras do barramento de eventos selecionado que se aplicam ao pipeline atual. Essa guia também se vincula diretamente ao EventBridge console para criar novos recursos.

Ações vinculadas ao EventBridge console

- Crie um barramento de eventos
- Criar regra

Para saber mais sobre isso EventBridge, consulte os tópicos a seguir no Guia EventBridge do usuário da Amazon.

- O que é a Amazon EventBridge
- Ônibus para EventBridge eventos da Amazon
- EventBridge Eventos da Amazon
- EventBridge Regras da Amazon

Use EventBridge regras para agendar a construção de um pipeline

Neste exemplo, criamos uma nova regra de agendamento para o barramento de eventos padrão, usando uma expressão de taxa. A regra neste exemplo gera um evento no barramento de eventos a cada 90 dias. O evento inicia a criação de um pipeline para atualizar a imagem.

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. Para ver uma lista dos pipelines de imagem criados em sua conta, escolha Pipelines de imagem no painel de navegação.

   > **Note**
   >
   > A lista de pipelines de imagem inclui um indicador do tipo de imagem de saída criada pelo pipeline: AMI ou Docker.

3. Para ver detalhes ou editar um pipeline, escolha o link do nome do pipeline. Isto abre a visão detalhada do pipeline.

   > **Note**
   >
   > Você também pode selecionar a caixa ao lado de Nome do pipeline, depois escolher Exibir detalhes.

4. Abra a guia de EventBridge regras.
5. Mantenha o barramento de eventos padrão pré-selecionado no painel Barramento de eventos.
6. Escolha Criar regra. Isso leva você à página Criar regra no EventBridge console da Amazon.
7. Insira um nome e uma descrição para a regra. O nome da regra deve ser exclusivo no barramento de eventos da região selecionada.
8. No painel Definir padrão, escolha a opção Programação. Isso expande o painel, com a taxa fixa de todas as opções selecionadas.
9. Insira 90 na primeira caixa e selecione Dias na lista suspensa.

10. Execute as seguintes ações no painel Selecionar alvos:

    a. Selecione na `EC2 Image Builder` lista suspensa Alvo.

    b. Para aplicar a regra a um pipeline do Image Builder, selecione o pipeline de destino na lista suspensa Pipeline de imagens.

    c. EventBridge precisa de permissão para iniciar uma construção para o pipeline selecionado. Neste exemplo, mantenha a opção padrão de Criar uma nova função para esse recurso específico.

    d. Escolha Adicionar destino.

11. Selecione Criar

> **Note**
>
> Para saber mais sobre as configurações das regras de expressão de taxas que não são abordadas neste exemplo, consulte Expressões de taxa no Guia EventBridge do usuário da Amazon.

Integrar produtos e serviços no Image Builder

EC2 O Image Builder se AWS Marketplace integra a outros aplicativos para ajudá-lo a criar imagens de máquina personalizadas robustas Serviços da AWS e seguras.

Produtos

As receitas do Image Builder podem incorporar produtos de imagem AWS Marketplace e componentes gerenciados do Image Builder para fornecer funcionalidade especializada de criação e teste, conforme a seguir.

- AWS Marketplace produtos de imagem — Use um produto de imagem AWS Marketplace como imagem base em sua receita para atender aos padrões organizacionais, como o CIS Hardening. Ao criar uma fórmula no console do Image Builder, você pode escolher entre suas assinaturas existentes ou pesquisar um produto específico em. AWS Marketplace Ao criar uma fórmula por meio da API, da CLI ou do SDK do Image Builder, você pode especificar o nome do recurso da Amazon (ARN) do produto de imagem para usar como imagem de base.
- Componentes do Image Builder: os componentes que você especifica em suas fórmulas podem realizar ações de compilação e teste para, por exemplo, instalar software ou realizar validação de conformidade. Alguns produtos de imagem que você assina no AWS Marketplace podem incluir um componente complementar que pode ser usado em suas fórmulas. As imagens CIS Hardened incluem um AWSTOE componente correspondente que você pode usar em sua receita para aplicar as diretrizes do CIS Benchmarks Nível 1 para sua configuração.

> **Note**
>
> Para obter mais informações sobre como criar produtos relacionados à conformidade, consulte Produtos de conformidade para suas imagens do Image Builder.

Serviços

O Image Builder se integra Serviços da AWS ao seguinte para fornecer métricas, registros e monitoramento detalhados de eventos. Essas informações ajudam você a monitorar sua atividade, solucionar problemas de compilação de imagem e criar automações com base em notificações de eventos.

- AWS Organizations— AWS Organizations permite que você aplique Políticas de Controle de Serviços (SCP) nas contas da sua organização. Você pode criar, gerenciar, ativar e desativar políticas individuais. Semelhante a todos os outros AWS artefatos e serviços, o Image Builder respeita as políticas definidas em. AWS Organizations AWS fornece um modelo SCPs para cenários comuns, como impor restrições às contas dos membros para iniciar instâncias somente com aprovação. AMIs
- AWS CloudTrail— Monitore os eventos do Image Builder que são enviados para CloudTrail o. Para obter mais informações sobre a CloudTrail integração com o Image Builder, consulteRegistre as chamadas da API Image Builder usando CloudTrail.

  Para saber mais sobre CloudTrail, inclusive como ativá-lo e encontrar seus arquivos de log, consulte o Guia AWS CloudTrail do usuário.
- Amazon CloudWatch Logs — Monitore, armazene e acesse seus arquivos de log do Image Builder com CloudWatch. Como opção, você pode salvar seus registros em um bucket do S3. Para saber mais sobre a CloudWatch integração com o Image Builder, consulteMonitore os registros do Image Builder com o Amazon CloudWatch Logs.

  Para obter mais informações sobre CloudWatch registros, consulte O que é Amazon CloudWatch Logs? no Guia do usuário do Amazon CloudWatch Logs.
- Amazon Elastic Container Registry (Amazon ECR): o Amazon ECR é um serviço de registro de imagem de contêiner gerenciado da AWS que é seguro, escalável e confiável. As imagens de contêiner que você cria com o Image Builder são armazenadas no Amazon ECR em sua região de origem (onde sua compilação é executada) e em qualquer região em que você distribui a imagem de contêiner. Para obter mais informações sobre o Amazon ECR, consulte o Guia do usuário do Amazon Elastic Container Registry.
- Amazon EventBridge — Conecte-se a um fluxo de dados de eventos em tempo real das atividades do Image Builder em sua conta. Para obter mais informações sobre EventBridge, consulte O que é a Amazon EventBridge? no Guia do EventBridge usuário da Amazon.
- Amazon Inspector — Descubra vulnerabilidades em suas configurações de software e rede com varreduras automáticas para a instância de EC2 teste que o Image Builder lança e crie uma nova imagem. O Image Builder salva as descobertas do seu recurso de imagem de saída para que você possa investigar e corrigir após o encerramento da instância de teste. Para obter mais informações sobre verificações e definição de preço, consulte What is Amazon Inspector? no Guia do usuário do Amazon Inspector.

O Amazon Inspector também pode escanear seus repositórios ECR se você configurar o escaneamento aprimorado. Para obter mais informações sobre o Amazon Inspector, consulte Verificação de imagens de contêiner com o Amazon Inspector no Guia do usuário do Amazon Inspector.

> **Note**
>
> O Amazon Inspector é um atributo pago.

- AWS License Manager: você pode anexar uma licença autogerenciada do License Manager a uma AMI de saída durante o processo de distribuição. A licença que você especifica para a região de destino já deve existir nessa região. Consulte mais informações sobre licenças autogerenciadas em Self-managed licenses in License Manager.
- AWS Marketplace: veja uma lista de suas assinaturas de produtos atuais no AWS Marketplace e pesquise produtos de imagem diretamente no Image Builder. Você também pode usar um produto de imagem que você assinou como imagem base para uma fórmula do Image Builder. Para obter mais informações sobre o gerenciamento de AWS Marketplace assinaturas, consulte Comprar produtos no Guia do AWS Marketplace comprador.
- AWS Resource Access Manager (AWS RAM) — Com AWS RAM, você pode compartilhar recursos com qualquer um Conta da AWS ou por meio de AWS Organizations. Se você tiver vários Contas da AWS, poderá criar recursos de forma centralizada e usá-los AWS RAM para compartilhar esses recursos com outras contas. EC2 O Image Builder permite o compartilhamento dos seguintes recursos: componentes, imagens e receitas de imagens. Para obter mais informações sobre AWS RAM, consulte o Guia AWS Resource Access Manager do usuário. Para obter informações sobre o compartilhamento de recursos do Image Builder, consulte Compartilhe recursos do Image Builder com AWS RAM.
- Amazon Simple Notification Service (Amazon SNS): se configurado, publique mensagens detalhadas sobre o status da sua imagem em um tópico do SNS que você assina. Para ter mais informações sobre o Amazon SNS, consulte O que é o Amazon SNS? no Guia do desenvolvedor do Amazon Simple Notification Service.

Tópicos de integrações de produtos e serviços

- EventBridge Integração da Amazon no Image Builder
- Integração do Amazon Inspector no Image Builder
- AWS Marketplace integração no Image Builder

- Integração do Amazon SNS no Image Builder
- Produtos de conformidade para suas imagens do Image Builder

EventBridge Integração da Amazon no Image Builder

EventBridge O Amazon é um serviço de ônibus de eventos sem servidor que você pode usar para conectar seu aplicativo Image Builder a dados relacionados de outros. Serviços da AWS Em EventBridge, uma regra combina os eventos recebidos e os envia aos destinos para processamento. Uma única regra pode enviar um evento para vários destinos, que podem ser executados paralelamente.

Com EventBridge, você pode automatizar Serviços da AWS e responder automaticamente a eventos do sistema, como problemas de disponibilidade de aplicativos ou alterações de recursos. Os eventos de Serviços da AWS são entregues quase EventBridge em tempo real. Você pode configurar regras que reajam aos eventos recebidos para iniciar ações. Por exemplo, enviar um evento para uma função Lambda quando o status de uma EC2 instância muda de pendente para em execução. Eles são chamados de padrões. Para criar uma regra com base em um padrão de evento, consulte Criação de EventBridge regras da Amazon que reagem a eventos no Guia EventBridge do usuário da Amazon.

As ações que podem ser automaticamente acionadas incluem as seguintes:

- Invocar uma função AWS Lambda
- Invoque o EC2 comando Amazon Run
- Retransmitir o evento para o Amazon Kinesis Data Streams
- Ativar uma máquina de AWS Step Functions estado
- Notificação de um tópico do Amazon SNS ou de uma fila do Amazon SQS

Você também pode configurar regras de agendamento para que o barramento de eventos padrão execute uma ação em intervalos regulares, como executar um pipeline do Image Builder para atualizar uma imagem trimestralmente. Há dois tipos de expressões de agendamento:

- expressões cron: o exemplo a seguir de uma expressão cron agenda uma tarefa para ser executada todos os dias ao meio-dia UTC+0:

```
cron(0 12 * * ? *)
```

Para obter mais informações sobre o uso de expressões cron com EventBridge, consulte Expressões cron no Guia EventBridge do usuário da Amazon.

- expressões rate: o exemplo a seguir de uma expressão rate agenda uma tarefa para ser executada a cada 12 horas:

  ```
  rate(12 hour)
  ```

  Para obter mais informações sobre o uso de expressões de taxa com EventBridge, consulte Expressões de taxa no Guia EventBridge do usuário da Amazon.

Para obter mais informações sobre como EventBridge as regras se integram aos pipelines de imagem do Image Builder, consulteUse EventBridge regras com pipelines do Image Builder.

Mensagens de eventos que o Image Builder envia

O Image Builder envia mensagens de eventos para EventBridge quando há mudanças significativas no status dos recursos do Image Builder. Por exemplo, quando há uma alteração no estado de uma imagem. Os exemplos a seguir mostram mensagens de eventos JSON típicas que o Image Builder pode enviar.

Tópicos

- EC2 Alteração do estado da imagem do Image Builder
- EC2 CVE do Image Builder detectado
- EC2 Etapa de espera do fluxo de trabalho do Image Builder
- EC2 O pipeline de imagens do Image Builder é desativado automaticamente

EC2 Alteração do estado da imagem do Image Builder

O Image Builder envia esse evento quando o estado de um recurso de imagem muda durante a criação da imagem. Por exemplo, quando o status da imagem muda de um estado para outro da seguinte forma:

- De `building` até `testing`
- De `testing` até `distribution`
- De `testing` até `failed`
- De `integrating` até `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 CVE do Image Builder detectado

Se você tiver a detecção de CVE habilitada para sua imagem, o Image Builder enviará uma mensagem com os resultados sempre que uma verificação de imagem for concluída.

```
{
    "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 Etapa de espera do fluxo de trabalho do Image Builder

O Image Builder envia uma mensagem quando uma etapa `WaitForAction` do fluxo de trabalho é pausada para aguardar a conclusão de uma ação assíncrona.

```
{
    "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"
    }
}
```

EC2 O pipeline de imagens do Image Builder é desativado automaticamente

Se você configurou o `autoDisablePolicy` para seu pipeline, o Image Builder desativará o pipeline e enviará uma mensagem de evento para EventBridge quando o número de falhas consecutivas programadas na execução do pipeline exceder o número máximo permitido pela política.

```
{
    "version": "0",
    "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "detail-type": "EC2 Image Builder Image Pipeline Automatically Disabled",
    "source": "aws.imagebuilder",
    "account": "111122223333",
```

