Manual do usuário

EC2 Image Builder

EC2 Image Builder: Manual do usuário

Table of Contents

O que é o EC2 Image Builder?

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

1. Escolha uma imagem de base para suas personalizações.
2. Adicione ou remova um software da sua imagem de base.
3. Personalize configurações e scripts com componentes de compilação.
4. Execute testes selecionados ou crie componentes de teste personalizados.
5. Distribua AMIs para Regiões da AWS e. Contas da AWS
6. Se o pipeline do Image Builder criar uma Amazon Machine Image (AMI) personalizada para distribuição, você poderá autorizar outras Contas da AWS organizações e OUs a iniciá-la a partir da sua conta. Sua conta é cobrada pelas cobranças associadas à AMI.

O Image Builder se integra Serviços da AWS ao seguinte para fornecer métricas, registros e monitoramento detalhados de eventos. Essas informações o ajudam a rastrear sua atividade, solucionar problemas de compilação de imagem e criar automações com base em notificações de eventos.

Conteúdo da seção

- Relacionado Serviços da AWS

Atributos do EC2 Image Builder

O EC2 Image Builder fornece os seguintes atributos:

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 fortalecimento do STIG gerenciados pela Amazon para o EC2 Image Builder.

Aplicação centralizada e rastreamento de linhagem

Usando integrações integradas com AWS Organizations, o Image Builder permite aplicar políticas que restringem as contas a executar instâncias somente de AMIs aprovadas.

Compartilhamento simplificado de recursos entre Contas da AWS

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

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

Para ter mais informações, consulte Compartilhar recursos do EC2 Image Builder.

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
Red Hat Enterprise Linux (RHEL)	7 e 8
SUSE Linux Enterprise Server (SUSE)	12 e 15
Ubuntu	18.04 LTS, 20.04 LTS e 22.04 LTS
Windows Server	2012 R2, 2016, 2019 e 2022

Formatos de imagem compatíveis

Para suas imagens AMI personalizadas, 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.

Conceitos

Os termos e conceitos a seguir são fundamentais para o entendimento e uso do EC2 Image Builder.

AMI

Uma imagem de máquina da Amazon (AMI) é a unidade básica de implantação no 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 implantar instâncias do EC2. Para obter mais informações, consulte Imagens de máquina da Amazon (AMIs).

Pipeline de imagens

Um pipeline de imagens fornece uma estrutura de automação para compilar AMIs seguras e imagens de contêiner em 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 receita de imagem para duplicar criações. 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 AWS Task Orchestrator and Executor gerenciador de 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

O EC2 Image Builder tem dois estágios de runtime: 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 compilação e teste. Os contêineres têm um fluxo de trabalho adicional para 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 o fluxo de trabalho de distribuição de imagens de contêiner.

Definição de preço

Não há custo para usar o EC2 Image Builder para criar AMIs ou imagens de contêiner personalizadas. 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.

- Inicializar uma instância do EC2
- Armazenar logs no Amazon S3
- Validar imagens com o Amazon Inspector
- Armazenar snapshots do Amazon EBS para suas 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 instâncias do Amazon EC2 forem executadas com ativação on-premises, você poderá ser cobrado pelos recursos por meio do Systems Manager

Relacionado Serviços da AWS

O EC2 Image Builder usa Serviços da AWS outros para criar imagens. Dependendo da sua configuração de fórmula de imagem ou de fórmula de contêiner do Image Builder, os serviços a seguir podem ser usados.

AWS License Manager

AWS License Manager permite criar e aplicar configurações de licença a partir de um repositório de configuração de licença de conta. Para cada AMI, você pode usar o Image Builder para anexar a uma configuração de licença preexistente à qual você Conta da AWS tem acesso como parte do fluxo de trabalho do Image Builder. As configurações de licença só podem ser aplicadas às AMIs. O Image Builder pode usar somente configurações de licença preexistentes e não pode criar ou modificar diretamente as configurações de licença. As configurações do License Manager não serão replicadas entre as Regiões da AWS que devem ser ativadas em sua conta, por exemplo, entre as `ap-east-1` regiões (Ásia-Pacífico: Hong Kong) e `me-south-1` (Oriente Médio: Bahrein).

AWS Organizations

AWS Organizations permite que você aplique Políticas de Controle de Serviços (SCP) em contas em 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 modelos de SCPs para cenários comuns, como impor restrições às contas dos membros para iniciar instâncias somente com AMIs aprovadas.

Amazon Inspector

O Image Builder usa o Amazon Inspector como o atendente padrão de verificação de vulnerabilidades para estabelecer linhas de base de segurança para o Amazon Linux 2, Windows Server 2012 e Windows Server 2016. Para obter mais informações, consulte O que é o Amazon Inspector?

AWS Resource Access Manager

AWS Resource Access Manager (AWS RAM) permite que você compartilhe seus 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. O EC2 Image Builder permite o compartilhamento dos seguintes recursos: componentes, imagens e fórmulas de imagem. 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 Compartilhar recursos do EC2 Image Builder.

CloudWatch Registros da Amazon

Você pode usar o Amazon CloudWatch Logs para monitorar, armazenar e acessar seus arquivos de log de instâncias do EC2 AWS CloudTrail, Amazon Route 53 e outras fontes.

Amazon Elastic Container Registry (Amazon ECR)

O Amazon ECR é um serviço gerenciado de registro de imagens de AWS contêineres 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.

Como o EC2 Image Builder funciona

Quando você usa o assistente do console de pipeline do EC2 Image Builder para criar uma imagem personalizada, um assistente o orienta nas 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. Definir configuração de infraestrutura – O Image Builder inicia instâncias do EC2 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 fórmula que pode ser usada com sistemas de controle de versão existentes e pipelines de integração contínua/implantação contínua (CI/CD) para automação repetível. Você pode compartilhar e criar novas versões da sua fórmula.

Conteúdo da seção

Elementos da AMI

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

Uma AMI inclui os seguintes elementos:

- Um modelo para o volume raiz da VM. Quando você executa uma VM do Amazon EC2, o volume do dispositivo raiz contém a imagem usada para iniciar 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 mais informações, consulte Volume do dispositivo raiz do Amazon EC2.
- Quando o Amazon EBS é usado, o dispositivo raiz é um volume do EBS criado em um snapshot do EBS.
- Permissões de inicialização que determinam Contas da AWS quem pode iniciar 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.

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.

Gerenciamento de componentes

O EC2 Image Builder usa um AWS Task Orchestrator and Executor aplicativo de gerenciamento de componentes 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, consulteComece 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, consulteGerencie componentes com o Image Builder. Para obter mais informações sobre a criação de componentes com AWSTOE, consulte AWS Task Orchestrator and Executor gerenciador de 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.

Versionamento semântico

O Image Builder usa versionamento semântico para organizar recursos e garantir que eles tenham IDs exclusivos. 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. ARNs sem versão e ARNs de nome não incluem valores específicos em nenhum dos nós. Os nós são totalmente excluídos ou são especificados como curingas, por exemplo: x.x.x.
2. Os ARNs da versão têm apenas os três primeiros nós: <major>.<minor>.<patch>
3. Os ARNs da versão da compilação têm todos os quatro nós e apontam para uma compilação específica de uma versão específica de um objeto.