```
    "time": "2025-09-18T16:54:44Z",
    "region": "us-west-2",
    "resources": ["arn:aws:imagebuilder:us-west-2:111122223333:image-pipeline/disabled-image-pipeline-name"],
    "detail": {
        "consecutive-failures": "5"
    }
}
```

Integração do Amazon Inspector no Image Builder

Quando você ativa a verificação de segurança com o Amazon Inspector, ele verifica continuamente as imagens da máquina e as instâncias em execução na sua conta em busca de vulnerabilidades do sistema operacional e da linguagem de programação. Se ativada, a verificação de segurança será executada automaticamente e o Image Builder poderá salvar um snapshot das descobertas da sua instância de teste quando você criar uma nova imagem. O Amazon Inspector é um serviço pago.

Quando o Amazon Inspector descobre vulnerabilidades em suas configurações de software ou rede, ele executa as seguintes ações:

- Notifica você de que houve uma descoberta.
- Classifica a severidade da descoberta. A classificação de gravidade categoriza as vulnerabilidades para ajudá-lo a priorizar suas descobertas e inclui os seguintes valores:
  - Não triado
  - Informativo
  - Baixo
  - Médio
  - Alto
  - Crítico
- Fornece informações sobre a descoberta e links para recursos adicionais para obter mais detalhes.
- Oferece orientação de reparos para ajudá-lo a resolver os problemas que geraram a descoberta.

Configurar verificações de segurança

Se você ativou o Amazon Inspector para sua conta, o Amazon Inspector digitaliza automaticamente as instâncias EC2 que o Image Builder lança para criar e testar uma nova imagem. Essas instâncias

têm uma vida útil curta durante o processo de compilação e teste, e suas descobertas normalmente expiram assim que essas instâncias são encerradas. Para ajudar você a investigar e corrigir as descobertas de sua nova imagem, o Image Builder pode, opcionalmente, salvar como um snapshot todas as descobertas que o Amazon Inspector identificou em sua instância de teste durante o processo de compilação.

Para configurar verificações de segurança do pipeline, consulte Configure escaneamentos de segurança para imagens do Image Builder no Console de gerenciamento da AWS.

Analise descobertas de segurança

No console do Image Builder, você pode visualizar as descobertas de segurança de todos os seus recursos do Image Builder em um só lugar. Você pode ver todas as descobertas na página Descobertas de segurança na seção Visão geral da segurança ou agrupá-las por vulnerabilidade, por pipeline de imagens ou por imagem. Por padrão, o console exibe todas as descobertas de segurança. O painel de resumo da opção Todas as descobertas de segurança mostra o número de descobertas que você tem para cada nível de severidade. Para obter mais informações, consulte Gerencie descobertas de segurança para imagens do Image Builder no Console de gerenciamento da AWS.

Para saber mais sobre as descobertas de vulnerabilidade do Amazon Inspector, consulte Entendendo as descobertas no Amazon Inspector no Guia do Usuário do Amazon Inspector.

AWS Marketplace integração no Image Builder

AWS Marketplace é um catálogo digital com curadoria onde você pode encontrar e assinar software, dados e serviços de terceiros que ajudam a criar soluções que atendam às suas necessidades comerciais. AWS Marketplace reúne compradores autenticados e vendedores registrados com listagens de software de categorias populares, como segurança, rede, armazenamento, aprendizado de máquina e muito mais.

Um AWS Marketplace vendedor pode ser um fornecedor independente de software (ISV), um revendedor ou um indivíduo que tem algo a oferecer que funcione com AWS produtos e serviços. Quando o vendedor envia um produto AWS Marketplace, ele define o preço do produto e os termos e condições de uso. Os compradores concordam com a definição de preço e os termos e condições definidos para a oferta. Para saber mais AWS Marketplace, consulte O que é AWS Marketplace?

AWS Marketplace recursos de integração

O Image Builder se AWS Marketplace integra para fornecer os seguintes recursos diretamente do console do Image Builder:

- Pesquise produtos de imagem que estejam disponíveis em AWS Marketplace.
- Pesquise produtos de AWS Marketplace imagem que forneçam componentes.
- Veja uma lista de suas assinaturas atuais de AWS Marketplace produtos.
- Use um produto de AWS Marketplace imagem que você assinou como imagem base para uma receita do Image Builder.
- Use AWS Marketplace componentes que você assinou em uma receita do Image Builder.

O Image Builder se AWS Marketplace integra para mostrar produtos e componentes de imagem nos quais você se inscreveu. Você também pode pesquisar produtos e componentes de AWS Marketplace imagem na página Discover products sem sair do console do Image Builder.

A AMI de saída que o Image Builder cria inclui os códigos de produto dos produtos e componentes de AWS Marketplace imagem. Você pode ter até quatro códigos de produto para sua imagem final personalizada.

AWS Marketplace assinaturas no Image Builder

A página Assinaturas na AWS Marketplace seção do console do Image Builder mostra uma lista dos AWS Marketplace produtos nos quais você está inscrito no momento. Cada produto inscrito mostra os seguintes detalhes:

- O nome do produto. Isso está vinculado à página de detalhes do produto em AWS Marketplace. A página de detalhes do produto inscrito é aberta em uma nova guia no seu navegador.
- A editora. Isso está vinculado à página de detalhes do editor em AWS Marketplace. A página de detalhes do editor é aberta em uma nova guia do seu navegador.
- A versão que você assinou.
- Se houver algum componente associado incluído em seu produto inscrito, o Image Builder exibirá um link para os detalhes do componente.

Na parte superior da página, você pode pesquisar um produto específico pelo nome ou folhear seus resultados com os controles de paginação. Para usar um produto de imagem inscrito em uma nova receita, selecione um produto inscrito e escolha Criar nova receita. O Image Builder pré-seleciona o primeiro produto em sua lista por padrão.

> **Note**
>
> Se você está procurando um produto que acabou de assinar e não o vê na lista, use o botão de atualização na parte superior da guia para atualizar seus resultados. Pode demorar alguns minutos para uma nova assinatura ser exibida na lista.

Descubra produtos de AWS Marketplace imagem no console do Image Builder

Esta seção se concentra em produtos de AWS Marketplace imagem para usar como imagem base em sua receita. Para produtos que incluem componentes de software associados, você pode filtrar pelo proprietário do produto no console e na API, SDK e CLI. Para obter mais informações, consulte Listar componentes do Image Builder. Para obter mais informações sobre como encontrar, assinar e usar AWS Marketplace componentes, consulteUse AWS Marketplace componentes para personalizar sua imagem.

Descobrir produtos

Para encontrar um produto de AWS Marketplace imagem no console do Image Builder, siga estas etapas:

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. No painel de navegação, escolha Descobrir produtos na AWS Marketplaceseção.
3. Você pode pesquisar produtos de imagem na guia Produtos de imagem na página Descobrir produtos.

   O Image Builder pré-filtra os produtos AWS Marketplace para se concentrar nas imagens da máquina que você pode usar em suas receitas do Image Builder. Para obter mais informações sobre a AWS Marketplace integração com o Image Builder, escolha a guia que corresponde ao que você deseja ver.

   Essa guia contém dois painéis. À esquerda, o painel Refinar resultados ajuda você a filtrar seus resultados para encontrar os produtos que você deseja assinar. À direita, o painel Pesquisar produtos mostra os produtos que atendem aos seus critérios de filtro e também oferece a opção de pesquisar pelo nome do produto.

   Refinar resultados

A lista a seguir mostra apenas alguns dos filtros que você pode aplicar à sua pesquisa de produtos:

- Selecione uma ou mais categorias de produtos, como software de infraestrutura ou aprendizado de máquina.
- Escolha os sistemas operacionais para seu produto de imagem ou escolha todos os produtos para uma plataforma de sistema operacional específica, por exemplo, All Linux/Unix.
- Escolha um ou mais editores para exibir seus produtos disponíveis. Selecione o link Mostrar tudo para exibir todos os editores que têm produtos que se encaixam nos filtros que você aplicou.

> **Note**
>
> Os nomes dos editores não estão em ordem alfabética. Se você estiver procurando por um editor específico, por exemplo`Center for Internet Security`, você pode inserir parte do nome na caixa de pesquisa na parte superior da caixa de diálogo Todos os editores. Você deve soletrar o nome, como uma abreviatura, de forma que `CIS` talvez não produza os resultados que você está procurando.
> Você também pode navegar pelos nomes dos editores página por página.

As opções de filtro são dinâmicas. Cada escolha que você faz afeta suas opções para todas as outras categorias. Existem milhares de produtos disponíveis em AWS Marketplace, portanto, quanto mais você puder filtrar, maior a probabilidade de encontrar o que deseja.

Pesquisar produtos

Para encontrar um produto específico pelo nome, você pode inserir parte do nome na barra de pesquisa na parte superior deste painel. Cada resultado do produto inclui os seguintes detalhes:

- O nome e o logotipo do produto. Ambos estão vinculados à página de detalhes do produto em AWS Marketplace. A página de detalhes é aberta em uma nova guia do seu navegador. A partir daí, você pode assinar o produto de imagem se quiser usá-lo em uma fórmula do Image Builder. Para obter mais informações, consulte Comprar produtos no AWS Marketplace Guia do comprador.

Se você assinar o produto de imagem em AWS Marketplace, volte para a guia Image Builder em seu navegador e atualize sua lista de produtos de imagem inscritos para vê-lo.

> **Note**
>
> Pode demorar alguns minutos até sua nova assinatura estar disponível.

- O nome do editor. Isso está vinculado à página de detalhes do editor em AWS Marketplace. A página de detalhes do editor é aberta em uma nova guia do seu navegador.
- A versão do produto.
- A classificação por estrelas do produto e links diretos para a seção de avaliação da página de detalhes do produto em AWS Marketplace. A página de detalhes é aberta em uma nova guia do seu navegador.
- As primeiras linhas da descrição do produto.

Diretamente abaixo da barra de pesquisa, você pode ver quantos resultados sua pesquisa produziu e qual subconjunto desses resultados está sendo exibido atualmente. Você pode usar controles adicionais no lado direito do painel para ajustar suas configurações para o número de produtos a serem exibidos ao mesmo tempo e a ordem de classificação a ser aplicada aos resultados. Você também pode usar o controle de paginação para ver sua página de resultados.

Use um produto de AWS Marketplace imagem nas receitas do Image Builder

Abra a página Criar receita e selecione um produto de AWS Marketplace imagem para usar como imagem base, da seguinte forma.

1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
2. No painel de navegação, escolha Fórmulas de imagem na seção AWS Marketplace. Isso mostra uma lista de fórmulas de imagens que você criou.
3. Escolha Criar fórmula de imagem. Isso abre a página Criar fórmula.
4. Insira o Nome e a Versão da sua fórmula na seção Detalhes da fórmula como de costume.

5. Na seção Imagem base, escolha a opção AWS Marketplace imagens. Isso mostra uma lista dos produtos de AWS Marketplace imagem nos quais você se inscreveu na guia Assinaturas. Você pode escolher sua imagem de base na lista.

   Você também pode pesquisar outros produtos de imagem que estão disponíveis AWS Marketplace diretamente na AWS Marketplaceguia. Escolha Adicionar produtos ou abra a AWS Marketplaceguia diretamente. Para obter mais informações sobre como definir filtros e pesquisar no AWS Marketplace, consulteDescubra produtos de AWS Marketplace imagem no console do Image Builder.
6. Insira os detalhes restantes, como de costume. Se alguma de suas assinaturas de produtos incluir componentes de compilação, você poderá selecioná-los na lista Componentes de criação. Selecione na lista `AWS Marketplace` de tipos de proprietário do componente para vê-los ou selecione `Third party managed` o componente CIS.
7. Escolha Create recipe (Criar fórmula).

   Sua imagem final pode conter até quatro códigos de produto de produtos e componentes de AWS Marketplace imagem. Se a imagem base e os componentes selecionados contiverem mais de quatro códigos de produto, o Image Builder retornará um erro quando você tentar criar a receita.

Integração do Amazon SNS no Image Builder

O Amazon Simple Notification Service (Amazon SNS) é um serviço gerenciado que fornece entrega de mensagens assíncronas de publicadores para assinantes (também conhecidos como publicadores e consumidores).

Você pode especificar um tópico do SNS na configuração da sua infraestrutura. Quando você cria uma imagem ou executa um pipeline, o Image Builder pode publicar mensagens detalhadas sobre o status da imagem neste tópico. Quando o status da imagem atinge um dos seguintes estados, o Image Builder publica uma mensagem:

- `AVAILABLE`
- `FAILED`

Para obter um exemplo de mensagem de SNS do Image Builder, consulteFormato da mensagem SNS. Se você deseja criar um novo tópico do SNS, consulte Conceitos básicos do Amazon SNS no Guia do desenvolvedor do Amazon Simple Notification Service.

Tópicos do SNS criptografado

Se o tópico do SNS estiver criptografado, você deverá conceder permissão na AWS KMS key política para que a função de serviço Image Builder execute as seguintes ações:

- `kms:Decrypt`
- `kms:GenerateDataKey`

> **Note**
>
> Se o tópico do SNS estiver criptografado, a chave que criptografa esse tópico deverá residir na conta em que o serviço Image Builder é executado. O Image Builder não pode enviar notificações para tópicos do SNS criptografados com chaves de outras contas.

Exemplo de adição da política de chaves do KMS

O exemplo a seguir mostra a seção adicional que você adiciona à política de chaves do KMS. Use o Amazon Resource Name (ARN) para a função vinculada ao serviço IAM que o Image Builder criou em sua conta quando você criou uma imagem do Image Builder pela primeira vez. Para saber mais sobre a função vinculada ao serviço para o Image Builder, consulte Usar perfis do IAM vinculados ao serviço para o 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": "*"
  }]
}
```

É possível usar um dos seguintes métodos para obter um ARN.

Console de gerenciamento da AWS

Para obter o ARN da função vinculada ao serviço que o Image Builder criou em sua conta a partir do Console de gerenciamento da AWS, siga estas etapas:

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação à esquerda, escolha Funções.
3. Pesquise por`ImageBuilder`, e escolha o seguinte nome da função nos resultados:`AWSServiceRoleForImageBuilder`. Isso exibe a página de detalhes da função.
4. Para copiar o ARN na área de transferência, escolha o ícone ao nome do ARN.

AWS CLI

Para obter o ARN da função vinculada ao serviço que o Image Builder criou em sua conta a partir do AWS CLI, use o comando IAM get-role, da seguinte forma.

```
aws iam get-role --role-name AWSServiceRoleForImageBuilder
```

Saída parcial da amostra:

```
{
    "Role": {
        "Path": "/aws-service-role/imagebuilder.amazonaws.com/",
        "RoleName": "AWSServiceRoleForImageBuilder",
        ...
        "Arn": "arn:aws:iam::123456789012:role/aws-service-role/
imagebuilder.amazonaws.com/AWSServiceRoleForImageBuilder",
        ...
}
```

Formato da mensagem SNS

Depois que o Image Builder publicar uma mensagem em seu tópico do Amazon SNS, outros serviços que assinam o tópico podem filtrar o formato da mensagem e determinar se ela atende aos critérios para ações futuras. Por exemplo, uma mensagem de sucesso pode iniciar uma tarefa para atualizar um armazenamento de AWS Systems Manager parâmetros ou iniciar um fluxo de trabalho externo de teste de conformidade para a AMI de saída.

O exemplo a seguir mostra a carga JSON de uma mensagem típica que o Image Builder publica quando uma compilação de pipeline é executada até a conclusão e cria uma imagem do 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"
  }
}
```

O exemplo a seguir mostra a carga JSON de uma mensagem típica que o Image Builder publica para uma falha de compilação de pipeline de uma imagem do Linux.

```
{
  "versionlessArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-
image",
  "semver": 123794003928538027489912423 1,
```