Atribuição: para os três primeiros nós, é possível atribuir qualquer valor inteiro positivo, incluindo 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

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

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

Pipelines de imagens de contêiner

- Contêiner Docker em execução em uma instância EC2 (temporário)
- Systems Manager Inventory Association (por meio do Systems Manager State Manager, se `EnhancedImageMetadata` estiver habilitado) na instância EC2
- Imagem do contêiner do docker
- Dockerfile

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

Distribuição

O EC2 Image Builder pode distribuir AMIs ou imagens de contêiner para AWS qualquer 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 iniciar instâncias do EC2 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 as AMIs em todas as regiões nas quais a AMI é distribuída.

Você também pode usar sua AWS Organizations conta para impor limitações às contas dos membros para iniciar instâncias somente com AMIs aprovadas e compatíveis. 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 nova versão de uma fórmula de imagem (console), ou Crie uma nova versão de receita de contêiner com o console.

Compartilhar recursos

Para compartilhar componentes, receitas ou imagens com outras contas ou dentro delas AWS Organizations, consulteCompartilhar recursos do EC2 Image Builder.

Conformidade

Para o 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, os CVEs sem patches, a conectividade pública com 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 Manual do usuário do Amazon Inspector. Para fortalecimento, o EC2 Image Builder é validado com STIG. Para obter uma lista completa dos componentes do STIG disponíveis por meio do Image Builder, consulte Componentes de fortalecimento do STIG gerenciados pela Amazon para o EC2 Image Builder. Para obter mais informações, consulte os benchmarks do Center for Internet Security (CIS).

Conceitos básicos do EC2 Image Builder

Este capítulo ajuda você a configurar seu ambiente e criar um pipeline de imagem automatizado ou um pipeline de contêiner pela primeira vez, usando o assistente de console Create image pipeline do EC2 Image Builder.

Conteúdos

Pré-requisitos

Verifique os seguintes pré-requisitos para criar um pipeline de imagem com o EC2 Image Builder. Salvo indicação específica em contrário, são necessários pré-requisitos para todos os tipos de pipeline.

Perfil vinculado ao serviço do EC2 Image Builder

O EC2 Image Builder usa uma função vinculada ao serviço para conceder permissões a AWS outros serviços em seu nome. Não é necessário criar manualmente uma função vinculada a 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 vinculados ao serviço para o EC2 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 EC2 Image Builder e endpoints de interface da VPC ()AWS PrivateLink.
- O Image Builder é compatível com o EC2-Classic.
- 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 Linux do Amazon EC2, o Image Builder instala o Systems Manager Agent na instância de compilação, se ela ainda não estiver presente, e a remove antes de criar a imagem.

  Windows

  O Image Builder não instala o Systems Manager Agent em instâncias de compilação do Amazon EC2 para 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 Systems Manager Agent em sua instância do Amazon EC2 para Windows Server, consulte Instalar manualmente o Systems Manager Agent em instâncias do EC2 para Windows Server no Guia do usuário do AWS Systems Manager

Repositório de contêineres (pipelines de imagens de contêineres)

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 Amazon ECR, siga as etapas descritas em Como criar um repositório no Guiado usuário do Amazon Elastic Container Registry.

AWS Identity and Access Management (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 SMS ManagedInstanceCore

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

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "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 políticas EC2InstanceProfileForImageBuilderECRContainerBuildse do AmazonSSM ManagedInstanceCore.

> **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 do Ec2ImageBuilderCrossAccountDistributionAccess à função. Para ter mais informações, consulte Compartilhar recursos do EC2 Image Builder.

Acessar o EC2 Image Builder

Você pode gerenciar o EC2 Image Builder a partir de uma das seguintes interfaces.

- Página inicial do console do EC2 Image Builder. Do console do EC2 Image Builder.
- AWS Command Line Interface (AWS CLI). Você pode usar o AWS CLI para acessar as operações AWS da API. Para obter mais informações, consulte Instalando a interface de linha de AWS comando no Guia AWS Command Line Interface do usuário.
- AWS Ferramentas para SDKs. Você pode usar SDKs e ferramentas do AWS para acessar e gerenciar o Image Builder usando seu idioma preferido.

Criar um pipeline de imagem usando o assistente do console do EC2 Image Builder

Esse tutorial explica como criar um pipeline automatizado para criar e manter uma imagem personalizada do EC2 Image Builder usando o assistente de console Criar pipeline de imagem. 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 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 imagens.
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 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 controle de versionamento semântico dos recursos do Image Builder, consulte Versionamento semântico.

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 (OS) e Origem da imagem. Isso resulta em uma lista de AMIs 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.
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 deve escolher pelo menos um componente de compilação.

   No painel Compilar 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.

      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 Gerencie os parâmetros AWSTOE do componente com o EC2 Image Builder.

   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 fortalecimento 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 Componentes selecionados para ver se há pelo menos dois resultados.
4. Os componentes recém-adicionados podem não ter seu versionamento ou configurações de parâmetros de entrada expandidas. 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.
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.

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

Etapa 3: Definir a configuração da infraestrutura - opcional

O Image Builder inicia instâncias do EC2 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 para as instâncias de compilação e teste do EC2 que são 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 Gerencie as configurações de distribuição do EC2 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 imagens no painel de navegação. Se sua compilação for bem-sucedida, seu novo pipeline será exibido na lista Pipelines de imagens.

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 do nome do pipeline para selecionar o pipeline do 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 do nome da fórmula para selecionar a fórmula do 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 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. Para confirmar a exclusão, digite `Delete` na caixa e selecione 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. Para confirmar a exclusão, digite `Delete` na caixa e selecione Excluir.

Criar um pipeline de imagens de contêiner usando o assistente do console do EC2 Image Builder

Este tutorial explica como criar um pipeline automatizado para criar e manter uma imagem personalizada do EC2 Image Builder Docker usando o assistente de console Criar pipeline de imagem. 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 imagens.
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 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 controle de versionamento semântico dos recursos do Image Builder, consulte Versionamento semântico.

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.

5. Na seção Componentes, você deve escolher pelo menos um componente de compilação.

   No painel Compilar 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.

   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 Gerencie os parâmetros AWSTOE do componente com o EC2 Image Builder.

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 fortalecimento 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 seguintes variáveis contextuais em seu Dockerfile.

ParentImage (obrigatório)

No momento da construção, essa variável é resolvida na imagem base da sua receita.

Exemplo:

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

ambientes (necessários se os componentes forem especificados)

Essa variável será resolvida em um script que executa componentes.

Exemplo:

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

componentes (opcional)

O Image Builder resolve scripts de componentes de criação e teste para os componentes que a receita do contêiner inclui. Essa variável pode ser colocada em qualquer lugar no Dockerfile, depois da variável de ambientes.

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 instâncias do EC2 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 imagens no painel de navegação. Se sua compilação for bem-sucedida, seu novo pipeline será exibido na lista Pipelines de imagens.

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 do nome do pipeline para selecionar o pipeline do 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 Deletar.

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 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. Para confirmar a exclusão, digite `Delete` na caixa e selecione 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. Para confirmar a exclusão, digite `Delete` na caixa e selecione Excluir.

AWS Task Orchestrator and Executor gerenciador de componentes

O EC2 Image Builder usa AWS Task Orchestrator and Executor o aplicativo AWSTOE() para orquestrar fluxos de trabalho complexos, modificar configurações do sistema e testar seus sistemas sem escrever código. Este aplicativo gerencia e executa componentes que implementam seu esquema de documento declarativo.

Por ser um aplicativo independente, ele não requer configuração adicional do servidor. Pode ser executado em qualquer infraestrutura de nuvem e no local.

Conteúdo

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 AWS postagem do blog de Segurança.

Arquitetura	Plataforma	Link para fazer download	Exemplo
386	AL 2 e 2023 RHEL 7 e 8 Ubuntu 16.04, 18.04, 20.04 e 22.04 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	Windows Server 2012 R2, 2016, 2019 e 2022	`https://awstoe- <region>.s3.com/latest/windows/amd64/awstoe.exe`	https://awstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/windows/amd64/awstoe.exe
AMD64	AL 2 e 2023 RHEL 7 e 8 Ubuntu 16.04, 18.04, 20.04 e 22.04 CentOS 7 e 8 CentOS Stream 8 SUSE 12 e 15	`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
ARM64	AL 2 e 2023 RHEL 7 e 8 Ubuntu 16.04, 18.04, 20.04 e 22.04 CentOS 7 e 8 CentOS Stream 8 SUSE 12 e 15	`https://awstoe- <region>.s3.s3.com/latest/linux/arm64/awstoe`	https://awstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/linux/arm64/awstoe

Regiões 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
Leste dos EUA (N. 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 (Norte 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
Asia Pacific (Osaka)	ap-northeast-3

Região da AWS nome	Região da AWS
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
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 (Barém)	me-south-1
América do Sul (São Paulo)	sa-east-1

Região da AWS nome	Região da AWS
China (Pequim)	cn-north-1
China (Ningxia)	cn-northwest-1

Comece 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 o AWSTOE aplicativo e usá-lo pela primeira vez.

Verifique a assinatura do download da AWSTOE instalação

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

Tópicos

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

Verifique a assinatura do download da instalação do AWSTOE no Linux

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

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 AWSTOE está alterado ou corrompido, não execute o arquivo de instalação. Em vez disso, entre em contato AWS Support para obter mais informações sobre suas opções de suporte, consulte AWS Support.

AWSTOE arquivos para sistemas operacionais baseados em Linux 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. O Amazon EC2 publica uma chave pública e assinaturas que você pode usar para verificar as ferramentas da CLI do Amazon 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

Se o seu sistema operacional for Linux ou Unix, as ferramentas do GPG 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 no Linux baseado em Debian

- Em um terminal, execute o comando a seguir: apt-get install gnupg.

Para instalar as ferramentas do GPG no Linux baseado em Red Hat

- Em um terminal, execute o comando a seguir: yum 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 AWS Support.

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 AWS Support.

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 AWS Support.

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:

F8 83 11 EE F0 4A A2 91 E3 79 21 BA 6B FC AF F8 19 92 12 D7

> **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, verifique se o valor da impressão digital é:
> 5B 77 F4 F0 C3 7A 8B 89 D9 A7 8F 54 B6 85 11 CE 9E A3 BF 17

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 Properties (Propriedades).
2. Escolha a guia Assinaturas digitais.
3. Em Signature List, escolha Amazon Services LLC e, em seguida, escolha Details.
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:

     F8 83 11 EE F0 4A A2 91 E3 79 21 BA 6B FC AF F8 19 92 12 D7

     então seu AWSTOE binário baixado é autêntico e pode ser instalado com segurança.

     > **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, verifique se o valor da impressão digital é:
     > 5B 77 F4 F0 C3 7A 8B 89 D9 A7 8F 54 B6 85 11 CE 9E A3 BF 17

   - 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 AWS Security Blog Post.

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, consulteVerifique a assinatura no Linux. Para verificar seu download em um servidor Linux, consulteVerifique a assinatura no Windows.

> **Important**
>
> 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 instância do EC2 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 instância do EC2, a execução AWSTOE usa as mesmas permissões da função do IAM anexada à instância do EC2.

Para obter mais informações sobre perfis do IAM para o EC2, consulte perfis do IAM para o 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`.

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

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

Para definir essas variáveis no Windows o prompt de comando, use `set`.

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

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

Etapa 3: Desenvolver documentos de componentes localmente

AWSTOE os componentes são criados com documentos YAML de texto simples. Para obter mais informações sobre sintaxe de documentos, consulte Use documentos de componentes em AWSTOE.

Veja a seguir exemplos de documentos do componente Hello World que você pode usar para desenvolver seus documentos localmente.

`hello-world-windows.yml`.

```
name: Hello World
description: This is Hello World testing document for Windows.
schemaVersion: 1.0
phases:
  - name: build
    steps:
      - name: HelloWorldStep
        action: ExecutePowerShell
        inputs:
          commands:
            - Write-Host 'Hello World from the build phase.'
  - name: validate
    steps:
      - name: HelloWorldStep
        action: ExecutePowerShell
        inputs:
          commands:
            - Write-Host 'Hello World from the validate phase.'
  - name: test
    steps:
      - name: HelloWorldStep
        action: ExecutePowerShell
        inputs:
          commands:
            - Write-Host 'Hello World from the test phase.'
```

`hello-world-linux.yml`.

```
name: Hello World
description: This is hello world testing document for Linux.
schemaVersion: 1.0
phases:
  - name: build
    steps:
      - name: HelloWorldStep
        action: ExecuteBash
        inputs:
          commands:
```

```
          - echo 'Hello World from the build phase.'
  - name: validate
    steps:
      - name: HelloWorldStep
        action: ExecuteBash
        inputs:
          commands:
            - echo 'Hello World from the validate phase.'
  - name: test
    steps:
      - name: HelloWorldStep
        action: ExecuteBash
        inputs:
          commands:
            - echo 'Hello World from the test phase.'
```

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`.

Windows

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

Linux

```
$ awstoe validate --documents /home/user/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 Gerencie a configuração da infraestrutura do EC2 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 <S3Bucket> --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
 <comma separated list of phases>
```

Obtenha ajuda com AWSTOE

```
awstoe --help
```

Use documentos de componentes em AWSTOE

Para criar um componente usando 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 que você cria. Serviços da AWS use seu componente ao criar uma nova Amazon Machine Image (AMI) ou imagem de contêiner.

Tópicos

- Esquemas de exemplo de documento
- Defina e referencie variáveis em 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.

Etapas

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, valores de entrada/saída e exemplos, consulte Módulos de ação suportados pelo gerenciador de componentes do AWSTOE.

Registro em log do componente

AWSTOE cria uma nova pasta de log nas instâncias do EC2 que é usada 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: `/var/lib/amazon/toe/`
   - Windows: `$env:ProgramFiles\Amazon\TaskOrchestratorAndExecutor\`
2. Prefixo do arquivo — Esse é um prefixo padrão usado para todos os componentes: “`TOE_`”.
3. Tempo de execução — Esse 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 }}
```

ou

```
{{ 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	Descrição	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	Descrição	Tipo	Obrigatório
name	Nome da fase.	String	Sim
etapas	Lista das etapas da fase.	Lista	Sim

As definições de esquema para uma etapa são as seguintes.