```
  "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"
  }
}
```

Produtos de conformidade para suas imagens do Image Builder

Com os padrões de segurança em constante evolução, pode ser um desafio manter a conformidade e proteger sua organização contra ameaças cibernéticas. Para ajudar a garantir que suas imagens personalizadas estejam em conformidade e permaneçam assim por meio de atualizações automáticas quando os editores lançarem novas versões, o Image Builder se integra aos produtos de AWS Marketplace conformidade e aos componentes do Image Builder.

O Image Builder se integra aos seguintes produtos de conformidade:

- Reforço de referências do Center for Internet Security (CIS)

  Você pode usar Imagens fortalecidas CIS e os componentes de fortalecimento do CIS relacionados para criar imagens personalizadas que estejam em conformidade com as diretrizes mais recentes do CIS Benchmarks Level 1. As imagens CIS endurecidas estão disponíveis em. AWS Marketplace Para saber mais sobre como configurar e usar imagens CIS Hardened e componentes de proteção, consulte o Guia de início rápido: CIS Hardening Components for Image EC2 Builder no wiki de segurança do CIS.

  > **Note**
  >
  > Ao assinar uma imagem fortalecida CIS, você também obtém acesso ao componente de compilação associado que executa um script para aplicar as diretrizes do CIS Benchmarks

de nível 1 para sua configuração. Para obter mais informações, consulte Componentes de fortalecimento do CIS.

- Guias de implementação técnica de segurança (STIG)

  Para conformidade com o STIG, você pode usar componentes STIG gerenciados pela Amazon AWS Task Orchestrator and Executor (AWSTOE) em suas receitas do Image Builder. Os componentes do STIG examinam sua instância de compilação em busca de configurações incorretas e executam um script de correção para corrigir os problemas encontrados. Não podemos garantir a conformidade com o STIG para as imagens que você cria com o Image Builder. Você deve trabalhar com a equipe de conformidade da sua organização para verificar se sua imagem final está em conformidade. Para obter uma lista completa dos componentes AWSTOE STIG que você pode usar em suas receitas do Image Builder, consulteComponentes de hardening do STIG gerenciados pela Amazon para o Image Builder.

Monitorar eventos e logs no Image Builder

Para manter a confiabilidade, a disponibilidade e o desempenho de seus pipelines do EC2 Image Builder, é importante monitorar eventos e registros. Eventos e logs ajudam você a ter uma visão geral e se aprofundar nos detalhes quando uma chamada de API falha. O Image Builder se integra a serviços que podem enviar alertas e iniciar respostas automatizadas quando os eventos correspondem aos critérios que você configurou.

Os tópicos a seguir descrevem as técnicas de monitoramento que você pode usar por meio de serviços que se integram ao Image Builder.

Monitore eventos e logs

- Registre as chamadas da API Image Builder usando CloudTrail
- Monitore os registros do Image Builder com o Amazon CloudWatch Logs

Registre as chamadas da API Image Builder usando CloudTrail

EC2 O Image Builder é integrado com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço no Image Builder. CloudTrail captura todas as chamadas de API para o Image Builder como eventos. As chamadas capturadas incluem as chamadas do console do Image Builder e as chamadas de código para as operações da API do Image Builder. Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para o Image Builder. Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita ao Image Builder, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais CloudTrail, consulte o Guia AWS CloudTrail do usuário.

Informações do Image Builder em CloudTrail

CloudTrail é ativado no seu Conta da AWS quando você cria a conta. Quando a atividade ocorre no Image Builder, essa atividade é registrada em um CloudTrail evento junto com outros eventos AWS de serviço no histórico de eventos. Você pode visualizar, pesquisar e baixar eventos recentes no

seu Conta da AWS. Para obter mais informações, consulte Visualização de eventos com histórico de CloudTrail eventos.

Para um registro contínuo dos eventos em seu Conta da AWS, incluindo eventos do Image Builder, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as Regiões da AWS. A trilha registra eventos de todas as regiões na AWS partição e entrega os arquivos de log ao bucket do Amazon S3 que você especificar. Além disso, você pode configurar outros AWS serviços para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para obter mais informações, consulte:

- Visão geral da criação de uma trilha
- CloudTrail serviços e integrações suportados
- Configurando notificações do Amazon SNS para CloudTrail
- Recebendo arquivos de CloudTrail log de várias regiões e Recebendo arquivos de CloudTrail log de várias contas

Todas as ações do Image Builder são registradas CloudTrail e documentadas na Referência da API EC2 Image Builder. Por exemplo, chamadas para as `StartImagePipelineExecution` ações `CreateImagePipelineUpdateInfrastructureConfiguration`, e geram entradas nos arquivos de CloudTrail log.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar o seguinte:

- Se a solicitação foi feita com credenciais de usuário root ou AWS Identity and Access Management (IAM).
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para obter mais informações, consulte Elemento userIdentity do CloudTrail .

Monitore os registros do Image Builder com o Amazon CloudWatch Logs

CloudWatch O suporte a registros está ativado por padrão. O Image Builder retém os registros de compilação na instância durante o processo de criação e os transmite para CloudWatch o Logs quando a compilação é concluída. Antes de criar a imagem de compilação final, o Image Builder remove os registros locais da instância.

Registros de criação de imagens

Os registros de criação de imagens contêm os registros detalhados de uma criação de imagem individual. O Image Builder grava registros de criação de imagens no seguinte grupo e stream de CloudWatch registros do Image Builder:

LogGroup: `/aws/imagebuilder/`*`ImageName`*

LogStream (x.x.x/x): *`ImageVersion/ImageBuildVersion`*

Registros de execução do pipeline

O Image Builder grava registros de execução do pipeline quando um pipeline de imagem é executado ou quando um pipeline pula uma execução programada regularmente, junto com detalhes sobre por que isso aconteceu. Por exemplo, quando as atualizações de dependência são configuradas para um pipeline agendado, os seguintes cenários terão mensagens associadas:

- O Image Builder grava uma entrada de registro quando pula a execução do pipeline porque não havia atualizações de dependências.
- Quando há uma atualização de dependência, o Image Builder registra o ARN do recurso que foi alterado.

O Image Builder grava registros de execução do pipeline no seguinte grupo e stream de CloudWatch registros do Image Builder:

LogGroup: `/aws/imagebuilder/pipeline/`*`pipeline-name`*

LogStream: *`2025/09/01`* (a data de execução do pipeline em YYYY/MM/DD formato)

Cada registro do pipeline é anexado ao stream desse dia.

Grupos de registros personalizados

Se você especificou grupos de registros personalizados para registro de imagens ou funis, o Image Builder grava os registros no LogGroup que você especificou em seu pipeline ou em um dos comandos create image se você executou a compilação manualmente. Para obter mais informações sobre como isso funciona no Image Builder, consulteConfigurar o registro do pipeline.

Rejeitar

Você pode desativar o streaming de CloudWatch registros removendo as seguintes permissões associadas ao papel personalizado que se aplica ao seu caso de uso.

- troncos: CreateLogGroup
- troncos: CreateLogStream
- troncos: PutLogEvents

Tipo de log	Função personalizada
Construção de imagem	Perfil de execução
Execução do pipeline	Perfil de execução
Registros de componentes	Função do perfil de instância

Para solucionar problemas avançados, você pode executar comandos e scripts predefinidos usando AWS Systems Manager Executar comando. Para obter mais informações, consulte Solução de problemas do Image Builder.

Segurança no Image Builder

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de uma arquitetura de data center e rede criada para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O modelo de responsabilidade compartilhada descreve isso como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que funciona Serviços da AWS na AWS nuvem. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de AWS de . Para saber mais sobre os programas de conformidade que se aplicam ao EC2 Image Builder, consulte Serviços da AWS Escopo por Programa de Conformidade AWS .
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Image Builder. Os tópicos a seguir mostram como configurar o Image Builder para atender aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros Serviços da AWS que o ajudem a monitorar e proteger seus recursos do Image Builder.

Tópicos

- Proteção de dados e o modelo de responsabilidade AWS compartilhada no Image Builder
- Integração do Identity and Access Management com o Image Builder
- Recursos de validação da conformidade do Image Builder
- Redundância e resiliência de dados no Image Builder
- Segurança da infraestrutura no Image Builder
- Gerenciamento de patches para imagens do Image Builder
- Práticas recomendadas de segurança para o Image Builder

Proteção de dados e o modelo de responsabilidade AWS compartilhada no Image Builder

O modelo de responsabilidade AWS compartilhada se aplica à proteção de dados no EC2 Image Builder. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre o conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para saber mais sobre a privacidade de dados, consulte as Data Privacy FAQ. Para saber mais sobre a proteção de dados na Europa, consulte a postagem do blog AWS Shared Responsibility Model and RGPD no Blog de segurança da AWS .

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com Centro de Identidade do AWS IAM ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com AWS os recursos. Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como trabalhar com CloudTrail trilhas no Guia AWS CloudTrail do usuário.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sensíveis armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-3 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para saber mais sobre os endpoints FIPS disponíveis, consulte Federal Information Processing Standard (FIPS) 140-3.

É altamente recomendável que nunca sejam colocadas informações confidenciais ou sensíveis, como endereços de e-mail de clientes, em tags ou campos de formato livre, como um campo Nome.

Isso inclui quando você trabalha com o Image Builder ou outro Serviços da AWS usando o console AWS CLI, a API ou AWS SDKs. Quaisquer dados inseridos em tags ou em campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL a um servidor externo, recomendamos enfaticamente que não inclua informações de credenciais no URL para validar sua solicitação a esse servidor.

Gerenciamento de criptografia e chaves no Image Builder

O Image Builder criptografa por padrão dados em trânsito e em repouso com uma chave do KMS de propriedade do serviço, com as seguintes exceções:

- Componentes personalizados: o Image Builder criptografa componentes personalizados com sua chave padrão do KMS ou com uma chave do KMS de propriedade do serviço.
- Fluxos de trabalho de imagem: se você especificar a chave durante a criação do fluxo de trabalho, o Image Builder poderá criptografar seus fluxos de trabalho de imagem com uma chave gerenciada pelo cliente. O Image Builder processa a criptografia e a descriptografia com sua chave para executar os fluxos de trabalho que você configurou para suas imagens.

Você pode gerenciar suas próprias chaves por meio de AWS KMS. No entanto, você não tem permissão para gerenciar a chave KMS do Image Builder de propriedade do Image Builder. Para obter mais informações sobre como gerenciar suas chaves do KMS com AWS Key Management Service, consulte Introdução no Guia do AWS Key Management Service desenvolvedor.

Contexto de criptografia

Para fornecer uma verificação adicional de integridade e autenticidade em seus dados criptografados, você tem a opção de incluir um contexto de criptografia ao criptografar os dados. Quando um recurso é criptografado com um contexto de criptografia, vincula AWS KMS criptograficamente o contexto ao texto cifrado. O recurso só poderá ser descriptografado se o solicitante fornecer uma correspondência exata e com distinção entre maiúsculas e minúsculas para o contexto.

Os exemplos de políticas nesta seção usam um contexto de criptografia semelhante ao nome do recurso da Amazon (ARN) de um recurso de fluxo de trabalho do Image Builder.

Criptografar fluxos de trabalho de imagens com uma chave gerenciada pelo cliente

Para adicionar uma camada de proteção, você pode criptografar seus recursos de fluxo de trabalho do Image Builder com sua própria chave gerenciada pelo cliente. Se usar sua chave gerenciada

pelo cliente para criptografar os fluxos de trabalho do Image Builder que você cria, você deverá conceder acesso na política de chaves para que o Image Builder use sua chave ao criptografar e descriptografar recursos do fluxo de trabalho. Você pode revogar esse acesso a qualquer momento. No entanto, se você revogar o acesso à chave, o Image Builder não terá acesso a nenhum fluxo de trabalho que já esteja criptografado.

O processo para conceder ao Image Builder acesso para usar sua chave gerenciada pelo cliente tem duas etapas, conforme segue:

Etapa 1: adicionar permissões de política de chave aos fluxos de trabalho do Image Builder

Para permitir que o Image Builder criptografe e descriptografe recursos do fluxo de trabalho ao criar ou usar esses fluxos de trabalho, você deve especificar as permissões na política de chave do KMS.

Este exemplo de política de chave concede acesso aos pipelines do Image Builder para criptografar recursos de fluxo de trabalho durante o processo de criação e descriptografar recursos de fluxo de trabalho para usá-los. A política também concede acesso aos principais administradores. O contexto de criptografia e a especificação do recurso usam um curinga para cobrir todas as regiões nas quais você tenha recursos de fluxo de trabalho.

Como pré-requisito para usar fluxos de trabalho de imagem, você criou um perfil de execução de fluxo de trabalho do IAM que concede permissão para o Image Builder executar ações de fluxo de trabalho. A entidade principal da primeira declaração exibida no exemplo de política de chave aqui deve especificar seu perfil de execução de fluxo de trabalho do IAM.

Para obter mais informações sobre chaves gerenciadas pelo cliente, consulte Gerenciar o acesso a chaves gerenciadas pelo cliente no Guia do desenvolvedor do AWS Key Management Service .

JSON

```
{
 "Version":"2012-10-17",
 "Statement": [
  {
   "Sid": "Allow access to build images with encrypted workflow",
   "Effect": "Allow",
   "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/YourImageBuilderExecutionRole"
   },
   "Action": [
```

```
    "kms:Decrypt",
    "kms:GenerateDataKey"
   ],
   "Resource": "*",
   "Condition": {
    "StringLike": {
     "kms:EncryptionContext:aws:imagebuilder:arn":
 "arn:aws:imagebuilder:*:111122223333:workflow/*"
    }
   }
  },
  {
   "Sid": "Allow access for key administrators",
   "Effect": "Allow",
   "Principal": {
    "AWS": "arn:aws:iam::111122223333:root"
   },
   "Action": [
    "kms:*"
   ],
   "Resource": "arn:aws:kms:*:111122223333:key/*"
  }
 ]
}
```

Etapa 2: permitir que seu perfil de execução do fluxo de trabalho tenha acesso à chave

O perfil do IAM que o Image Builder assume para executar seus fluxos de trabalho precisa de permissão para usar sua chave gerenciada pelo cliente. Sem acesso à sua chave, o Image Builder não conseguirá criptografar ou descriptografar seus recursos de fluxo de trabalho com ela.

Edite a política do seu perfil de execução do fluxo de trabalho para adicionar a seguinte declaração de política.

JSON

```
{
 "Version":"2012-10-17",
 "Statement": [
  {
   "Sid": "AllowAccessToWorkflowKey",
```

```
    "Effect": "Allow",
    "Action": [
     "kms:Decrypt",
     "kms:GenerateDataKey"
    ],
    "Resource": "arn:aws:kms:us-west-2:111122223333:key/key_ID",
    "Condition": {
     "StringLike": {
      "kms:EncryptionContext:aws:imagebuilder:arn":
  "arn:aws:imagebuilder:*:111122223333:workflow/*"
     }
    }
   }
  ]
}
```

AWS CloudTrail eventos para fluxos de trabalho de imagem

Os exemplos a seguir mostram AWS CloudTrail entradas típicas para criptografar e descriptografar fluxos de trabalho de imagem que são armazenados com uma chave gerenciada pelo cliente.

Exemplo: GenerateDataKey.

Este exemplo mostra a aparência de um CloudTrail evento quando o Image Builder invoca a ação da AWS KMS GenerateDataKey API a partir da ação da CreateWorkflow API Image Builder. Antes de criar o recurso de fluxo de trabalho, o Image Builder deve criptografar um novo fluxo de trabalho.

```
{
 "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"
}
```


Exemplo: Decrypt

Este exemplo mostra a aparência de um CloudTrail evento quando o Image Builder invoca a ação da AWS KMS Decrypt API a partir da ação da GetWorkflow API Image Builder. Antes que possam usar um recurso de fluxo de trabalho, os pipelines do Image Builder precisam descriptografá-lo.


```
{
 "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"
}
```

Armazenamento de dados no Image Builder

O Image Builder não armazena nenhum dos seus registros no serviço. Todos os registros são salvos na sua EC2 instância da Amazon que é usada para criar a imagem ou nos seus registros de automação do Systems Manager.

Privacidade do tráfego entre redes no Image Builder

As conexões são protegidas entre o Image Builder e locais locais, entre uma AZs AWS região e entre AWS regiões por meio de HTTPS. Não há conexões diretas entre contas.

Integração do Identity and Access Management com o Image Builder

Tópicos

- Público
- Autenticação com identidades
- Como o Image Builder utiliza políticas e perfis do IAM
- Gerencie perímetros de dados para acesso ao download do bucket S3 no Image Builder

- Políticas baseadas em identidade do Image Builder
- Permissões do IAM para fluxos de trabalho personalizados
- Políticas baseadas em recursos do Image Builder
- Use políticas AWS gerenciadas para EC2 Image Builder
- Usar perfis do IAM vinculados ao serviço para o Image Builder
- Solucionar problemas do IAM no Image Builder

Público

A forma como você usa AWS Identity and Access Management (IAM) difere com base na sua função:

- Usuário do serviço: solicite permissões ao seu administrador se você não conseguir acessar os atributos (consulte Solucionar problemas do IAM no Image Builder).
- Administrador do serviço: determine o acesso do usuário e envie solicitações de permissão (consulte Como o Image Builder utiliza políticas e perfis do IAM)
- Administrador do IAM: escreva políticas para gerenciar o acesso (consulte Políticas baseadas em identidade do Image Builder)

Autenticação com identidades

Para obter informações detalhadas sobre como fornecer autenticação para pessoas e processos em seu Conta da AWS, consulte Identidades no Guia do usuário do IAM.

Como o Image Builder utiliza políticas e perfis do IAM

Antes de usar o IAM para gerenciar o acesso ao Image Builder, saiba quais atributos do IAM estão disponíveis para uso com o Image Builder.

Para ter uma visão de alto nível de como o Image Builder e outros AWS serviços funcionam com a maioria dos recursos do IAM, consulte AWS os serviços que funcionam com o IAM no Guia do usuário do IAM.

Políticas baseadas em identidade para o Image Builder

Compatível com políticas baseadas em identidade: sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que podem ser anexados a uma identidade, como usuário do IAM, grupo de usuários ou perfil. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte Referência de elemento de política JSON do IAM no Guia do usuário do IAM.

Exemplos de políticas baseadas em identidade para o Image Builder

Para visualizar exemplos de políticas baseadas em identidade do Image Builder, consulte Políticas baseadas em identidade do Image Builder.

Políticas baseadas em recursos no Image Builder

Compatível com políticas baseadas em recursos: sim

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. É necessário especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para permitir o acesso entre contas, é possível especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em recursos. Consulte mais informações em Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Ações de política para Image Builder

Compatível com ações de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Action` de uma política JSON descreve as ações que podem ser usadas para permitir ou negar acesso em uma política. Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista de ações do Image Builder, consulte Ações definidas pelo EC2 Image Builder na Referência de Autorização de Serviço.

As ações de políticas no Image Builder usam o seguinte prefixo antes da ação:

```
imagebuilder
```

Para especificar várias ações em uma única declaração, separe-as com vírgulas.

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

Para visualizar exemplos de políticas baseadas em identidade do Image Builder, consulte Políticas baseadas em identidade do Image Builder.

Recursos de política para Image Builder

Compatível com recursos de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Para ações que não oferecem compatibilidade com permissões em nível de recurso, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

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

Para ver uma lista dos tipos de recursos do Image Builder e seus ARNs, consulte Recursos definidos pelo EC2 Image Builder na Referência de Autorização de Serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte Ações definidas pelo EC2 Image Builder.

Para visualizar exemplos de políticas baseadas em identidade do Image Builder, consulte Políticas baseadas em identidade do Image Builder.

Chaves de condição de políticas do Image Builder

Compatível com chaves de condição de política específicas de serviço: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Condition` especifica quando as instruções são executadas com base em critérios definidos. É possível criar expressões condicionais que usem agentes de condição, como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação. Para ver todas as chaves de condição AWS globais, consulte as chaves de contexto de condição AWS global no Guia do usuário do IAM.

Para ver uma lista das chaves de condição do Image Builder, consulte Chaves de condição do EC2 Image Builder na Referência de Autorização de Serviço. Para saber com quais ações e recursos você pode usar uma chave de condição, consulte Ações definidas pelo EC2 Image Builder.

Para visualizar exemplos de políticas baseadas em identidade do Image Builder, consulte Políticas baseadas em identidade do Image Builder.

ACLs em Image Builder

Suportes ACLs: Não

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

ABAC com Image Builder

Compatível com ABAC (tags em políticas): parcial

O controle de acesso por atributo (ABAC) é uma estratégia de autorização que define permissões com base em atributos chamados de tags. Você pode anexar tags a entidades e AWS recursos do IAM e, em seguida, criar políticas ABAC para permitir operações quando a tag do diretor corresponder à tag no recurso.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou chaves de condição `aws:TagKeys`.

Se um serviço for compatível com as três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço for compatível com as três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para saber mais sobre o ABAC, consulte Definir permissões com autorização do ABAC no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte Usar controle de acesso por atributo (ABAC) no Guia do usuário do IAM.

Usar credenciais temporárias com o Image Builder

Compatível com credenciais temporárias: sim

As credenciais temporárias fornecem acesso de curto prazo aos AWS recursos e são criadas automaticamente quando você usa a federação ou troca de funções. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para ter mais informações, consulte Credenciais de segurança temporárias no IAM e Serviços da Serviços da AWS que funcionam com o IAM no Guia do usuário do IAM.

Permissões de entidade principal entre serviços para o Image Builder

Compatibilidade com o recurso de encaminhamento de sessões de acesso (FAS): sim

As sessões de acesso direto (FAS) usam as permissões do principal chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) de fazer solicitações aos serviços posteriores. Para obter detalhes da política ao fazer solicitações de FAS, consulte Sessões de acesso direto.

Perfis de serviço para o Image Builder

Compatível com perfis de serviço: sim

O perfil de serviço é um perfil do IAM que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para saber mais, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do Usuário do IAM.

> **Warning**
>
> Alterar as permissões de um perfil de serviço pode prejudicar a funcionalidade do Image Builder. Edite os perfis de serviço somente quando o Image Builder orientar você a fazê-lo.

Funções vinculadas ao serviço para o Image Builder

Compatibilidade com perfis vinculados a serviços: sim

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode assumir o perfil de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para perfis vinculados ao serviço.

Para detalhes sobre a função vinculada ao serviço para o Image Builder, consulte Usar perfis do IAM vinculados ao serviço para o Image Builder.

Políticas baseadas em identidade do Image Builder

Com as políticas baseadas em identidade do IAM, é possível especificar ações ou recursos permitidos ou negados, além das condições sob as quais as ações são permitidas ou negadas. O Image Builder oferece suporte a ações, recursos e chaves de condição específicos. Para obter informações sobre todos os elementos que você usa em uma política JSON, consulte Ações, recursos e chaves de condição para o Amazon EC2 Image Builder no Guia do usuário do IAM.

Ações

As ações de políticas no Image Builder usam o seguinte prefixo antes da ação: `imagebuilder:`. As instruções de política devem incluir um elemento `Action` ou `NotAction`. O Image Builder define seu próprio conjunto de ações que descrevem as tarefas que você pode executar com esse serviço.

Para especificar várias ações em uma única instrução, separe-as com vírgulas, como segue:

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

Você também pode especificar várias ações usando caracteres curinga (*). Por exemplo, para especificar todas as ações que começam com a palavra `List`, inclua a seguinte ação:

```
"Action": "imagebuilder:List*"
```

Para ver uma lista de ações do Image Builder , consulte Serviços da AWSAções, recursos e chaves de condição para o no Manual do usuário do IAM.

Gerenciar o acesso usando políticas

Para obter informações detalhadas sobre como gerenciar o acesso criando políticas e anexando-as às identidades ou AWS recursos do IAM, consulte Políticas e permissões no Guia do usuário do IAM. AWS

O perfil do IAM que você associa ao seu perfil de instância precisa ter permissões para executar os componentes de criação e teste incluídos na sua imagem. As seguintes políticas de perfil do IAM devem ser anexadas à função do IAM associada ao perfil de instância:

- EC2InstanceProfileForImageBuilder
- EC2InstanceProfileForImageBuilderECRContainerBuilds
- AmazonSSMManagedInstanceCore

Recursos

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Para ações que não oferecem compatibilidade com permissões em nível de recurso, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

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

O ARN é composto por vários nós que ajudam a identificar o recurso e garantir que o nome seja exclusivo. Os últimos nós no nome incluem diversas variações na formatação do tipo, nome e ID do recurso. Quando o Image Builder cria um recurso, ele usa o seguinte formato:

`arn:aws:imagebuilder:region:owner:resource-type/resource-name/version/build-version`

> **Note**
>
> A versão de compilação nem sempre está incluída no ARN do recurso. No entanto, algumas operações de API, como GetComponent, precisam da versão de compilação para identificar de forma exclusiva um recurso a ser recuperado.

Para os recursos que o Image Builder usa em suas receitas, como a imagem base ou os componentes, o nó proprietário pode ser um dos seguintes:

- O número da conta do proprietário do recurso
- Para recursos gerenciados pela Amazon: `aws`
- Para obter AWS Marketplace recursos: `aws-marketplace`

O exemplo a seguir mostra o ARN de um componente gerenciado para instalar o CloudWatch agente da Amazon no Linux:

```
arn:aws:imagebuilder:us-east-1:aws:component/amazon-cloudwatch-agent-linux/1.0.1/1
```

Este exemplo mostra o ARN de um componente gerenciado fictício do: AWS Marketplace

```
arn:aws:imagebuilder:us-east-1:aws-marketplace:component/example-linux-software-component/1.0.1
```

Para obter mais informações sobre como obter uma lista de componentes, incluindo o uso de um filtro de propriedade, consulteListar componentes do Image Builder.

Exemplo ARNs

Veja a seguir alguns exemplos de recursos ARNs que você pode especificar em uma política do IAM:

- Instância ARN

```
"Resource": "arn:aws:imagebuilder:us-east-1:111122223333:instance/i-1234567890abcdef0"
```

- Exemplo de curinga (*) para especificar todas as instâncias de uma determinada conta

```
"Resource": "arn:aws:imagebuilder:us-east-1:111122223333:instance/*"
```

- Exemplo de curinga (*) para especificar todas as versões de um fluxo de trabalho de imagem gerenciada

```
"Resource": "arn:aws:imagebuilder:us-east-1:aws:workflow/build/build-image/*"
```

- ARN de imagem gerenciada

```
"Resource": "arn:aws:imagebuilder:us-east-1:aws:image/amazon-linux-2-
arm64/2024.12.17/1"
```

- Exemplo de curinga (*) para especificar todas as versões de uma imagem gerenciada

```
"Resource": "arn:aws:imagebuilder:us-east-1:aws:image/amazon-linux-2-arm64/x.x.x"
```

Muitas ações da API EC2 Image Builder envolvem vários recursos. Para especificar vários recursos em uma única instrução, separe-os ARNs com vírgulas.

```
"Resource": [
   "resource1",
   "resource2"
]
```

Chaves de condição

O Image Builder fornece chaves de condições específicas ao serviço e é compatível com o uso de algumas chaves de condição globais. Para ver todas as chaves de condição AWS globais, consulte Chaves de contexto de condição AWS global no Guia do usuário do IAM. A seguintes chaves de condição específicas ao serviço são fornecidas.

construtor de imagens: CreatedResourceTagKeys

Funciona com operadores de string.

Use esta chave para filtrar acesso por presença de chaves de tag na solicitação. Isso permite gerenciar os recursos que o Image Builder cria.

Disponibilidade — Essa chave está disponível somente para `CreateInfrastrucutreConfiguration` `UpdateInfrastructureConfiguration` APIs e.

construtor de imagens:/CreatedResourceTag<key>

Funciona com operadores de string.

Use esta chave para filtrar o acesso por pares de chave-valor da etiqueta anexados ao recurso criado pelo Image Builder. Isso permite gerenciar os recursos do Image Builder por meio de tags definidas.

Disponibilidade — Essa chave está disponível somente para `CreateInfrastrucutreConfiguration` `UpdateInfrastructureConfiguration` APIs e.

construtor de imagens: LifecyclePolicyResourceType

Funciona com operadores de string.

Use essa chave para filtrar o acesso pelo tipo de recurso do ciclo de vida especificado na solicitação.

O valor dessa chave pode ser `AMI_IMAGE` ou`CONTAINER_IMAGE`.

Disponibilidade — Essa chave está disponível somente para `CreateLifecyclePolicy` `UpdateLifecyclePolicy` APIs e.

Construtor de imagens: EC2 MetadataHttpTokens

Funciona com operadores de string.

Use essa chave para filtrar o acesso pelo requisito de token HTTP de metadados da EC2 instância especificado na solicitação.

Esse valor para essa chave pode ser `optional` o`urequired`.

Disponibilidade — Essa chave está disponível somente para `CreateInfrastrucutreConfiguration` `UpdateInfrastructureConfiguration` APIs e.

construtor de imagens: StatusTopicArn

Funciona com operadores de string.

Use esta chave para filtrar o acesso pelo ARN do tópico do SNS na solicitação na qual as notificações de estado do terminal serão publicadas.

Disponibilidade — Essa chave está disponível somente para `CreateInfrastrucutreConfiguration` `UpdateInfrastructureConfiguration` APIs e.

Exemplos

Para visualizar exemplos de políticas baseadas em identidade do Image Builder, consulte Políticas baseadas em identidade do Image Builder.

Políticas baseadas em recursos do Image Builder

As políticas baseadas em recursos especificam quais ações uma entidade principal específica pode executar no recurso do Image Builder e sob quais condições. O Image Builder oferece suporte a políticas de permissões baseadas em recursos para componentes, imagens e fórmulas de imagens. As políticas baseadas em recursos permitem conceder permissão de uso a outras contas especificada por recurso. Você também pode usar uma política baseada em recursos para permitir que um AWS serviço acesse seus componentes, imagens e receitas de imagens.

Para obter informações sobre como anexar uma política baseada em recursos a um componente, uma imagem ou uma fórmula de imagem, consulte Compartilhe recursos do Image Builder com AWS RAM.

> **Note**
>
> Quando você atualiza uma política de recursos usando o Image Builder, a atualização aparecerá no console da RAM.

Autorização baseada em tags do construtor de imagens do construtor

Você pode anexar tags a recursos do Image Builder ou passar tags em uma solicitação ao Image Builder. Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as `imagebuilder:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* ou chaves de condição `aws:TagKeys`. Para obter mais informações sobre marcação de recursos no Image Builder, consulte Marque um recurso do AWS CLI.

Funções do IAM do construtor de imagens

Uma função do IAM é uma entidade dentro da sua Conta da AWS que tem permissões específicas.

Usar credenciais temporárias com o Image Builder

É possível usar credenciais temporárias para fazer login com federação, assumir um perfil do IAM ou assumir um perfil entre contas. Você obtém credenciais de segurança temporárias chamando operações de AWS STS API, como AssumeRoleou GetFederationToken.

Perfis vinculados ao serviço

As funções vinculadas ao serviço permitem Serviços da AWS acessar recursos em outros serviços para concluir uma ação em seu nome. Os perfis vinculados a serviço aparecem em sua conta do IAM e são de propriedade do serviço. Um usuário com acesso administrativo pode visualizar, mas não pode editar as permissões para funções vinculadas ao serviço.

O Image Builder é compatível com funções vinculadas ao serviço. Para obter informações sobre como criar ou gerenciar funções vinculadas ao serviço do Image Builder, consulte Usar perfis do IAM vinculados ao serviço para o Image Builder.

Perfis de serviço

Esse atributo permite que um serviço assuma um perfil de serviço em seu nome. O perfil permite que o serviço acesse recursos em outros serviços para concluir uma ação em seu nome. Os perfis de serviço aparecem em sua conta do IAM e são de propriedade da conta. Isso significa que um usuário com acesso administrativo pode alterar as permissões para essa função. Porém, fazer isso pode alterar a funcionalidade do serviço.

Gerencie perímetros de dados para acesso ao download do bucket S3 no Image Builder

EC2 O Image Builder mantém duas classes de buckets S3 AWS de propriedade do serviço que contêm os recursos disponíveis para download necessários para executar as cargas de trabalho do Image Builder em sua conta. Se você usa perímetros de dados para controlar o acesso ao Amazon S3 em seu ambiente, talvez seja necessário permitir explicitamente o acesso a esses buckets. Você pode usar o ARN do bucket ou o URL do bucket para permitir esses buckets, dependendo de como você controla o acesso ao Amazon S3.

Scripts de inicialização do gerenciamento de componentes (obrigatório)

> Esse bucket do S3 contém scripts de inicialização para configurar o AWSTOE aplicativo nas EC2 instâncias usadas para criar imagens. O Image Builder requer acesso para baixar os scripts para apoiar a criação e o teste de novas imagens.

- ARN do bucket S3: `arn:<AWS partition>:s3:::ec2imagebuilder-managed-resources-<AWS Region>-prod`
- URL do bucket S3: `https://ec2imagebuilder-managed-resources-<AWS Region>.s3.<AWS Region>.<AWS partition-specific domain name>`

Componentes gerenciados

Esse bucket do S3 contém cargas úteis de pacotes para componentes gerenciados pela Amazon. O Image Builder requer acesso para baixar quaisquer componentes gerenciados que estejam configurados em suas receitas.

- ARN do bucket S3: `arn:<AWS partition>:s3:::ec2imagebuilder-toe-<AWS Region>-prod`
- URL do bucket S3: `https://ec2imagebuilder-toe-<AWS Region>.s3.<AWS Region>.<AWS partition-specific domain name>`

Políticas baseadas em identidade do Image Builder

Tópicos

- Melhores práticas de política baseada em identidade
- Usando o console Image Builder

Melhores práticas de política baseada em identidade

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos do Image Builder em sua conta. Essas ações podem gerar custos para sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para saber mais, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para funções de trabalho no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações

que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para saber mais sobre como usar o IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do usuário do IAM.

- Use condições nas políticas do IAM para restringir ainda mais o acesso: é possível adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, é possível escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como CloudFormation. Para saber mais, consulte Elementos da política JSON do IAM: condição no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para saber mais, consulte Validação de políticas do IAM Access Analyzer no Guia do Usuário do IAM.
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para saber mais, consulte Configuração de acesso à API protegido por MFA no Guia do Usuário do IAM.

Para saber mais sobre as práticas recomendadas do IAM, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

Usando o console Image Builder

Para acessar o console do EC2 Image Builder, você deve ter um conjunto mínimo de permissões. Essas permissões devem autorizar você a listar e visualizar detalhes sobre os recursos do Image Builder na sua Conta da AWS. Se você criar uma política baseada em identidade que seja mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis do IAM) com essa política.

Para garantir que suas entidades do IAM possam usar o console do Image Builder, você deve anexar uma das seguintes políticas AWS gerenciadas a elas:

- Política AWSImageBuilderReadOnlyAccess
- Política AWSImageBuilderFullAccess

Para obter mais informações sobre políticas gerenciadas pelo Image Builder, consulte Use políticas AWS gerenciadas para EC2 Image Builder.

> **Important**
>
> A AWSImageBuilderFullAccesspolítica é necessária para criar a função vinculada ao serviço do Image Builder. Ao anexar essa política a uma entidade do IAM, você também deve anexar a seguinte política personalizada e incluir os recursos que deseja usar e que não têm `imagebuilder` no nome do recurso:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sns:Publish"
            ],
            "Resource": "arn:aws:sns:*:*:*imagebuilder*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetInstanceProfile"
            ],
            "Resource": "arn:aws:iam::*:instance-profile/*imagebuilder*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::*:instance-profile/*imagebuilder*",
                "arn:aws:iam::*:role/*imagebuilder*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ec2.amazonaws.com"
                }
```

```
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3::*:*imagebuilder*"
        }
    ]
}
```

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a AWS API. Em vez disso, permita o acesso somente às ações que corresponderem a operação da API que você estiver tentando executar.

Permissões do IAM para fluxos de trabalho personalizados

Ao usar fluxos de trabalho personalizados com ações de etapas específicasRegisterImage, como permissões adicionais do IAM, podem ser necessárias além das políticas gerenciadas padrão do Image Builder. Esta seção descreve as permissões adicionais necessárias para ações personalizadas das etapas do fluxo de trabalho.

RegisterImage permissões de ação de etapa

A ação da `RegisterImage` etapa requer EC2 permissões específicas da Amazon para registrar AMIs e, opcionalmente, recuperar tags de snapshot. Ao usar o `includeSnapshotTags` parâmetro, são necessárias permissões adicionais para descrever os instantâneos.

Permissões necessárias para a ação da RegisterImage etapa:

Para todos os recursos, permita as seguintes ações:

- `ec2:RegisterImage`
- `ec2:DescribeSnapshots`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Action": [
                "ec2:RegisterImage",
                "ec2:DescribeSnapshots"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*::image/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "RegisterImage"
                }
            }
        }
    ]
}
```

Detalhes da permissão:

- `ec2:RegisterImage`- Necessário para registrar novos a AMIs partir de instantâneos
- `ec2:DescribeSnapshots`- Obrigatório ao usar `includeSnapshotTags: true` para recuperar tags de instantâneo para mesclagem com tags AMI
- `ec2:CreateTags`- Obrigatório para aplicar tags à AMI registrada, incluindo as tags padrão do Image Builder e as tags de snapshot mescladas

> **Note**
>
> A `ec2:DescribeSnapshots` permissão só é usada quando o `includeSnapshotTags` parâmetro é definido como`true`. Se você não usar esse recurso, poderá omitir essa permissão.

Comportamento de mesclagem de tags:

Quando `includeSnapshotTags` estiver ativada, a ação da RegisterImage etapa:

- Recupere as tags do primeiro instantâneo especificado no mapeamento do dispositivo de bloco
- Exclua todas as tags AWS reservadas (aquelas com chaves começando com “aws:”)
- Mescle tags de snapshot com as tags de registro de AMI padrão do Image Builder
- Dê precedência às tags do Image Builder quando as chaves de tag entrarem em conflito

Políticas baseadas em recursos do Image Builder

Para informações sobre como criar um componente, consulte Usar componentes para personalizar sua imagem do Image Builder.

Restringir o acesso do componente do construtor de imagens a endereços IP específicos

O exemplo a seguir concede permissões a qualquer usuário para executar qualquer operação do Image Builder em componentes. No entanto, a solicitação deve se originar no intervalo de endereços IP especificados na condição.

A condição nesta declaração identifica o intervalo 54.240.143.* de endereços IP permitidos do Protocolo de Internet versão 4 (IPv4), com uma exceção: 54.240.143.188.

O `Condition` bloco usa as `NotIpAddress` condições `IpAddress` e e a chave de `aws:SourceIp` condição, que é uma chave AWS de condição ampla. Para obter mais informações sobre essas chaves de condições, consulte Especificar condições em uma política. Os `aws:sourceIp` IPv4 valores usam a notação CIDR padrão. Para obter mais informações, consulte Operadores de condição de endereço IP no Guia do usuário do IAM.

JSON