Campo	Descrição	Tipo	Obrigatório	Valor padrão
name	Nome definido pelo usuário para a etapa.	String		
ação	Palavra-chave referente ao módulo que executa a etapa.	String		

Campo	Descrição	Tipo	Obrigatório	Valor padrão
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)

Campo	Descrição	Tipo	Obrigatório	Valor padrão
onFailure	Especifica o que a etapa deve fazer em caso de falha. Os valores válidos são os seguintes: • 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	String	Não	Anular

Campo	Descrição	Tipo	Obrigatório	Valor padrão
	`IgnoredFailure` depois do número máximo de tentativas 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	

Esquemas de exemplo de documento

Veja a seguir um exemplo de esquema 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}}'
```

Veja a seguir um exemplo de esquema de documento para baixar e executar um arquivo binário Linux personalizado.

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

Veja a seguir um exemplo de esquema de documento para instalar 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 }}'
```

A seguir está um exemplo de esquema de documento para instalar o AWS CLI usando 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 }}'
```

Defina e referencie variáveis em AWSTOE

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.

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 Guia do usuário do AWS Secrets Manager . Para obter mais informações sobre os AWS Systems Manager Parameter Store, consulte Parameter Store do AWS Systems Manager no AWS Systems Manager Guia do usuário.

Sintaxe

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

Opção da CLI:	Obrigatório	Descrição
--parâmetros *nome*=*valor*,...	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
```

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	Nome 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 de parâmetro 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 }}
```

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 `start``end`, 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

Campo	Descrição	Tipo	Obrigatório	Padrão
`end`	Valor final da iteração. Não aceita expressões de encadeamento.	Inteiro	Sim	n/a
`updateBy`	Diferença pela qual um valor iterativo é atualizado por meio da adição. Deve ser um valor negativo ou positivo diferente de zero. Não aceita expressões de encadeamento.	Inteiro	Sim	n/a

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	Lista de strings	Sim	n/a

Campo	Descrição	Tipo	Obrigatório	Padrão
	strings na lista. As expressões encadeadas devem estar entre aspas duplas para serem interpretadas corretamente 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 exclusivo quando comparado com outros nomes de loop na mesma fase.	String	Não	""
`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: `"_"` Expressões encadeadas não podem ser usadas.	String	Não	Vírgula: `","`

> **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)

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

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, 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 suportados pelo gerenciador de componentes do AWSTOE

Os serviços de criação de imagens, como o EC2 Image Builder, AWSTOE usam módulos de ação para ajudar a configurar as instâncias do EC2 que são 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.

AWSTOE os componentes são criados com documentos YAML de texto simples. Para obter mais informações sobre sintaxe de documentos, consulte Use documentos de componentes em AWSTOE.

> **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.

Tipos de módulo de ação

- Módulos de execução geral
- Módulos de download e upload de arquivos
- Módulos de operação do sistema de arquivos
- Ações de instalação de software
- Módulos de ação do sistema

Módulos de execução geral

A seção a seguir contém detalhes dos módulos de ação que executam comandos e instruções de execução geral.

Módulos de execução geral

- ExecuteBash
- ExecuteBinary
- ExecuteDocument
- ExecutePowerShell

ExecuteBash

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

Primitivo	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

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 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

Primitivo	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

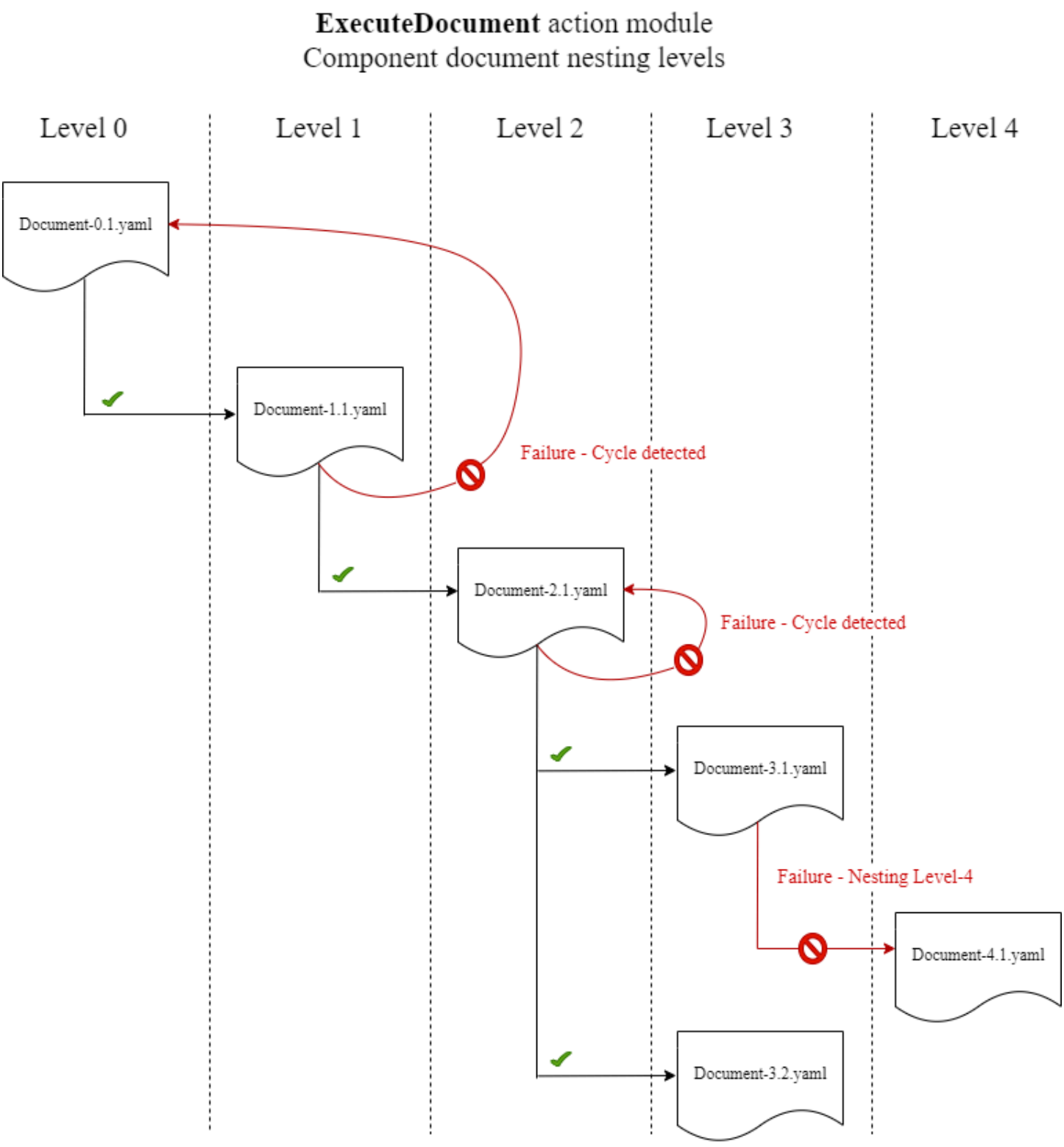
O módulo de ExecuteDocumentação adiciona suporte para documentos de componentes aninhados, executando vários documentos de componentes a partir de um documento. AWSTOE valida o documento que é passado no parâmetro de entrada em tempo de execução.

Restrições

- Esse módulo de ação é executado uma vez, sem permitir novas tentativas e sem a opção de definir limites de timeout. 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

Primitivo	Descrição	Tipo	Obrigatório
`document`	Caminho do documento do componente. Entre as opções válidas estão: • Caminhos de arquivo locais • S3-URI • ARNs da versão de compilação do componente do EC2 Image Builder	String	Sim
`document-s3-bucket-owner`	O ID da conta da do proprietário do bucket do S3 onde são armazenad os os documento s do component e. (Recomendado se você estiver usando URIs do S3 no documento do componente.)	String	Não
`phases`	Fases a serem executadas no documento do componente, expressas como uma lista separada por vírgulas. Se nenhuma fase for especificada,	String	Não

Primitivo	Descrição	Tipo	Obrigatório
	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

Primitivo	Descrição	Tipo	Obrigatório
`name`	O nome do parâmetro de entrada a ser passado para o documento do componente que o módulo de ExecuteDocumentação 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`.

```
"outputs": "[{\"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

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 comandos/instruções especificados no bloco de comandos são convertidos em um arquivo de script (por exemplo,`input.ps1`) e executados usando o Windows. PowerShell O resultado da execução do arquivo shell é o código de saída da etapa.

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

Primitivo	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 iniciar 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 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."
}
```

Módulos de download e upload de arquivos

A seção a seguir contém detalhes dos módulos de ação que executam comandos e instruções de execução geral.

Baixe e faça upload de módulos de ação

- S3Download
- S3Upload
- WebDownload

S3Download

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/*`*.

> **Note**
>
> Todas as pastas no caminho de destino devem existir antes do download, ou o download falhará.

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 suportados

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` no bucket/objeto (por exemplo,). `arn:aws:s3:::BucketName/*`
- Vários arquivos: `s3:ListBucket` no bucket/objeto (por exemplo, `arn:aws:s3:::BucketName`) e `s3:GetObject` no bucket/objeto (por exemplo, `arn:aws:s3:::BucketName/*`).

Entrada

Primitivo	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 asterisco (`/*`), para baixar um conjunto	String	Sim	N/D

Primitivo	Descrição	Tipo	Obrigatório	Padrão
	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
`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

Primitivo	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 "destination" file is set`	Booleano	Não	verdadeiro

Primitivo	Descrição	Tipo	Obrigatório	Padrão
	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://mybucket/path/to/package.zip
      destination: C:\myfolder\package.zip
      expectedBucketOwner: 123456789022
      overwrite: false
    - source: s3://mybucket/path/to/package.zip
      destination: C:\myfolder\package.zip
      expectedBucketOwner: 123456789022
      overwrite: true
    - source: s3://mybucket/path/to/package.zip
      destination: C:\myfolder\package.zip
      expectedBucketOwner: 123456789022
```

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

Saída

Nenhum.

S3Upload

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 suportados

- 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

Primitivo	Descrição	Tipo	Obrigatório	Padrão
`source`	O caminho local onde os arquivos/pastas fonte se originam. 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 para o bucket do Amazon S3 especificado no caminho de destino.	String	Não	N/D

Primitivo	Descrição	Tipo	Obrigatório	Padrão
	Recomendamos que você verifique a propriedade do bucket Amazon S3 especificado no destino.			

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

Saída

Nenhum.

WebDownload

O módulo de WebDownloadação permite que você baixe arquivos e recursos de um local remoto pelo protocolo HTTP/HTTPS (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

Primitivo	Descrição	Tipo	Obrigatório	Padrão
`source`	O URL HTTP/HTTPS válido (é recomendado HTTPS),	String	Sim	N/D

Primitivo	Descrição	Tipo	Obrigatório	Padrão
	que segue o padrão RFC 3986. Expressões de encadeamento são permitidas.			
`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

Primitivo	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`

Primitivo	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, devem ser fornecido s a soma de verificação e o algoritmo. Expressões de encadeamento são permitidas.	String	Não	N/D

Primitivo	Descrição	Tipo	Obrigatório	Padrão
`algorithm`	O algoritmo usado para calcular a soma de verificação. As opções são MD5, SHA1, SHA256 e SHA512. Para que a verificação do arquivo seja ativada, devem ser fornecido s a soma de verificação e o algoritmo. Expressões de encadeamento são permitidas.	String	Não	N/D
`ignoreCertificateErrors`	A validação do certificado SSL é ignorada quando ativada.	Booleano	Não	`false`

Saída

Primitivo	Descrição	Tipo				
`destination`	String delimitada por caracteres de nova linha que especifica	String				

Primitivo	Descrição	Tipo				
	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


Output:
{
  "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/
```

```
Output:
{
  "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


Output:
{
  "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


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

```
}
```

Módulos de operação do sistema de arquivos

A seção a seguir contém detalhes dos módulos de ação que executam comandos e instruções de execução geral.

Módulos de operação do sistema de arquivos

- AppendFile
- CopyFile
- CopyFolder
- CreateFile
- CreateFolder
- CreateSymlink
- DeleteFile
- DeleteFolder
- ListFiles
- MoveFile
- MoveFolder
- ReadFile
- SetFileEncoding
- SetFileOwner
- SetFolderOwner
- SetFilePermissions
- SetFolderPermissions

AppendFile

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

Primitivo	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-16`, `utf16-LE`, `utf-16-LE` `utf16-BE`, `utf-16-BE` , `utf32`, `utf-32`, `utf32-LE`,`utf-32-LE` ,	Sim

Primitivo	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
					`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 (Linux)

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

Exemplo de entrada: anexar arquivo sem 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 sem codificação (Linux)

```
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 sem codificação (Linux)

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

Exemplo de entrada: anexar arquivo sem codificação (Linux)

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

Saída

Nenhum.

CopyFile

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 já houver um arquivo 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

Primitivo	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

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

Primitivo	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

Primitivo	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 um arquivo (Linux)

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

Exemplo de entrada: copiar um arquivo (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

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 já houver um arquivo 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

Primitivo	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 do 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

Primitivo	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.

Primitivo	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: copiar 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: copiar 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 mais informações, consulte Substitua o script de limpeza do Linux.

Saída

Nenhum.

CreateFolder

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

Primitivo	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.

Primitivo	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

O módulo de CreateSymlinkação cria links simbólicos ou arquivos que contêm uma referência a outro arquivo. Este módulo não é suportado em 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

Primitivo	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.

Primitivo	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

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

Primitivo	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

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

Primitivo	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

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

Primitivo	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos	Compatível com todas as plataformas
`path`	O caminho da pasta.	String	Sim	N/D	N/D	Sim

Primitivo	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

Primitivo	Descrição	Tipo
`files`	A lista de arquivos.	String

Exemplo de saída

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

MoveFile

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 já houver um arquivo 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

Primitivo	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

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 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 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

Primitivo	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

Primitivo	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

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

Primitivo	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

Primitivo	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

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

Primitivo	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

Primitivo	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

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 suportado em 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

Primitivo	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

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 suportado em 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

Primitivo	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.

Primitivo	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

O módulo de SetFilePermissionsação modifica o `permissions` de um arquivo existente. Este módulo não é suportado em 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

Primitivo	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

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 suportado em 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

Primitivo	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.

Primitivo	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

Esta seção descreve os módulos de ação que executam comandos e instruções de ação de instalação de software.

Requisitos do IAM

Se o caminho de download da instalação for um URI do S3, o perfil do IAM que você associa 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.

Ações de instalação de software

- InstallMSI
- Desinstalar MSI

InstallMSI

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

Primitivo	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

Primitivo	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
	Um URI de objeto S3 válido. • Um URL HTTP/HTTPS da web válido (é recomendado HTTPS) que siga o padrão RFC 3986. Expressões de encadeamento são permitidas.				

Primitivo	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`

Primitivo	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				

Primitivo	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
	de saída indicando que uma reinicial ização é necessária.				

Primitivo	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 logo do MSI, consulte	String	Não	`*VX`	`i,w,e,a,r,u,c,m,o,p,v,x,+,!,*`

Primitivo	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 arquivo de log não existir, ele 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

Primitivo	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="ReallySuppress"` • `REINSTALLMODE="ecmus"` • `REINSTALL="ALL"`	Map[String]String	Não	N/D	N/D

Primitivo	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	Booleano	Não	`false`	`true`, `false`

Primitivo	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
	validação não são ignorados. O instalador é executado somente quando a validação é bem-sucedida. Esse é o comportamento padrão.				

Primitivo	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	Booleano	Não	`false`	`true`, `false`

Primitivo	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
	que 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 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": ""
}
```