```
{
  "Version":"2012-10-17",
  "Id": "IBPolicyId1",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Allow",
      "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"}
        }
      }
    ]
}
```

Use políticas AWS gerenciadas para EC2 Image Builder

Uma política AWS gerenciada é uma política autônoma criada e administrada por AWS. AWS as políticas gerenciadas são projetadas para fornecer permissões para muitos casos de uso comuns, para que você possa começar a atribuir permissões a usuários, grupos e funções.

Lembre-se de que as políticas AWS gerenciadas podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque estão disponíveis para uso de todos os AWS clientes. Recomendamos que você reduza ainda mais as permissões definindo as políticas gerenciadas pelo cliente que são específicas para seus casos de uso.

Você não pode alterar as permissões definidas nas políticas AWS gerenciadas. Se AWS atualizar as permissões definidas em uma política AWS gerenciada, a atualização afetará todas as identidades principais (usuários, grupos e funções) às quais a política está anexada. AWS é mais provável que atualize uma política AWS gerenciada quando uma nova AWS service (Serviço da AWS) for lançada ou novas operações de API forem disponibilizadas para serviços existentes.

Para saber mais, consulte AWS Políticas gerenciadas pela no Guia do usuário do IAM.

Política AWSImageBuilderFullAccess

A política AWSImageBuilderFullAccess concede acesso total aos recursos do Image Builder para a função à qual está associada, permitindo que a função liste, descreva, crie, atualize e exclua recursos do Image Builder. A política também concede permissões direcionadas aos relacionados Serviços da AWS que são necessários, por exemplo, para verificar recursos ou exibir os recursos atuais da conta no Console de gerenciamento da AWS.

Detalhes das permissões

Esta política inclui as seguintes permissões:

- Image Builder: o acesso administrativo é concedido para que a função possa listar, descrever, criar, atualizar e excluir recursos do Image Builder.

- Amazon EC2 — O acesso é concedido para ações do Amazon EC2 Describe que são necessárias para verificar a existência de recursos ou obter listas de recursos pertencentes à conta.
- IAM: o acesso é concedido para obter e usar perfis de instância cujo nome contém “imagebuilder”, para verificar a existência da função vinculada ao serviço Image Builder por meio da ação de API `iam:GetRole` e para criar a função vinculada ao serviço Image Builder.
- License Manager: o acesso é concedido para listar as configurações de licenças ou licenças de um recurso.
- Amazon S3: o acesso é concedido aos buckets da lista pertencentes à conta e também aos buckets do Image Builder com “imagebuilder” em seus nomes.
- Amazon SNS: permissões de gravação são concedidas ao Amazon SNS para verificar a propriedade dos tópicos que contêm “imagebuilder”.

Para visualizar as permissões para esta política, consulte AWSImageBuilderFullAccess na Referência de políticas gerenciadas pela AWS .

Política AWSImageBuilderReadOnlyAccess

A política do AWSImageBuilderReadOnlyAccess concede acesso de somente leitura a todos os recursos do Image Builder. As permissões são concedidas para verificar se a função vinculada ao serviço Image Builder existe por meio da ação da API `iam:GetRole`.

Detalhes das permissões

Esta política inclui as seguintes permissões:

- Image Builder: o acesso é concedido de somente leitura aos recursos do Image Builder.
- IAM: o acesso é concedido para verificar a existência da função vinculada ao serviço Image Builder por meio da ação da API `iam:GetRole`.

Para visualizar as permissões para esta política, consulte AWSImageBuilderReadOnlyAccess na Referência de políticas gerenciadas pela AWS .

Política AWSServiceRoleForImageBuilder

A AWSServiceRoleForImageBuilderpolítica permite que o Image Builder ligue Serviços da AWS em seu nome.

Detalhes das permissões

Esta política é anexada à função vinculada ao serviço Image Builder quando a função é criada por meio do Systems Manager. Para obter mais informações sobre a função vinculada ao serviço do Image Builder, consulte Usar perfis do IAM vinculados ao serviço para o Image Builder.

A política inclui as seguintes permissões:

- CloudWatch Registros — O acesso é concedido para criar e carregar CloudWatch registros em qualquer grupo de registros cujo nome comece com`/aws/imagebuilder/`.
- Amazon EC2 — O acesso é concedido ao Image Builder para criar, tirar fotos e registrar imagens (AMIs) que ele cria e executar EC2 instâncias em sua conta. O Image Builder usa instantâneos, volumes, interfaces de rede, sub-redes, grupos de segurança, configuração de licenças e pares de chaves relacionados conforme necessário, desde que a imagem, a instância e os volumes que estão sendo criados ou usados estejam marcados com ou. `CreatedBy: EC2 Image Builder` `CreatedBy: EC2 Fast Launch`

  O Image Builder pode obter informações sobre EC2 imagens da Amazon, atributos da instância, status da instância, tipos de instância que estão disponíveis para sua conta, modelos de lançamento, sub-redes, hosts e tags em seus recursos da Amazon EC2 .

  O Image Builder pode atualizar as configurações de imagem para ativar ou desativar o lançamento mais rápido de instâncias do Windows em sua conta, onde a imagem está marcada com `CreatedBy: EC2 Image Builder`.

  Além disso, o Image Builder pode iniciar, interromper e encerrar instâncias em execução na sua conta, compartilhar snapshots do Amazon EBS, criar e atualizar imagens e executar modelos, cancelar o registro de imagens existentes, adicionar tags e replicar imagens em contas às quais você concedeu permissões por meio da política Ec2ImageBuilderCrossAccountDistributionAccess. A marcação do Image Builder é necessária para todas essas ações, conforme descrito anteriormente.
- Amazon ECR: o acesso é concedido ao Image Builder para criar um repositório, se necessário, para escanear vulnerabilidades de imagens de contêiner e marcar os recursos que ele cria para limitar o escopo de suas operações. O acesso também é concedido ao Image Builder para excluir as imagens de contêiner que ele criou para as digitalizações depois de tirar snapshots das vulnerabilidades.
- EventBridge— O acesso é concedido ao Image Builder para criar e gerenciar EventBridge regras.

- IAM — O acesso é concedido para que o Image Builder passe qualquer função em sua conta para a Amazon EC2 e para a VM Import/Export.
- Amazon Inspector: o acesso é concedido ao Image Builder para determinar quando o Amazon Inspector conclui as varreduras de instâncias de compilação e para coletar descobertas de imagens que estão configuradas para permitir isso.
- AWS KMS: o acesso é concedido ao Amazon EBS para criptografar, descriptografar ou recriptografar volumes do Amazon EBS. Isso é crucial para garantir que os volumes criptografados funcionem quando o Image Builder cria uma imagem.
- License Manager: o acesso é concedido ao Image Builder para atualizar as especificações do License Manager via `license-manager:UpdateLicenseSpecificationsForResource`.
- Amazon SNS: as permissões de gravação são concedidas para qualquer tópico do Amazon SNS na sua conta.
- Systems Manager: o acesso é concedido ao Image Builder para listar os comandos do Systems Manager e as respectivas invocações e entradas de inventário, descrever informações de instância e status de execução de automação, descrever hosts para suporte ao posicionamento de instâncias e obter detalhes de invocação de comandos. O Image Builder também pode enviar sinais de automação e interromper as execuções de automação para qualquer recurso em sua conta.

  O Image Builder é capaz de emitir invocações de comando de execução para qualquer instância que esteja com a tag `"CreatedBy": "EC2 Image Builder"` para os seguintes arquivos de script: `AWS-RunPowerShellScript`, `AWS-RunShellScript` ou `AWSEC2-RunSysprep`. O Image Builder é capaz de iniciar uma execução de automação do Systems Manager em sua conta para documentos de automação em que o nome começa com `ImageBuilder`.

  O Image Builder também pode criar ou excluir associações do State Manager para qualquer instância em sua conta, desde que o documento de associação seja `AWS-GatherSoftwareInventory`. Além disso, pode criar a função vinculada ao serviço Systems Manager em sua conta.

  O Image Builder é capaz de ler parâmetros públicos do Parameter Store e ler e atualizar parâmetros privados prefixados com `/imagebuilder/` para que ele possa atualizar o valor do parâmetro com a AMI de saída IDs que o Image Builder cria a partir de uma nova compilação.
- AWS STS: o acesso é concedido para que o Image Builder assuma funções nomeadas EC2ImageBuilderDistributionCrossAccountRole de sua conta em qualquer conta em que a política de confiança da função permita. Isso é usado para distribuição de imagens entre contas.

Para visualizar as permissões para esta política, consulte AWSServiceRoleForImageBuilder na Referência de políticas gerenciadas pela AWS .

Política Ec2ImageBuilderCrossAccountDistributionAccess

A política do Ec2ImageBuilderCrossAccountDistributionAccess concede permissões para o Image Builder distribuir imagens entre contas nas regiões de destino. Além disso, o Image Builder pode descrever, copiar e aplicar tags a qualquer EC2 imagem da Amazon na conta. A política também concede a capacidade de modificar as permissões da AMI por meio da ação da API `ec2:ModifyImageAttribute`.

Detalhes das permissões

Esta política inclui as seguintes permissões:

- Amazon EC2 — O acesso é concedido EC2 à Amazon para descrever, copiar e modificar atributos de uma imagem e criar tags para qualquer EC2 imagem da Amazon na conta.

Para visualizar as permissões para esta política, consulte Ec2ImageBuilderCrossAccountDistributionAccess na Referência de políticas gerenciadas pela AWS .

Política EC2ImageBuilderLifecycleExecutionPolicy

A EC2ImageBuilderLifecycleExecutionPolicypolítica concede permissões para que o Image Builder execute ações como descontinuar, desativar ou excluir recursos de imagem do Image Builder e seus recursos subjacentes (AMIsinstantâneos) para dar suporte a regras automatizadas para tarefas de gerenciamento do ciclo de vida de imagens.

Detalhes das permissões

Esta política inclui as seguintes permissões:

- Amazon EC2 — O acesso é concedido EC2 à Amazon para realizar as seguintes ações para Amazon Machine Images (AMIs) na conta que está marcada com`CreatedBy: EC2 Image Builder`.
  - Habilite e desabilite uma AMI.
  - Habilite e desabilite a descontinuação de imagens.
  - Descreva e cancele o registro de uma AMI.

  - Descreva e modifique os atributos da imagem de AMI.
  - Exclua os snapshots de volume associados à AMI.
  - Recupere as tags para um recurso.
  - Adicionar ou remover tags de uma AMI para descontinuação.
- Amazon ECR: o acesso é concedido ao Amazon ECR para realizar as seguintes ações em lote nos repositórios dp ECR com a tag `LifecycleExecutionAccess: EC2 Image Builder`. As ações em lote são compatíveis com regras automatizadas de ciclo de vida de imagens de contêineres.
  - `ecr:BatchGetImage`
  - `ecr:BatchDeleteImage`

  O acesso é concedido no nível do repositório para repositórios do ECR marcados com `LifecycleExecutionAccess: EC2 Image Builder`.
- AWS Grupos de recursos — O acesso é concedido ao Image Builder para obter recursos com base em tags.
- EC2 Image Builder — O acesso é concedido ao Image Builder para excluir recursos de imagem do Image Builder.

Para visualizar as permissões para esta política, consulte EC2ImageBuilderLifecycleExecutionPolicy na Referência de políticas gerenciadas pela AWS .

Política EC2InstanceProfileForImageBuilder

A EC2InstanceProfileForImageBuilderpolítica concede as permissões mínimas necessárias para que uma EC2 instância funcione com o Image Builder. Isso não inclui as permissões necessárias para usar o Systems Manager Agent.

Detalhes das permissões

Esta política inclui as seguintes permissões:

- CloudWatch Registros — O acesso é concedido para criar e carregar CloudWatch registros em qualquer grupo de registros cujo nome comece com`/aws/imagebuilder/`.
- Amazon EC2 — O acesso é concedido para descrever volumes e instantâneos, criar instantâneos de recursos de volume ou instantâneos criados pelo Image Builder e criar tags para recursos do Image Builder.

- Image Builder — O acesso é concedido para obter qualquer Image Builder ou AWS Marketplace componente.
- AWS KMS— O acesso é concedido para descriptografar um componente do Image Builder, se ele tiver sido criptografado via. AWS KMS
- Amazon S3 — O acesso é concedido para obter objetos armazenados em um bucket do Amazon S3 cujo nome comece `ec2imagebuilder-` com, ou recursos que tenham uma extensão de arquivo ISO.

Para visualizar as permissões para esta política, consulte EC2InstanceProfileForImageBuilder na Referência de políticas gerenciadas pela AWS .

Política EC2InstanceProfileForImageBuilderECRContainerBuilds

A EC2InstanceProfileForImageBuilderECRContainerBuildspolítica concede as permissões mínimas necessárias para uma EC2 instância ao trabalhar com o Image Builder para criar imagens do Docker e, em seguida, registrar e armazenar as imagens em um repositório de contêineres do Amazon ECR. Isso não inclui as permissões necessárias para usar o Systems Manager Agent.

Detalhes das permissões

Esta política inclui as seguintes permissões:

- CloudWatch Registros — O acesso é concedido para criar e carregar CloudWatch registros em qualquer grupo de registros cujo nome comece com`/aws/imagebuilder/`.
- Amazon ECR: o acesso é concedido ao Amazon ECR para obter, registrar e armazenar uma imagem de contêiner e obter um token de autorização.
- Image Builder: o acesso é concedido para obter um componente ou fórmula de contêiner do Image Builder.
- AWS KMS— O acesso é concedido para descriptografar um componente ou uma receita de contêiner do Image Builder, se ele tiver sido criptografado via. AWS KMS
- Amazon S3: o acesso é concedido para obter objetos armazenados em um bucket do Amazon S3 cujo nome começa com `ec2imagebuilder-`.

Para visualizar as permissões para esta política, consulte EC2InstanceProfileForImageBuilderECRContainerBuilds na Referência de políticas gerenciadas pela AWS .

Atualizações do Image Builder para políticas AWS gerenciadas

Esta seção fornece informações sobre atualizações nas políticas AWS gerenciadas do Image Builder desde que esse serviço começou a rastrear essas alterações. Para receber alertas automáticos sobre alterações realizadas nesta página, inscreva-se no feed RSS na página Histórico de documentos do Image Builder.

Alteração	Descrição	Data
AWSServiceRoleForImageBuilder: atualizar para uma política existente	O Image Builder fez as seguintes alterações na função de serviço para oferecer suporte ao uso dos parâmetros de armazenamento de parâmetros AWS Systems Manager (SSM) em receitas e durante a distribuição de imagens. • SSM adicionado: GetParameter para permitir que o Image Builder leia parâmetros SSM públicos e parâmetros SSM privados prefixados com `/imagebuilder/` para que possam ser usados em receitas. • Foi adicionado ssm: PutParameter para permitir que o Image Builder atualize parâmetros SSM privados prefixados `/imagebuilder/` com a AMI de saída que o Image IDs	23 de julho de 2025

Alteração	Descrição	Data
	Builder cria a partir de uma nova compilação.	
EC2InstanceProfileForImageBuilder – atualização para uma política existente	O Image Builder fez as seguintes alterações na política de perfil da instância para oferecer suporte a mais extensões de arquivo para downloads de arquivos ISO. • Foram adicionadas duas extensões de arquivo à lista de recursos de `s3:GetObject` ação: `"arn:aws:s3:::*/*.iso"` `"arn:aws:s3:::*/*.Iso"` e.	19 de maio de 2025
AWSServiceRoleForImageBuilder – atualização para uma política existente	O Image Builder fez as seguintes alterações na função de serviço para oferecer suporte à importação de arquivos ISO do sistema operacional cliente Microsoft como imagem base. • Foi adicionado ec2: RegisterImage para permitir que o Image Builder crie um instantâneo e crie e registre uma AMI cujo sistema operacional básico foi importado de arquivos de disco ISO verificados.	30 de dezembro de 2024

Alteração	Descrição	Data
EC2InstanceProfileForImageBuilder – atualização para uma política existente	O Image Builder fez as seguintes alterações na política de perfil da instância para oferecer suporte à criação de imagens a partir de arquivos de imagem de disco. • Foram adicionadas as seguintes ações ec2: DescribeVolumes e DescribeSnapshots em todos os recursos para recuperar detalhes. Também foram adicionadas as seguintes ações para recursos criados pelo Image Builder: CreateSnapshot para recursos de volume e instantâneo, e. CreateTags Adicionado s3: GetObject para recursos com uma extensão de arquivo “ISO”.	30 de dezembro de 2024
EC2InstanceProfileForImageBuilder: política atualizada	O Image Builder atualizou a `EC2InstanceProfileForImageBuilder` política para permitir que o Image Builder obtenha AWS Marketplace componentes.	2 de dezembro de 2024

Alteração	Descrição	Data
EC2ImageBuilderLifecycleExecutionPolicy – Nova política	O Image Builder adicionou a nova política `EC2ImageBuilderLifecycleExecutionPolicy` que contém permissões para o gerenciamento do ciclo de vida de imagem.	17 de novembro de 2023
AWSServiceRoleForImageBuilder – atualização para uma política existente	O Image Builder fez as alterações a seguir no perfil de serviço para oferecer compatibilidade com o posicionamento de instâncias. • Foi adicionado o ec2:DescribeHosts permite que o Image Builder pesquise o HostID para determinar quando ele está em um estado válido para iniciar uma instância. • Foi adicionada a ação de API ssm:GetCommandInvocation, para melhorar o método que o Image Builder usa para obter detalhes da invocação do comando.	19 de outubro de 2023

Alteração	Descrição	Data
AWSServiceRoleForImageBuilder – atualização para uma política existente	O Image Builder fez as alterações a seguir no perfil de serviço para oferecer compatibilidade com o posicionamento de instâncias. • Foi adicionado o ec2:DescribeHosts permite que o Image Builder pesquise o HostId para determinar quando ele está em um estado válido para iniciar uma instância. • Foi adicionada a ação de API ssm:GetCommandInvocation, para melhorar o método que o Image Builder usa para obter detalhes da invocação do comando.	28 de setembro de 2023

Alteração	Descrição	Data
AWSServiceRoleForImageBuilder – atualização para uma política existente	O Image Builder fez as seguintes alterações no perfil de serviço para permitir que os fluxos de trabalho do Image Builder coletassem descobertas de vulnerabilidade para compilações de imagens de contêiner AMI e ECR. As novas permissões são compatíveis com o atributo de detecção e geração de relatórios do CVE. • Foram adicionados inspetor2: ListCoverage e inspetor2: para permitir que o ListFindings Image Builder determine quando o Amazon Inspector conclui as varreduras de instâncias de teste e colete resultados para imagens que estão configuradas para permitir isso. • Foi adicionado ecr:CreateRepository, com a exigência de que o Image Builder marque o repositório com `CreatedBy: EC2 Image Builder` ()tag-on-create. Também foi adicionado ecr: TagResource (obrigatório para tag-	30 de março de 2023

Alteração	Descrição	Data
	on-create) com a mesma restrição de CreatedBy tag e uma restrição adicional que exige que o nome do repositório comece. `image-builder-*` A restrição de nome impede o aumento de privilégios e impede alterações nos repositórios que o Image Builder não criou. • Adicionado ecr: BatchDeleteImage para repositórios ECR marcados com. `CreatedBy: EC2 Image Builder` Essa permissão requer que o nome do repositório comece com `image-builder-*` . • Foram adicionadas permissões de eventos para o Image Builder criar e gerenciar regras EventBridge gerenciadas pela Amazon que incluem `ImageBuilder-*` no nome.	

Alteração	Descrição	Data
AWSServiceRoleForImageBuilder – atualização para uma política existente	O Image Builder fez as alterações a seguir no perfil de serviço: • Licenças do License Manager adicionadas como um recurso para a RunInstance chamada ec2: para permitir que os clientes usem imagens AMIs básicas associadas a uma configuração de licença.	22 de março de 2022
AWSServiceRoleForImageBuilder – atualização para uma política existente	O Image Builder fez as alterações a seguir no perfil de serviço: • Foram adicionadas permissões para a ação EC2 EnableFastLaunch da API, para ativar e desativar o lançamento mais rápido de instâncias do Windows. • Maior escopo para ec2: condições de tag de CreateTags ação e recurso.	21 de fevereiro de 2022

Alteração	Descrição	Data
AWSServiceRoleForImageBuilder – atualização para uma política existente	O Image Builder fez as alterações a seguir no perfil de serviço: • Foram adicionadas permissões para chamar o serviço VMIE para importar uma VM e criar uma AMI básica a partir dela. • Escopo reforçado para ec2: condições de tag de CreateTags ação e recurso.	20 de novembro de 2021
AWSServiceRoleForImageBuilder – atualização para uma política existente	O Image Builder adicionou novas permissões para corrigir problemas em que mais de uma associação de inventário faz com que a criação da imagem fique paralisada.	11 de agosto de 2021

Alteração	Descrição	Data
AWSImageBuilderFullAccess – atualização para uma política existente	O Image Builder fez as alterações a seguir no perfil de acesso total: • Permissões adicionadas para permitir `ec2:DescribeInstanceTypeOfferings` . • Foram adicionadas permissões para chamar `ec2:DescribeInstanceTypeOffereings` e permitir que o console do Image Builder corresponda com precisão aos tipos de instância disponíveis na conta.	13 de abril de 2021
O Image Builder começou a monitorar alterações	O Image Builder começou a monitorar as mudanças em suas políticas AWS gerenciadas.	02 de abril de 2021

Usar perfis do IAM vinculados ao serviço para o Image Builder

EC2 O Image Builder usa AWS Identity and Access Management funções vinculadas ao serviço (IAM). Um perfil vinculado ao serviço é um tipo exclusivo de perfil do IAM que é vinculado diretamente ao Image Builder. As funções vinculadas ao serviço são predefinidas pelo Image Builder e incluem todas as permissões que o serviço exige para ligar para outras pessoas Serviços da AWS em seu nome.

Um perfil vinculado ao serviço torna a configuração do Image Builder mais eficiente, porque você não precisa adicionar as permissões necessárias manualmente. O Image Builder define as permissões

de seus perfis vinculados ao serviço e, exceto se definido de outra forma, somente o Image Builder pode assumir seus perfis. As permissões definidas incluem a política de confiança e a política de permissões. A política de permissões não pode ser anexada a nenhuma outra entidade do IAM.

Para obter informações sobre outros serviços que são compatíveis com os perfis vinculados ao serviço, consulte Serviços da AWS Compatíveis com o IAM e procure os serviços que contêm Sim na coluna Perfil vinculado ao serviço. Escolha um Sim com um link para visualizar a documentação do perfil vinculado para esse serviço.

Permissões de perfil vinculado ao serviço para o Image Builder

O Image Builder usa a função AWSServiceRoleForImageBuildervinculada ao serviço para permitir que o EC2 Image Builder acesse AWS recursos em seu nome. O perfil vinculado ao serviço confia no serviço imagebuilder.amazonaws.com para assumir o perfil.

Você não precisa criar manualmente esse perfil vinculado ao serviço. Quando você cria sua primeira imagem do Image Builder no AWS Management Console, ou na AWS API AWS CLI, o Image Builder cria a função vinculada ao serviço para você.

As ações a seguir criam uma nova imagem:

- Execute o assistente de pipeline no console do Image Builder para criar uma imagem personalizada.
- Use uma das ações de API a seguir ou o AWS CLI comando correspondente:
  - A ação CreateImage da API (create-imageno AWS CLI).
  - A ação ImportVmImage da API (import-vm-imageno AWS CLI).
  - A ação StartImagePipelineExecution da API (start-image-pipeline-executionno AWS CLI).

> **Important**
>
> Se o perfil vinculado ao serviço for excluído da sua conta, você poderá usar o mesmo processo para criá-lo novamente. Quando você cria seu primeiro recurso do EC2 Image Builder, o Image Builder cria a função vinculada ao serviço para você novamente.

Para ver as permissões do AWSServiceRoleForImageBuilder, consulte a página Política AWSServiceRoleForImageBuilder. Para saber mais sobre como configurar permissões para um perfil vinculado ao serviço, consulte Permissões do perfil vinculado ao serviço no Guia do usuário do IAM.

Remover um perfil vinculado ao serviço do Image Builder da sua conta

Você pode usar o console do IAM AWS CLI, o ou a AWS API para remover manualmente a função vinculada ao serviço do Image Builder da sua conta. No entanto, antes de fazer isso, você deve garantir que não haja recursos habilitados do Image Builder que se refiram a ele.

> **Note**
>
> Se o serviço do Image Builder estiver usando o perfil quando você tenta excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Limpe os recursos do Image Builder usados pelo perfil **`AWSServiceRoleForImageBuilder`**

1. Verifique se não há compilações de pipeline em execução antes de começar. Para cancelar uma compilação em execução, use o comando `cancel-image-creation` do AWS CLI.

   ```
   aws imagebuilder cancel-image-creation --image-build-version-
   arn arn:aws:imagebuilder:us-east-1:123456789012:image-pipeline/sample-pipeline
   ```

2. Altere todas as programações do pipeline para usar um processo de compilação manual ou as exclua se você não quiser usá-las novamente. Para obter mais informações sobre exclusão de recursos, consulte Excluir recursos desatualizados ou não utilizados do Image Builder.

Excluir o perfil vinculado a serviço usando o IAM

Você pode usar o console do IAM AWS CLI, o ou a AWS API para excluir a `AWSServiceRoleForImageBuilder` função da sua conta. Para obter mais informações, consulte Excluir uma função vinculada ao serviço no Guia do usuário do IAM.

Regiões compatíveis com os perfis vinculados ao serviço do Image Builder

O Image Builder oferece suporte ao uso de funções vinculadas ao serviço em todas as AWS regiões em que o serviço está disponível. Para ver a lista de AWS regiões suportadas, consulteAWS Regiões e endpoints.

Solucionar problemas do IAM no Image Builder

Tópicos

- Não tenho autorização para executar uma ação no Image Builder
- Não estou autorizado a realizar iam: PassRole
- Quero permitir que pessoas de fora da minha acessem meus Conta da AWS recursos do Image Builder

Não tenho autorização para executar uma ação no Image Builder

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, suas políticas deverão ser atualizadas para permitir que você realize a ação.

O erro do exemplo a seguir ocorre quando o usuário do IAM `mateojackson` tenta usar o console para visualizar detalhes sobre um atributo *`my-example-widget`* fictício, mas não tem as permissões `imagebuilder:`*`GetWidget`* fictícias.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 imagebuilder:GetWidget on resource: my-example-widget
```