Desinstalar MSI

O módulo de ação `UninstallMSI` 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. 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

Primitivo	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

Primitivo	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
	Um URL HTTP/ HTTPS da web válido (é recomendado HTTPS) que siga o padrão RFC 3986. Expressões de encadeamento são permitidas.				

Primitivo	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`

Primitivo	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				

Primitivo	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
	de saída indicando que uma reinicial ização é necessária.				

Primitivo	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 logo do MSI, consulte	String	Não	`*VX`	`i,w,e,a,r,u,c,m,o,p,v,x,+,!,*`

Primitivo	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 arquivo de log não existir, ele 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

Primitivo	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="ReallySuppress"` • `REINSTALLMODE="ecmus"` • `REINSTALL="ALL"`	Map[String]String	Não	N/D	N/D

Primitivo	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	Booleano	Não	`false`	`true`, `false`

Primitivo	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
	validação não são ignorados. O instalador é executado somente quando a validação é bem-sucedida. Esse é o comportamento padrão.				

Primitivo	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	Booleano	Não	`false`	`true`, `false`

Primitivo	Descrição	Tipo	Obrigatório	Valor padrão	Valores aceitos
	que 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 comandos e instruções de sistema de arquivo.

Módulos de ação do sistema

- Reinicializar

- SetRegistry
- UpdateOS

Reinicializar

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

Primitivo	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

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

Primitivo	Descrição	Tipo	Obrigatório
`path`	Caminho da chave de registro.	String	Sim

Primitivo	Descrição	Tipo	Obrigatório
`name`	Nome da chave de registro.	String	Sim
`value`	Valor da chave de registro.	Sequência/número/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

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

Primitivo	Descrição	Tipo	Obrigatório
`include`	Para Windows, você pode especificar o seguinte: • Uma ou mais IDs de artigo da Base de Dados de Conhecimento Microsoft (KB) devem ser incluídas na lista de atualizações 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

Primitivo	Descrição	Tipo	Obrigatório
`exclude`	Para Windows, você pode especificar o seguinte: • Uma ou mais IDs de artigo da Base de Dados de Conhecimento Microsoft (KB) devem ser incluídas na lista de atualizações para 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.	Lista de strings	Não

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:////config.json`* <bucket-path><bucket-name>).

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.

Um 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- 2:123456789012/ /2021.12.02/1). 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 Defina e referencie variáveis em AWSTOE.

- 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>"
}
```

Componentes gerenciados do pacote do Distributor 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 receita de contêiner para fórmulas de contêiner. Para obter mais informações sobre o pacote `AwsVssComponents`, consulte Criar um snapshot consistente com o aplicativo VSS no Guia do usuário do Amazon EC2 para instâncias do Windows.

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 do Distributor `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 AWS Management Console 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, escolha 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.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AllowDistributorSendCommand",
   "Effect": "Allow",
   "Action": [
    "ssm:SendCommand"
   ],
   "Resource": [
    "arn:${AWS::Partition}:ssm:${AWS::Region}::document/AWS-ConfigureAWSPackage",
    "arn:${AWS::Partition}:ec2:${AWS::Region}:${AWS::AccountId}:instance/*"
   ]
  },
  {
   "Sid": "AllowGetCommandInvocation",
   "Effect": "Allow",
   "Action": [
    "ssm:GetCommandInvocation"
   ],
   "Resource": [
    "*"
   ]
  }
 ]
}
```

4. 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. Escolha Create policy (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 Choose the service that will use this role (Escolher o serviço que usará essa função), selecione EC2 e Next: Permissions (Próximo: permissões).
4. Em Select your use case (Selecione seu caso de uso), escolha EC2 e Next: Permissions (Próximo: permissões).
5. Na lista de políticas, marque a caixa de seleção ao lado do AmazonSSM ManagedInstanceCore. (Digite `SSM` na caixa de texto se você precisar restringir a lista.)
6. Nessa lista de políticas, escolha a caixa ao lado de EC2 InstanceProfileForImageBuilder. (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 ter 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 do 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 do Distribuidor

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

Para ver os pacotes disponíveis no AWS Management Console, 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 da 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 fortalecimento do CIS não seguem as regras padrão de ordenação de componentes em receitas do Image Builder. 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.

Componentes de fortalecimento do STIG gerenciados pela Amazon para o EC2 Image Builder

Guias de implementação técnica de segurança (STIGs) são os padrões de fortalecimento de configuração criados pela Agência de Sistemas de Informação de Defesa (DISA) para proteger os sistemas e softwares de informação. 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 STIGs relacionados que o Image Builder suporta para aquele pacote, o componente de fortalecimento os aplica.

Esta página lista todos os STIGs que o Image Builder suporta e que são aplicados às instâncias do EC2 que o Image Builder inicializa quando você compila 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 Gerencie componentes com o Image Builder.

Quando você cria uma imagem, os componentes de fortalecimento do STIG registram em log se os STIGs compatíveis foram aplicados ou 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 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-Build-Windows-Low versão 2022.4.x

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 dos STIGs do Windows, consulte a STIGs Document Library. Para obter informações sobre como visualizar a lista completa, consulte STIG Viewing Tools.

- Windows Server 2022 STIG versão 1 release 1

  V-254335, V-254336, V-254337, V-254338, V-254351, V-254357, V-254363 e V-254481
- Windows Server 2019 STIG Versão 2 Release 5

  V-205691, V-205819, V-205858, V-205859, V-205860, V-205870, V-205871 e V-205923
- Windows Server 2016 STIG Versão 2 Release 5

  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-225537, V-225536, V-225526, V-225525, V-225514, V-225511, V-225490, V-225489, V-225488, V-225487, V-225485, V-225484, V-225483, V-225482, V-225481, V-225480, V-225479, V-225476, V-225473, V-225468, V-225462, V-225460, V-225459, V-225412, V-225394, V-225392, V-225376, V-225363, V-225362, V-225360, V-225359, V-225358, V-225357, V-225355, V-225343, V-225342, V-225336, V-225335, V-225334, V-225333, V-225332, V-225331, V-225330, V-225328, V-225327, V-225324, V-225319, V-225318 e V-225250

- Microsoft .NET Framework 4.0 STIG Versão 2 Release 2

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

- Windows Firewall STIG Versão 2 Release 1

  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 Release 3

  V-46477, V-46629 e V-97527

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

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

STIG-Build-Windows-Medium versão 2022.4.x

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 dos STIGs do Windows, consulte a STIGs Document Library. Para obter informações sobre como visualizar a lista completa, consulte STIG Viewing Tools.