Nesse caso, a política do usuário `mateojackson` deve ser atualizada para permitir o acesso ao recurso *`my-example-widget`* usando a ação `imagebuilder:`*`GetWidget`*.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Não estou autorizado a realizar iam: PassRole

Se você receber uma mensagem de erro informando que não tem autorização para executar a ação `iam:PassRole`, as suas políticas deverão ser atualizadas para permitir a passagem de um perfil para o Image Builder.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O erro de exemplo a seguir ocorre quando uma usuária do IAM chamada `marymajor` tenta usar o console para executar uma ação no Image Builder. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que pessoas de fora da minha acessem meus Conta da AWS recursos do Image Builder

É possível criar um perfil que os usuários de outras contas ou pessoas fora da organização podem usar para acessar seus recursos. É possível especificar quem é confiável para assumir o perfil. Para serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se o Image Builder oferece suporte a esses atributos, consulte Como o Image Builder utiliza políticas e perfis do IAM.
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte Como fornecer acesso a um usuário do IAM em outro Conta da AWS que você possui no Guia do usuário do IAM.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Como fornecer acesso Contas da AWS a terceiros no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para saber a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Recursos de validação da conformidade do Image Builder

EC2 O Image Builder não está no escopo de nenhum programa de AWS conformidade.

Para obter uma lista do Serviços da AWS escopo de programas de conformidade específicos, consulte AWS Serviços no escopo por programa de conformidade Serviços da AWS . Para obter informações gerais, consulte Programas de AWS conformidade Programas AWS de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte Baixando relatórios no AWS Artifact .

Sua responsabilidade com relação à compatibilidade ao usar o Image Builder é determinada pela confidencialidade dos dados, pelos objetivos de compatibilidade da empresa e pelos regulamentos e leis aplicáveis. A AWS oferece os seguintes recursos para ajudar na compatibilidade:

- Guias de início rápido de segurança e compatibilidade: esses guias de implantação abordam as considerações de arquitetura e fornecem etapas para implantação de ambientes de linha de base focados em compatibilidade e segurança na AWS.
- AWS Recursos de https://aws.amazon.com/compliance/resources/ de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- Avaliação de recursos com regras no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- AWS Security Hub CSPM— Esse AWS serviço fornece uma visão abrangente do seu estado de segurança interno, AWS que ajuda você a verificar sua conformidade com os padrões e as melhores práticas do setor de segurança.

Você pode incorporar produtos de conformidade AWS Marketplace ou componentes de AWS Task Orchestrator and Executor (AWSTOE) em suas imagens do Image Builder para ajudar a garantir que suas imagens estejam em conformidade. Para obter mais informações, consulte Produtos de conformidade para suas imagens do Image Builder.

Redundância e resiliência de dados no Image Builder

A infraestrutura AWS global é construída em torno de AWS regiões e zonas de disponibilidade. AWS As regiões fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

O serviço EC2 Image Builder permite que você distribua imagens criadas em uma região com outras regiões, oferecendo resiliência multirregional para. AMIs Não há mecanismo para “fazer backup” de pipelines, fórmulas ou componentes de imagens. Você pode armazenar a fórmula e os documentos do componente fora do serviço Image Builder, como em um bucket do Amazon S3.

O EC2 Image Builder não pode ser configurado para alta disponibilidade (HA). Você pode distribuir imagens para várias regiões para tornar as imagens mais altamente disponíveis.

Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte Infraestrutura AWS global.

Segurança da infraestrutura no Image Builder

A rede AWS global fornece recursos de segurança e controla o acesso à rede para serviços como o EC2 Image Builder. Para obter mais informações sobre a segurança da infraestrutura que AWS fornece seus serviços, consulte a seção Segurança da infraestrutura no whitepaper Introdução à AWS segurança.

Para enviar solicitações por meio da rede AWS global para ações da API Image Builder, seu software cliente deve estar em conformidade com as seguintes diretrizes de segurança:

- Para enviar solicitações de ações da API Image Builder, o software cliente deve usar uma versão compatível do Transport Layer Security (TLS).

  > **Note**
  >
  > AWS está eliminando gradualmente o suporte para as versões 1.0 e 1.1 do TLS. É altamente recomendável que você atualize seu software cliente para usar o TLS versão 1.2 ou posterior para que você ainda possa se conectar. Para obter mais informações, consulte esta publicação do blog de Segurança da AWS.

- O software cliente também deve oferecer suporte a pacotes de criptografia com Perfect Forward Secrecy (PFS: sigilo de encaminhamento perfeito), como Ephemeral Diffie-Hellman (DHE) ou Ephemeral Elliptic Curve Diffie-Hellman (ECDHE). A maioria dos sistemas atuais, como Java 7 e posteriores, oferece suporte a esses modos.
- Você deve assinar suas solicitações de API com um ID de chave de acesso e uma chave de acesso secreta associada a um principal AWS Identity and Access Management (IAM). Ou é possível usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para suas solicitações.

Além disso, as EC2 instâncias que o Image Builder usa para criar e testar imagens devem ter acesso AWS Systems Manager.

Gerenciamento de patches para imagens do Image Builder

AWS fornece atualizações gerenciadas a AMIs cada mês com as atualizações e os patches de segurança mais recentes aplicados aos seguintes sistemas operacionais. Você pode usá-los AMIs como imagem base para suas personalizações. Para obter mais informações, consulte Sistemas operacionais compatíveis.

- Distribuições Linux, incluindo Amazon Linux AL2 2.023, Red Hat Enterprise Linux (RHEL), CentOS, Ubuntu, SUSE Linux Enterprise Server
- Windows Server 2016 e posterior
- macOS 10.14.x e posterior

Depois de criar uma imagem personalizada, você é responsável pela aplicação de patches no EC2 sistema da Amazon, de acordo com o modelo de responsabilidade compartilhada. Se as EC2 instâncias na carga de trabalho do seu aplicativo puderem ser facilmente substituídas, talvez seja mais eficiente atualizar a AMI básica e reimplantar todos os nós de computação com base nessa imagem.

> **Note**
>
> Quanto à aplicação de patches do macOS, recomendamos que você crie uma versão da sua fórmula que use a AMI gerenciada mais recente para a imagem de base e, em seguida, crie uma imagem personalizada atualizada com base na fórmula e em seus outros recursos de compilação de imagens. Se suas instâncias Mac não forem facilmente substituídas, consulte a página Atualizar o sistema operacional e o software em instâncias Mac no Guia EC2 do usuário da Amazon para obter mais informações.

A seguir estão duas maneiras de manter seu Image Builder AMIs atualizado.

- AWS-componentes de correção fornecidos — O EC2 Image Builder fornece os seguintes componentes de compilação que instalam todas as atualizações pendentes do sistema operacional:
  - `update-linux`

- `update-windows`

Esses componentes usam o módulo de ação `UpdateOS`. Para obter mais informações, consulte UpdateOS. Os componentes podem ser adicionados aos seus pipelines de criação de imagens selecionando-os na lista AWS de componentes fornecidos.

- Componentes de compilação personalizados com operações de correção — Para instalar ou atualizar seletivamente os patches em sistemas operacionais compatíveis AMIs, você pode criar um componente do Image Builder para instalar os patches necessários. Um componente personalizado pode instalar patches usando scripts de shell (Bash ou PowerShell) ou pode usar o módulo de `UpdateOS` ação para especificar patches para instalação ou exclusão. Para obter mais informações, consulte Módulos de ação compatíveis com o gerenciador de componentes do AWSTOE.

  Componente que usa o módulo de ação `UpdateOS`. (Somente Linux e Windows. O módulo de ação `UpdateOS` não é compatível com macOS.)

```
schemaVersion: 1.0
phases:
  - name: build
 steps:
   - name: UpdateOS
  action: UpdateOS
```

  Componente que usa o Bash para instalar atualizações do yum

```
schemaVersion: 1.0
phases:
  - name: build
 steps:
   - name: InstallYumUpdates
  action: ExecuteBash
  inputs:
    commands:
   - sudo yum update -y
```

Práticas recomendadas de segurança para o Image Builder

EC2 O Image Builder fornece vários recursos de segurança a serem considerados ao desenvolver e implementar suas próprias políticas de segurança. As práticas recomendadas a seguir são diretrizes gerais e não representam uma solução completa de segurança. Como essas práticas recomendadas podem não ser adequadas ou suficientes para o seu ambiente, trate-as como considerações úteis em vez de prescrições.

- Não use grupos de segurança excessivamente permissivos nas fórmulas do Image Builder.
- Não compartilhe imagens com contas nas quais você não confia.
- Não torne públicas imagens que tenham dados privados ou confidenciais.
- Aplique todos os patches de segurança disponíveis para Windows ou Linux durante a criação de imagens.
- Aplique periodicamente atualizações gerenciadas de AMIs às suas fórmulas do macOS e crie imagens para iniciar instâncias que tenham os patches de segurança mais recentes.

É altamente recomendável que você teste suas imagens para validar a postura de segurança e os níveis de conformidade de segurança aplicáveis. Soluções como o Amazon Inspector podem ajudar a validar a postura de segurança e conformidade das imagens.

IMDSv2 para pipelines do Image Builder

Quando o pipeline do Image Builder é executado, ele envia solicitações HTTP para iniciar EC2 instâncias que o Image Builder usa para criar e testar sua imagem. Para configurar a versão do IMDS que seu pipeline usa para as solicitações de lançamento, defina o `httpTokens` parâmetro nas configurações de metadados da instância de configuração da infraestrutura do Image Builder.

> **Note**
>
> Recomendamos que você configure todas as EC2 instâncias que o Image Builder executa a partir de um pipeline criado para uso, de IMDSv2 forma que as solicitações de recuperação de metadados da instância exijam um cabeçalho de token assinado.

Para obter mais informações sobre a configuração da infraestrutura do Image Builder, consulte Gerenciar a configuração da infraestrutura do Image Builder. Para obter mais informações sobre as opções de metadados da EC2 instância para imagens do Linux, consulte Configurar as opções de

metadados da instância no Guia EC2 do usuário da Amazon. Para imagens do Windows, consulte Configurar as opções de metadados da instância no Guia do EC2 usuário da Amazon.

Limpeza necessária após a construção

Depois que o Image Builder concluir todas as etapas de criação da sua imagem personalizada, o Image Builder prepara a instância de compilação para testes e criação de imagens. Antes de encerrar a instância de compilação para criar o snapshot, o Image Builder executa a seguinte limpeza para garantir a segurança da sua imagem:

Linux

O pipeline do Image Builder executa um script de limpeza para ajudar a garantir que a imagem final siga as melhores práticas de segurança e para remover quaisquer artefatos ou configurações de construção que não devam ser transferidos para seu snapshot. No entanto, você pode pular seções do script ou substituir totalmente os dados do usuário. Portanto, as imagens produzidas pelos pipelines do Image Builder não estão necessariamente em conformidade com nenhum critério regulatório específico.

Quando o pipeline conclui seus estágios de criação e teste, o Image Builder executa automaticamente o script de limpeza a seguir antes de criar a imagem de saída.

> **Important**
>
> Se você substituir os dados do usuário em sua fórmula, o script não será executado. Nesse caso, certifique-se de incluir um comando nos dados do usuário que crie um arquivo vazio chamado`perform_cleanup`. O Image Builder detecta esse arquivo e executa o script de limpeza antes de criar a nova imagem.

```
#!/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

Depois que o pipeline do Image Builder personalizar as imagens do Windows, ele executará o utilitário Sysprep da Microsoft. Essas ações seguem as AWS melhores práticas para endurecer e limpar a imagem.

macOS

O pipeline do Image Builder executa um script de limpeza para ajudar a garantir que a imagem final siga as melhores práticas de segurança e para remover quaisquer artefatos ou configurações de construção que não devam ser transferidos para seu snapshot. No entanto, você pode pular seções do script ou substituir totalmente os dados do usuário. Portanto, as imagens produzidas pelos pipelines do Image Builder não estão necessariamente em conformidade com nenhum critério regulatório específico.

Quando o pipeline conclui seus estágios de criação e teste, o Image Builder executa automaticamente o script de limpeza a seguir antes de criar a imagem de saída.

> **Important**
>
> Se você substituir os dados do usuário em sua fórmula, o script não será executado. Nesse caso, certifique-se de incluir um comando nos dados do usuário que crie um arquivo vazio chamado`perform_cleanup`. O Image Builder detecta esse arquivo e executa o script de limpeza antes de criar a nova imagem.

```
#!/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
```

Substitua o script de limpeza do Linux

O Image Builder cria imagens que são seguras por padrão e seguem nossas melhores práticas de segurança. No entanto, alguns casos de uso mais avançados podem exigir que você pule uma ou mais seções do script de limpeza integrado. Se você precisar pular parte da limpeza, é altamente recomendável testar sua AMI de saída para garantir a segurança da sua imagem.

> **Important**
>
> Ignorar seções no script de limpeza pode resultar na inclusão de informações confidenciais, como detalhes da conta do proprietário ou chaves SSH, na imagem final e, em qualquer instância, na execução dessa imagem. Você também pode ter problemas com o lançamento em diferentes zonas de disponibilidade, regiões ou contas.

A tabela a seguir descreve as seções do script de limpeza, os arquivos que são excluídos nessa seção e os nomes dos arquivos que você pode usar para sinalizar uma seção que o Image Builder deve ignorar. Para ignorar uma seção específica do script de limpeza, você pode usar o módulo de ação do CreateFile componente ou um comando nos dados do usuário (se estiver substituindo) para criar um arquivo vazio com o nome especificado na coluna Ignorar nome do arquivo da seção.

> **Note**
>
> Os arquivos que você cria para ignorar uma seção do script de limpeza não devem incluir uma extensão de arquivo. Por exemplo, se você quiser pular a `CLOUD_INIT_FILES` seção do script, mas criar um arquivo chamado`skip_cleanup_cloudinit_files.txt`, o Image Builder não reconhecerá o arquivo ignorado.

Input

Seção de limpeza	Arquivos removidos	Ignorar nome do arquivo da seção

`CLOUD_INIT_FILES`	`/etc/sudoers.d/90-cloud-init-users`
	
	`/etc/locale.conf`
		`skip_cleanup_cloud init_files`

Seção de limpeza	Arquivos removidos	Ignorar nome do arquivo da seção
	`/var/log/cloud-init.log` `/var/log/cloud-init-output.log`	

`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`
		`skip_cleanup_instance_files`

Seção de limpeza	Arquivos removidos	Ignorar nome do arquivo da seção
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
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

Seção de limpeza	Arquivos removidos	Ignorar nome do arquivo da seção
SSM_LOG_FILES	/var/log/amazon/ssm/*	skip_cleanup_ssm_log_files

Solução de problemas do Image Builder

EC2 O Image Builder se integra Serviços da AWS ao monitoramento e solução de problemas para ajudá-lo a solucionar problemas de criação de imagens. O Image Builder rastreia e exibe o progresso de cada etapa do processo de criação da imagem. Além disso, o Image Builder pode exportar logs para um local do Amazon S3 fornecido por você.

Para solucionar problemas avançados, você pode executar comandos e scripts predefinidos usando AWS Systems Manager Executar comando.

Conteúdo

- Solucionar problemas em compilação de pipelines
- Cenários de solução de problemas

Solucionar problemas em compilação de pipelines

Se a compilação do pipeline do Image Builder falhar, o Image Builder retornará uma mensagem de erro descrevendo a falha. O Image Builder também retorna um `workflow execution ID` na mensagem de falha, como o do exemplo de saída a seguir:

```
Workflow Execution ID: wf-12345abc-6789-0123-abc4-567890123abc failed with reason: …
```

O Image Builder organiza e direciona as ações de criação de imagens por meio de uma série de etapas que são definidas para os estágios de runtime em seu processo padrão de criação de imagem. Cada um dos estágios de compilação e teste do processo tem um fluxo de trabalho associado. Quando o Image Builder executa um fluxo de trabalho para criar ou testar uma nova imagem, ele gera um recurso de metadados do fluxo de trabalho que acompanha os detalhes do runtime.

As imagens de contêiner têm um fluxo de trabalho adicional que é executado durante a distribuição.

Pesquise detalhes sobre falhas de instância de runtime para seu fluxo de trabalho

Para solucionar uma falha de tempo de execução em seu fluxo de trabalho, você pode chamar as ações de ListWorkflowStepExecutionsAPI GetWorkflowExecutione de API com seu`workflow execution ID`.

Revise os logs de runtime

- CloudWatch Registros da Amazon

  O Image Builder publica registros detalhados de execução do fluxo de trabalho no seguinte grupo e stream de CloudWatch registros do Image Builder:

  Com o CloudWatch Logs, você pode pesquisar dados de registro com padrões de filtro. Para obter mais informações, consulte Pesquisar dados de log usando padrões de filtro no Guia do usuário do Amazon CloudWatch Logs.

- AWS CloudTrail

  Todas as atividades de criação também são registradas CloudTrail se estiverem ativadas em sua conta. Você pode filtrar CloudTrail eventos pela fonte`imagebuilder.amazonaws.com`. Como alternativa, você pode pesquisar o ID da EC2 instância da Amazon que é retornado no log de execução para ver mais detalhes sobre a execução do pipeline.

- Amazon Simple Storage Service (S3)

  Se você especificou um nome de bucket e um prefixo de chave do S3 na configuração da sua infraestrutura, o caminho do log de runtime da etapa do fluxo de trabalho segue esse padrão:

  ```
  S3://S3BucketName/KeyPrefix/ImageName/ImageVersion/ImageBuildVersion/
  WorkflowExecutionId/StepName
  ```

  Os registros que você envia para o bucket do S3 mostram as etapas e as mensagens de erro da atividade na EC2 instância durante o processo de criação da imagem. Os logs incluem saídas de log do gerenciador de componentes, as definições dos componentes que foram executados e a saída detalhada (em JSON) de todas as etapas realizadas na instância. Se você encontrar um problema, revise esses arquivos, começando por `application.log`, para diagnosticar a causa do problema na instância.

Por padrão, o Image Builder encerra a instância de EC2 compilação ou teste da Amazon que está em execução quando o pipeline falha. Você pode alterar as configurações da instância do recurso de configuração de infraestrutura que seu pipeline usa para reter sua instância de compilação ou teste para solução de problemas.

Para alterar as configurações da instância no console, você deve desmarcar a caixa de seleção Encerrar instância em caso de falha localizada na seção Configurações de solução de problemas do seu recurso de configuração de infraestrutura.

Você também pode alterar as configurações da instância com o update-infrastructure-configuration comando no AWS CLI. Defina o valor de `terminateInstanceOnFailure` para `false` no arquivo JSON ao qual o comando faz referência com o parâmetro `--cli-input-json`. Para obter detalhes, consulte Atualizar uma configuração de infraestrutura.

Cenários de solução de problemas

Esta seção lista os seguintes cenários detalhados de solução de problemas:

- Acesso negado: código de status 403
- O tempo limite de compilação é verificado ao verificar a disponibilidade do Systems Manager Agent na instância de compilação
- O disco secundário do Windows está off-line no lançamento
- A compilação falha com a imagem base reforçada do CIS
- AssertInventoryCollection falha (Systems Manager Automation)

Para ver os detalhes de um cenário, escolha o título do cenário para expandi-lo. É possível ter vários títulos expandidos ao mesmo tempo.

Acesso negado: código de status 403

Descrição

A construção do pipeline falha com "AccessDenied: Código de status de acesso negado: 403".

Causa

As possíveis causas incluem:

- O perfil da instância não tem as permissões necessárias para acessar APIs ou compor recursos.
- Aefunção do perfil de instância não tem as permissões necessárias para fazer login no Amazon S3. Geralmente, isso ocorre quando a função do perfil da instância não tem PutObjectpermissões para seus buckets do S3.

Solução

Dependendo da causa, esse problema pode ser resolvido da seguinte maneira:

- O perfil de instância não tem políticas gerenciadas: adicione as políticas ausentes à sua função de perfil da instância. Em seguida, execute o pipeline novamente.
- O perfil da instância não tem permissões de gravação para o bucket do S3 — adicione uma política à sua função de perfil da instância que conceda PutObjectpermissões para gravar no bucket do S3. Em seguida, execute o pipeline novamente.

O tempo limite de compilação é verificado ao verificar a disponibilidade do Systems Manager Agent na instância de compilação

Descrição

A construção do pipeline falha com “status = 'TimedOut'” e “mensagem de falha = 'A etapa expirou enquanto a etapa está verificando a disponibilidade do Systems Manager Agent na (s) instância (s) de destino'”.

Causa

As possíveis causas incluem:

- A instância que foi iniciada para realizar as operações de compilação e executar os componentes não conseguiu acessar o endpoint do Systems Manager.
- O perfil de instância não tem as permissões necessárias.

Solução

Dependendo da causa possível, esse problema pode ser resolvido da seguinte maneira:

- Problema de acesso, sub-rede privada — Se você estiver criando em uma sub-rede privada, certifique-se de ter configurado PrivateLink endpoints para Systems Manager, Image Builder e, se quiser fazer login, Amazon S3/. CloudWatch Para obter mais informações sobre como configurar PrivateLink endpoints, consulte Acesse AWS serviços por meio AWS PrivateLink de.
- Permissões ausentes: adicione as seguintes políticas gerenciadas à sua função vinculada ao serviço do IAM para o Image Builder:
  - EC2InstanceProfileForImageBuilder

- EC2InstanceProfileForImageBuilderECRContainerConstrói
- Amazon SSMManaged InstanceCore

Para obter mais informações sobre a função vinculada ao serviço do Image Builder, consulte Usar perfis do IAM vinculados ao serviço para o Image Builder.

O disco secundário do Windows está off-line no lançamento

Descrição

Quando o tipo de instância usado para criar uma AMI do Windows do Image Builder não corresponde ao tipo de instância usado para iniciar a partir da AMI, pode ocorrer um problema em que os volumes não raiz estejam off-line na inicialização. Isso acontece principalmente quando a instância de compilação está usando uma arquitetura mais recente do que a instância de execução.

O exemplo a seguir demonstra o que acontece quando uma AMI do Image Builder é criada em um tipo de instância EC2 Nitro e executada em uma instância EC2 Xen:

Tipo de instância de compilação: m5.large (Nitro)

Tipo de instância de execução: 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
```

Causa

Devido às configurações padrão do Windows, os discos recém-descobertos não são automaticamente colocados on-line e formatados. Quando o tipo de instância é alterado EC2, o Windows trata isso como novos discos sendo descobertos. Isso se deve à mudança de driver subjacente.

Solução

Recomendamos que você use o mesmo sistema de tipos de instância ao criar sua AMI do Windows a partir da qual você pretende executar. Não inclua tipos de instância criados em sistemas diferentes na configuração da sua infraestrutura. Se algum dos tipos de instância que você especificar usar o sistema Nitro, todos eles devem usar o sistema Nitro.

Para obter mais informações sobre instâncias criadas no sistema Nitro, consulte Instâncias criadas no sistema Nitro no Guia do EC2 usuário da Amazon.

A compilação falha com a imagem base reforçada do CIS

Descrição

Você está usando uma imagem base reforçada do CIS e a compilação falha.

Causa

Quando o diretório `/tmp` é classificado como `noexec`, isso pode causar falha no Image Builder.

Solução

Escolha um local diferente para seu diretório de trabalho no campo `workingDirectory` da fórmula da imagem. Para obter mais informações, consulte a descrição do tipo de ImageRecipedados.

AssertInventoryCollection falha (Systems Manager Automation)

Descrição

O Systems Manager Automation mostra uma falha na etapa `AssertInventoryCollection` de automação.

Causa

Você ou sua organização podem ter criado uma associação do Systems Manager State Manager que coleta informações de inventário para EC2 instâncias. Se a coleta aprimorada de metadados de imagem estiver habilitada para seu pipeline do Image Builder (esse é o padrão), o Image Builder tentará criar uma nova associação de inventário para a instância de compilação. No entanto, o Systems Manager não permite várias associações de inventário para instâncias gerenciadas e impede uma nova associação, caso já exista. Isso faz com que a operação falhe e resulta em uma falha na construção do pipeline.

Solução

Para resolver esse problema, desative a coleta aprimorada de metadados de imagem usando um dos seguintes métodos:

- Atualize seu pipeline de imagens no console para desmarcar a caixa de seleção Ativar coleta avançada de metadados. Salve as alterações e execute uma compilação do pipeline.

  Para obter mais informações sobre como atualizar seu pipeline de imagens da AMI usando o console do EC2 Image Builder, consulteAtualizar pipelines de imagens de AMI com o console. Para obter mais informações sobre como atualizar seu pipeline de imagens de contêiner usando o console do EC2 Image Builder, consulteAtualizar um pipeline de imagens de contêiner com o console.