> **Note**
>
> Os componentes de proteção STIG-Build-Windows-Medium incluem todas as configurações STIG listadas que AWSTOE se aplicam aos componentes STIG-Build-Windows-Low

Hardening, além das configurações STIG listadas especificamente para vulnerabilidades da Categoria II.

- Windows Server 2022 STIG versão 1 Release 1

  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-254358, V-254359, V-254360, V-254361, V-254362, V-254364, V-254365, 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-254431, V-254432, V-254433, V-254434, V-254435, V-254436, V-254438, V-254439, V-254442, V-254443, V-254444, V-254445, V-254449, V-254450, V-254451, V-254452, V-254453, V-254454, V-254455, V-254456, V-254459, V-254460, V-254461, V-254462, V-254463, V-254464, V-254468, V-254470, V-254471, V-254472, V-254473, V-254476, V-254477, V-254478, V-254479, V-254480, V-254482, V-254483, V-254484, V-254485, V-254486, V-254487, V-254488, V-254489, V-254490, V-254493, V-254494, V-254495, V-254497, V-254499, V-254501, V-254502, V-254503, V-254504, V-254505, V-254507, V-254508, V-254509, V-254510, V-254511 e V-254512
- Windows Server 2019 STIG Versão 2 Release 5

  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-205652, V-205655, V-205656, V-205659, V-205660, V-205662, V-205671, V-205672, V-205673, V-205675, V-205676, V-205678, V-205679, V-205680, V-205681, V-205682, V-205683, V-205684, 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-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-205825, V-205826, V-205827, V-205828, V-205830, V-205832, V-205833, V-205834, V-205835, V-205836, V-205837, V-205838, V-205839, V-205840, V-205841, V-205861, V-205863, V-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 e V-236001

- Windows Server 2016 STIG Versão 2 Release 5

  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-225016, V-225017, V-225018, V-225019, V-225021, V-225022, V-225023, V-225024, V-225028, V-225029, V-225030, V-225031, V-225032, V-225033, V-225034, V-225035, V-225038, V-225039, V-225040, V-225041, V-225042, V-225043, V-225047, V-225049, V-225050, V-225051, V-225052, V-225055, V-225056, V-225057, V-225058, 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 e V-236000

- 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-225574, V-225573, V-225572, V-225571, V-225570, V-225569, V-225568, V-225567, V-225566, V-225565, V-225564, V-225563, V-225562, V-225561, V-225560, V-225559, V-225558, V-225557, V-225555, V-225554, V-225553, V-225551, V-225550, V-225549, V-225548, V-225546, V-225545, V-225544, V-225543, V-225542, V-225541, V-225540, V-225539, V-225538, V-225535, V-225534, V-225533, V-225532, V-225531, V-225530, V-225529, V-225528, V-225527, V-225524, V-225523, V-225522, V-225521, V-225520, V-225519, V-225518, V-225517, V-225516, V-225515, V-225513, V-225510, V-225509, V-225508, V-225506, V-225504, V-225503, V-225502, V-225501, V-225500, V-225494, V-225486, V-225478, V-225477, V-225475, V-225474, V-225472, V-225471, V-225470, V-225469, V-225464, V-225463, V-225461, V-225458, V-225457, V-225456, V-225455, V-225454, V-225453, V-225452, V-225448, V-225443, V-225442, V-225441, V-225415, V-225414, V-225413, V-225411, V-225410, V-225409, V-225408, V-225407, V-225406, V-225405, V-225404, V-225402, V-225401, V-225400, V-225398, V-225397, V-225395, V-225393, V-225391, V-225389, V-225386, V-225385, V-225384, V-225383, V-225382, V-225381, V-225380, V-225379, V-225378, V-225377, V-225375, V-225374, V-225373, V-225372, V-225371, V-225370, V-225369, V-225368, V-225367, V-225356, V-225353, V-225352, V-225351, V-225350, V-225349, V-225348, V-225347, V-225346, V-225345, V-225344, V-225341, V-225340, V-225339, V-225338, V-225337, V-225329, V-225326, V-225325, V-225317, V-225316, V-225315, V-225314, V-225305, V-225304, V-225303, V-225302, V-225301, V-225300, V-225299, V-225298, V-225297, V-225296, V-225295, V-225294, V-225293, V-225292, V-225291, V-225290, V-225289, V-225288, V-225287, V-225286, V-225285, V-225284, V-225283, V-225282, V-225281, V-225280, V-225279, V-225278, V-225277, V-225276, V-225275, V-225273, V-225272, V-225271, V-225270, V-225269, V-225268, V-225267, V-225266, V-225265, V-225264, V-225263, V-225261, V-225260, V-225259 e V-225239

- Microsoft .NET Framework 4.0 STIG Versão 2 Release 2

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

- Windows Firewall STIG Versão 2 Release 1

  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-241998 e V-242003

- Internet Explorer 11 STIG Versão 2 Release 3

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

V-46473, V-46475, V-46481, V-46483, V-46501, V-46507, V-46509, V-46511, V-46513, V-46515, V-46517, V-46521, V-46523, V-46525, V-46543, V-46545, V-46547, V-46549, V-46553, V-46555, V-46573, V-46575, V-46577, V-46579, V-46581, V-46583, V-46587, V-46589, V-46591, V-46593, V-46597, V-46599, V-46601, V-46603, V-46605, V-46607, V-46609, V-46615, V-46617, V-46619, V-46621, V-46625, V-46633, V-46635, V-46637, V-46639, V-46641, V-46643, V-46645, V-46647, V-46649, V-46653, V-46663, V-46665, V-46669, V-46681, V-46685, V-46689, V-46691, V-46693, V-46695, V-46701, V-46705, V-46709, V-46711, V-46713, V-46715, V-46717, V-46719, V-46721, V-46723, V-46725, V-46727, V-46729, V-46731, V-46733, V-46779, V-46781, V-46787, V-46789, V-46791, V-46797, V-46799, V-46801, V-46807, V-46811, V-46815, V-46819, V-46829, V-46841, V-46847, V-46849, V-46853, V-46857, V-46859, V-46861, V-46865, V-46869, V-46879, V-46883, V-46885, V-46889, V-46893, V-46895, V-46897, V-46903, V-46907, V-46921, V-46927, V-46939, V-46975, V-46981, V-46987, V-46995, V-46997, V-46999, V-47003, V-47005, V-47009, V-64711, V-64713, V-64715, V-64717, V-64719, V-64721, V-64723, V-64725, V-64729, V-72757, V-72759, V-72761, V-72763, V-75169 e V-75171

- Microsoft Edge STIG Versão 1 Release 6 (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

- Defender STIG Versão 2 Release 4 (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-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-Build-Windows-High versão 2022.4.x

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 dos STIGs do Windows, consulte a STIGs Document Library. Para obter informações sobre como visualizar a lista completa, consulte STIG Viewing Tools.