- Você pode atualizar seu pipeline de imagem com o comando update-image-pipeline na AWS CLI. Para fazer isso, inclua a propriedade `EnhancedImageMetadataEnabled` em seu arquivo JSON, definida como `false`. O exemplo a seguir mostra a propriedade definida como `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"
}
```

Para evitar que isso aconteça com novos pipelines, desmarque a caixa de seleção Habilitar coleta avançada de metadados ao criar um novo pipeline usando o console do EC2 Image Builder ou defina o valor da `EnhancedImageMetadataEnabled` propriedade em seu arquivo JSON como `false` quando você cria seu pipeline usando o. AWS CLI

Histórico da documentação de alterações no Guia do usuário do Image Builder

A tabela a seguir descreve alterações importantes feitas na documentação por data. Para receber notificações sobre atualizações dessa documentação, é possível inscrever-se em um feed RSS.

- Versão da API: 2025-04-30

Alteração	Descrição	Data
Aprimoramentos na distribuição de imagens	Os aprimoramentos incluem a capacidade de realizar atividades de distribuição diretamente sem a necessidade de executar novamente uma criação de imagem completa.	20 de novembro de 2025
Versionamento automático e aprimoramentos de IaC	O incremento automático de versões e a referência de versões curinga agora eliminam o gerenciamento manual de versões para receitas, componentes e fluxos de trabalho. Experiência em testes de componentes com execução a seco e capacidade aprimorada de criação no console.	20 de novembro de 2025
O fluxo de trabalho de imagem suporta Lambda e Step Functions	O fluxo de trabalho de imagem agora suporta a invocação de funções Lambda e a execução da máquina de estado Step Functions, permitindo fluxos de trabalho complexos de	13 de novembro de 2025

	várias etapas e lógica de validação personalizada na criação de imagens.	
Aprimoramentos do pipeline de imagens	Os aprimoramentos incluem suporte para registros de execução de pipeline, grupos de CloudWatch registros configuráveis e configuração para desativar automaticamente pipelines programados que falham repetidamente. As atualizações adicionais incluem informações expandidas do cronograma do pipeline e exemplos atualizados de mensagens de eventos do CloudWatch Logs.	29 de setembro de 2025
Novo script para Amazon Linux 2023	Foi adicionado suporte para o sistema operacional SUSE Linux Enterprise Server (SLES). Versões do STIG e STIGS aplicado para a versão do terceiro trimestre de 2025 para todos os níveis de conformidade ()low/medium/high.	4 de setembro de 2025
Novo script para Amazon Linux 2023	Foi adicionado um novo script especificamente adaptado ao Amazon Linux 2023.	26 de agosto de 2025

Atualização da política do IAM: política de função de serviço	Atualizou a política de função vinculada ao serviço para oferecer suporte ao uso de parâmetros SSM em receitas e durante a distribuição de imagens. Para obter mais informações, consulte a política AWSServiceRoleForImageBuilder.	23 de julho de 2025
Atualizações do STIG Q2 2025	Versões atualizadas do Windows STIG e STIGS aplicado para a versão do segundo trimestre de 2025.	26 de junho de 2025
Atualização da política do IAM: política de perfil de instância	Adicione `.iso` e `.Iso` como extensões de arquivo válidas para baixar imagens de disco ISO do Amazon S3. Para obter mais informações, consulte a política EC2InstanceProfileForImageBuilder.	19 de maio de 2025
Atualizações do STIG Q1 2025	Versões atualizadas do Windows STIG e versões aplicadas dos componentes STIGS e SCAP para a versão do primeiro trimestre de 2025.	5 de maio de 2025
Lançamento de recursos: Support SSM Parameters	O Image Builder agora suporta o uso de parâmetros de armazenamento de parâmetros AWS Systems Manager(SSM) em receitas e durante a distribuição de imagens.	30 de abril de 2025

Atualizações do STIG no quarto trimestre	Versões atualizadas do Windows STIG e STIGS aplicado para a versão do quarto trimestre de 2024.	4 de fevereiro de 2025
Atualização da política do IAM: política de função de serviço	Atualizou a política de função vinculada ao serviço para oferecer suporte à criação de imagens a partir de arquivos ISO do sistema operacional do cliente Microsoft oficiais importados. Para obter mais informações, consulte a política AWSServiceRoleForImageBuilder.	30 de dezembro de 2024
Atualização da política do IAM: política de perfil de instância	Atualizou a política de função do perfil da instância para oferecer suporte à criação de imagens a partir de arquivos de imagem de disco. Para obter mais informações, consulte a política EC2InstanceProfileForImageBuilder.	30 de dezembro de 2024
Lançamento do recurso: ISO para AMI	O Image Builder agora pode criar uma imagem a partir de arquivos de disco ISO oficiais para sistemas operacionais cliente Microsoft Windows 11 e versões posteriores.	30 de dezembro de 2024

Atualizações do STIG no quarto trimestre	Versões Linux STIG atualizadas e STIGS aplicado para a versão do quarto trimestre de 2024. Foram adicionadas informações sobre dois novos parâmetros de entrada para os componentes Linux.	10 de dezembro de 2024
Atualização da política do IAM: política de perfil de instância	Atualizou a política de perfil da instância para conceder acesso para obter AWS Marketplace componentes. Para obter mais informações, consulte a política EC2InstanceProfileForImageBuilder.	2 de dezembro de 2024
Lançamento de recursos: AWS Marketplace componentes	Foi adicionado suporte para componentes AWS Marketplace de software.	2 de dezembro de 2024
Lançamento de recursos: suporte para macOS	Foi adicionado suporte para imagens do macOS.	22 de outubro de 2024
Suporte de documentos para operadores lógicos em AWS Task Orchestrator and Executor	Use operadores lógicos para adicionar ou modificar expressões condicionais no documento do componente.	16 de agosto de 2024
Suporte de documentos para construções condicionais em AWS Task Orchestrator and Executor	Use constructos condicionais, como a instrução “if”, para direcionar o fluxo de controle das ações do componente no respectivo documento.	16 de agosto de 2024
Suporte de documentos para operadores de comparação em AWS Task Orchestrator and Executor	Use operadores de comparação para comparar valores no documento do componente.	16 de agosto de 2024

Módulo de ação Assert adicionado	Foi adicionada documentação para o módulo de ação `ExecuteDocument` em `General execution` .	14 de agosto de 2024
Compatibilidade com sistema operacional	Compatibilidade com RHEL 9 e Ubuntu 24.04 LTS adicionada.	2 de agosto de 2024
Atualização do documento: reorganização do conteúdo	Documentação reorganizada para melhorar a apresentação e a navegação.	21 de junho de 2024
Atualizações do STIG no segundo trimestre	Versões do Linux STIG atualizadas e STIGS aplicados para o lançamento do segundo trimestre de 2024. Compatibilidade com RHEL 9, CentOS Stream 9 e Ubuntu 22.04 adicionada. Não houve alterações nas versões do Windows.	10 de maio de 2024
Atualizações do STIG no primeiro trimestre	Versões do Linux STIG atualizadas e STIGS aplicados para o lançamento do primeiro trimestre de 2024. Não houve alterações nas versões do Windows.	23 de fevereiro de 2024
Atualizações do STIG no primeiro trimestre	Versões do Linux STIG atualizadas e STIGS aplicados para o lançamento do primeiro trimestre de 2024. Não houve alterações nas versões do Windows.	23 de fevereiro de 2024

Lançamento de recurso: gerenciamento de fluxo de trabalho de imagem	Com os fluxos de trabalho de imagem, você tem mais flexibilidade, visibilidade e controle sobre o processo de criação de imagens. Você pode personalizar as etapas de compilação e teste para seus fluxos de trabalho ou usar o fluxo de trabalho padrão do Image Builder.	12 de dezembro de 2023
Atualizações do STIG no quarto trimestre	Versões do Linux STIG atualizadas e STIGS aplicados para o lançamento do quarto trimestre de 2023. Não houve alterações nas versões do Windows. O SCAP para Linux e Windows também foi atualizado para novos componentes, software e números de referência.	7 de dezembro de 2023
Lançamento de recurso: gerenciamento do ciclo de vida de imagem	Com políticas e regras de gerenciamento do ciclo de vida de imagem, você pode definir sua estratégia de gerenciamento de recursos a fim de garantir que imagens desatualizadas e seus recursos associados passem por um processo de marcação e remoção.	17 de novembro de 2023

Atualização da política do IAM: perfil de serviço	Política de perfil vinculado ao serviço atualizada para suporte a posicionamento de instância. Para obter mais informações, consulte a política AWSServiceRoleForImageBuilder.	19 de outubro de 2023
Atualizações do terceiro trimestre do STIG	Versões do STIG atualizadas e STIGS aplicados para o release do terceiro trimestre de 2023. Mensagens adicionalmente atualizadas para esclarecer que pacotes de terceiros não são instalados automaticamente, com pouquíssimas exceções. Todos os ignorados STIGs são registrados.	5 de outubro de 2023
Novas versões do STIG	Versões do STIG atualizadas e STIGS aplicados para o release do segundo trimestre de 2023.	3 de maio de 2023
Novas versões do STIG	Versões do STIG atualizadas e STIGS aplicados para o release do primeiro trimestre de 2023. Foi adicionado suporte para AL2 023.	14 de abril de 2023

Atualize as regiões suportadas para AWSTOE	Foi adicionado AWSTOE suporte para o seguinte Regiões da AWS: Ásia-Pacífico (Hyderabad), Ásia-Pacífico (Jacarta), Europa (Zurique), Europa (Espanha) e Oriente Médio (EAU).	13 de abril de 2023
AWSTOE atualizações de download de aplicativos	Atualizou a assinatura para o download da AWSTOE instalação no Windows. Além disso, o TLS atualizado, observe que os downloads de aplicativos de buckets do S3 agora exigem a versão 1.2 ou posterior do TLS.	31 de março de 2023
Release do atributo: fluxos de trabalho de compilação aprimorados	Foram adicionados detalhes de runtime para compilações de imagem na nova guia de fluxo de trabalho nos detalhes da versão de compilação da imagem. Informações aprimoradas para solução de problemas de compilações.	30 de março de 2023

Release do atributo: detecção e geração de relatórios de CVE	Para contas que ativaram varreduras do Amazon Inspector, o Image Builder pode capturar as descobert as de vulnerabilidades e exposições comuns (CVE) do Amazon Inspector durante a etapa de teste do processo de compilação de novas imagens, incluindo imagens de contêiner armazenadas no Amazon ECR. O Image Builder cria um snapshot das descobertas para apoiar a análise de detalhes. O Image Builder também relata as contagens de descobertas que podem ser filtradas por conta, por pipeline ou por imagem, com a capacidade de aprofundar os detalhes.	30 de março de 2023
Histórico de versões adicionad o	Histórico de versões adicionad o às seções do Windows e Linux.	17 de fevereiro de 2023
Novas versões do STIG	Versões do STIG atualizad as e STIGS aplicados para o release do quarto trimestre de 2022.	1° de fevereiro de 2023

Lançamento de recursos: AWS Marketplace integração e fortalecimento do CIS	AWS Marketplace Integração adicional para encontrar e usar facilmente uma imagem inscrita como base para uma nova imagem personalizada, incluindo imagens CIS reforçadas e um novo componente CIS Hardening do Center for Internet Security.	13 de janeiro de 2023
Componentes de fortalecimento do CIS	Adicionados componentes de fortalecimento do CIS que são de propriedade e mantidos pelo CIS.	13 de janeiro de 2023
Novas versões do STIG	Suporte do Ubuntu introduzido, versões do STIG atualizadas e STIGS aplicados para o release do segundo trimestre de 2022.	20 de julho de 2022
Atualização do documento: navegação para a página do documento Criar componente YAML	Movido o conteúdo do documento Criar componente YAML para sua própria página e atualizadas outras páginas para referenciá-lo.	7 de junho de 2022
Novas versões do STIG	Versões do STIG atualizadas e STIGS aplicados para o release do primeiro trimestre de 2022.	25 de abril de 2022
Módulo ExecuteDocument de ação adicionado	Foi adicionada documentação para o módulo de ação `ExecuteDocument` em `General execution`.	28 de março de 2022

Release do atributo: oferecer suporte à inicialização mais rápida da AMI do Windows	Foram adicionadas configurações de distribuição para oferecer suporte ao lançamento mais rápido do Windows AMIs.	21 de fevereiro de 2022
Versão de manutenção: atualize a impressão AWSTOE digital binária	Impressão digital binária atualizada para o certificado do AWSTOE assinante.	18 de fevereiro de 2022
Versão do recurso: configure a entrada para AWSTOE	Foi adicionado suporte para usar um arquivo de configuração JSON como entrada para o AWSTOE run comando.	3 de fevereiro de 2022
Novas versões do STIG	Versões do STIG atualizadas e STIGS aplicados para o release do quarto trimestre de 2021. Também foi adicionada uma seção para os novos componentes do SCAP Compliance Checker (SCC).	22 de dezembro de 2021
Lançamento do recurso: integração com VM Import/Export (VMIE)	Foi adicionado suporte para importação de VMs por meio de todos os canais (console, API/CLI, etc.), and for VM export via API/CLI. Atualmente, a exportação de VM não está disponível no console do Image Builder.	20 de dezembro de 2021
Versão do recurso: compartilhamento de AMI para AWS Organizations e OUs	Configuração de distribuição atualizada para adicionar suporte ao compartilhamento de saída AMIs com AWS Organizations OUs e.	24 de novembro de 2021

Atualização do documento: atualizar os estágios e fases dos componentes	Conteúdo expandido para estágios de componentes no Image Builder e como eles interagem com as fases dos AWSTOE componentes.	22 de setembro de 2021
Atualização do documento : adicionar conteúdo de CloudTrail integração	Resumo de monitoramento e conteúdo de CloudTrail integração adicionados.	17 de setembro de 2021
Novas versões do STIG	Versões do STIG atualizad as e STIGS aplicados para o release do terceiro trimestre de 2021.	10 de setembro de 2021
Lançamento de recursos: EventBridge integração com a Amazon	EventBridge Suporte adicional que permite conectar o Image Builder a eventos relaciona dos Serviços da AWS e iniciar eventos com base nas regras definidas em EventBridge.	18 de agosto de 2021
Atualização do documento: reordenar páginas AWSTOE	AWSTOE Páginas reorganiz adas para maior clareza.	11 de agosto de 2021

Release do atributo: componentes parametrizados e configuração de instância adicional	Adicionado suporte para especificar parâmetros para personalizar componentes para fórmulas. Configuração expandida das EC2 instâncias usadas para criar e testar imagens, incluindo a capacidade de especificar comandos a serem executados na inicialização e mais controle sobre a instalação e remoção do agente do Systems Manager.	7 de julho de 2021
Novas versões do STIG	Versões do STIG atualizadas e STIGS aplicados para o release do segundo trimestre de 2021.	30 de junho de 2021
Aprimoramento: aprimoramentos de marcação	Mensagens aperfeiçoadas sobre marcação de recursos.	25 de junho de 2021
Release do atributo: inicializar a integração do modelo	Foi adicionado suporte para o uso de modelos de EC2 lançamento da Amazon para distribuição de AMI nas configurações de distribuição.	7 de abril de 2021
Release do atributo: aprimoramentos da compilação de contêineres	Foi adicionado suporte para configurar mapeamentos de dispositivos de blocos e especificar o uso como imagem base AMIs para compilações de contêineres.	7 de abril de 2021
Novas versões do STIG	Versões do STIG atualizadas e STIGs aplicados.	5 de março de 2021

Atualizadas as expressões cron	O processamento do cron do Image Builder é atualizado para aumentar a granularidade da expressão cron a cada minuto e usar um mecanismo de agendamento de cron padrão. Os exemplos são atualizados com o novo formato.	8 de fevereiro de 2021
Release do atributo: suporte para contêineres	Adicionado suporte para criar imagens de contêiner do Docker usando o Image Builder, com registro e armazenamento das imagens resultantes no Amazon Elastic Container Registry (Amazon ECR). O conteúdo foi reorganizado para refletir nova funcionalidade e acomodar o crescimento futuro.	17 de dezembro de 2020
Documentação do cron reestruturada	Agora, esta página destaca mais informações sobre como o cron funciona com as compilações de pipeline do Image Builder e inclui detalhes sobre o horário UTC. Os curingas que não são permitidos em campos específicos foram removidos. Os exemplos agora incluem amostras de expressões para console e para CLI.	13 de novembro de 2020

Versão 2.0 do console: edição de pipeline atualizada	Alterações do conteúdo de conceitos básicos e tutoriais de pipeline de criação, além da página de pipelines de imagem de gerenciamento, para incorporar novos atributos e fluxo do console.	13 de novembro de 2020
Novas versões do STIG	Versões do STIG atualizadas e STIGs aplicados. Nota - o formato da lista foi alterado para mostrar STIGs que são aplicados por padrão.	15 de outubro de 2020
Support para construções em loop em AWSTOE	Criar estruturas em loop para definir uma sequência repetida de instruções no aplicativo AWSTOE .	29 de julho de 2020
Support para o desenvolvimento local de AWSTOE componentes	Desenvolva e teste componentes de imagem localmente com o AWSTOE aplicativo.	28 de julho de 2020
Encriptado AMIs	EC2 O Image Builder adiciona suporte para distribuição criptografada de AMI.	1.º de julho de 2020
AutoScaling depreciação	Depreciação do uso de AutoScaling para iniciar instâncias.	15 de junho de 2020

Support para conectividade por meio de AWS PrivateLink	Você pode estabelecer uma conexão privada entre sua VPC e o EC2 Image Builder criando uma interface VPC endpoint. Os endpoints de interface são alimentados por AWS PrivateLink uma tecnologia que permite acessar de forma privada o Image Builder APIs sem um gateway de internet, dispositivo NAT, conexão VPN ou conexão AWS Direct Connect. As instâncias em sua VPC não precisam de endereços IP públicos para se comunicar com o Image Builder. APIs O tráfego entre sua VPC e o Image Builder não deixa a rede da Amazon.	10 de junho de 2020
Novas versões do STIG	Versões do STIG atualizadas e STIGs aplicados.	23 de janeiro de 2020
Solução de problemas	Adicionados cenários gerais de solução de problemas.	22 de janeiro de 2020
Componentes do STIG	Você pode criar imagens compatíveis com STIG com componentes STIG. AWSTOE	22 de janeiro de 2020

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.