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

- Windows Server 2022 STIG versão 1 Release 1

  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 2 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-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 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-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-225556, V-225552, V-225547, V-225507, V-225505, V-225498, V-225497, V-225496, V-225493, V-225492, V-225491, V-225449, V-225444, V-225399, V-225396, V-225390, V-225366, V-225365, V-225364, V-225354 e V-225274

- Microsoft .NET Framework 4.0 STIG Versão 2 Release 2

  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.

- Windows Firewall STIG Versão 2 Release 1

  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 Release 3

  Inclui todas as configurações do STIG compatíveis que o componente de fortalecimento aplica às vulnerabilidades das Categorias II e III (Média e Baixa) do Internet Explorer 11. Nenhuma configuração adicional do STIG se aplica a vulnerabilidades da Categoria I.

- Microsoft Edge STIG Versão 1 Release 6 (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

- Defender STIG Versão 2 Release 4 (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-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 primeiro trimestre de 2024 - 02/06/2024 (sem alterações):

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

Alterações no quarto trimestre de 2023 - 12/04/2023 (sem alterações):

Não houve alterações no componente STIGS do Windows para a versão 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. 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 2 (AL2)

Configurações do RHEL 8 STIG

- RHEL 8
- CentOS 8
- Amazon Linux 2023 (AL 2023)

STIG-Build-Linux-low versão 2024.1.x

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

RHEL 7 STIG Versão 3 Versão 14

- RHEL 7/CentOS 7

  V-204452, V-204576 e V-204605

- AL2

  V-204452, V-204576 e V-204605

RHEL 8 STIG Versão 1 Versão 13

- RHEL 8/CentOS 8/AL 2023

  V-230241, V-244527, V-230269, V-230270, V-230285, V-230253, V-230346, V-230381, V-230395, V-230468, V-230469, V-230491, V-230485, V-230486, V-230494, V-230495, V-230496, V-230497, V-230497, V-230498, V-230498, V-230496, V-230497, V-230498, V-230498, V-230496 230499 e V-230281

Ubuntu 18.04 STIG versão 2 versão 13

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

Ubuntu 20.04 STIG Versão 1 Versão 11

V-238202, V-238234, V-238235, V-238237, V-238323, V-238373, V-238221, V-238222, V-238223, V-238224, V-238226, V-238362, V-238357 e V-238308

STIG-Build-Linux versão média 2024.1.x

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

> **Note**
>
> Os componentes de fortalecimento STIG-Build-Linux-medium incluem todas as configurações STIG listadas que AWSTOE se aplicam aos componentes de fortalecimento

STIG-Build-Linux-low, além das configurações STIG listadas especificamente para vulnerabilidades da Categoria II.

RHEL 7 STIG Versão 3 Versão 14

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

  V-204585, V-204490, V-204491, V-255928, V-204405, V-204406, V-204407, V-204408, V-204409, V-204410, V-204411, V-204412, V-204413, V-204414, V-204415, V-204422, V-204423, V-204427, V-204416, V-204418, V-204426, V-204431, V-204457, V-204466, V-204417, V-204434, V-204435, V-204587, V-204588, V-204589, V-204590, V-204591, V-204592, V-204593, V-204596, V-204597, V-204597, V-204597, V-204597 4598, V-204599, V-204600, V-204601, V-204602, V-204622, V-233307, V-255925, V-204578, V-204595, V-204437, V-204503, V-204507, V-204508, V-204510, V-204511, V-204512, V-204514, V-204514, 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-204547, V-204548, V-204549, V-204550, V-204551, V-204552, V-204553, V-204554, V-204555, V-204556, V-204557, V-204558, V-204559, V-204560, V-204562, V-204563, V-204564, V-204565, V-204566, V-204567, V-204568, V-204572, V-204584, V-204609, V-204610, V-204611, V-204612, V-204613, V-204614, V-204615, V-204616, V-204617, V-204625, V-204630, V-255927, V-237634, V-237635, V-251703, V-204449, V-204450, V-204451, V-204619, V-204519, V-20451, V-204619, V-204519, V-20451 79, V-204631, V-204633 e V-256970
- AL2:

  V-204585, V-204490, V-204491, V-255928, V-204405, V-204406, V-204407, V-204408, V-204409, V-204410, V-204411, V-204412, V-204413, V-204414, V-204415, V-204422, V-204423, V-204427, V-204416, V-204418, V-204426, V-204431, V-204457, V-204466, V-204417, V-204434, V-204435, V-204587, V-204588, V-204589, V-204590, V-204591, V-204592, V-204593, V-204596, V-204597, V-204597, V-204597, V-204597 4598, V-204599, V-204600, V-204601, V-204602, V-204622, V-233307, V-255925, V-204578, V-204595, V-204437, V-204503, V-204507, V-204508, V-204510, V-204511, V-204512, V-204514, V-204514, 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-204547, V-204548, V-204549, V-204550, V-204551, V-204552, V-204553, V-204554, V-204555, V-204556, V-204557, V-204558, V-204559, V-204560,

V-204562, V-204563, V-204564, V-204565, V-204566, V-204567, V-204568, V-204572, V-204584, V-204609, V-204610, V-204611, V-204612, V-204613, V-204614, V-204615, V-204616, V-204617, V-204625, V-204630, V-255927, V-237634, V-237635, V-251703, V-204449, V-204450, V-204451, V-204619, V-204519, V-20451, V-204619, V-204519, V-20451 79, V-204631, V-204633 e V-256970

RHEL 8 STIG Versão 1 Versão 13

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/AL 2023

  V-230257, V-230258, V-230259, V-230550, V-230248, V-230249, V-230250, V-230245, V-230246, V-230247, V-230397, V-230399, V-230400, V-230401, V-230228, V-230298, V-230387, V-230231, V-230231, V-23033 24, V-230365, V-230370, V-230378, V-230383, V-230236, V-230314, V-230315, V-244523, V-230266, V-230267, V-230268, V-230280, V-230310, V-230311, V-230312, V-230502, 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-244550, V-244551, V-244552, V-244553, V-244554, V-250317, V-251718, V-230237, V-230313, V-230356, V-230357, V-230358, V-230359, V-230360, V-230361, V-230362, V-230363, V-230368, V-230369, V-230375, V-230376, V-230377, V-244524, V-244533 251713, V-251717, V-251714, V-251715, V-251716, V-230332, V-230334, V-230336, V-230338, V-230340, V-230342, V-230344, V-230333, V-230335, V-230337, V-230339, V-230341, V-230343, V-230343, V-230345, V-230343, V-230345 230240, V-230282, V-250315, V-250316, V-230255, V-230277, V-230278, V-230348, V-230353, V-230386, V-230390, V-230392, V-230394, V-230396, V-230393, V-230398, V-230402, V-230403, V-230403, V-230404, V-230405, 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-230464, V-230465, V-230466, V-230467, V-230471, V-230472, V-230473, V-230474, V-230480, V-230483, V-244542, V-230503, V-230244, V-230286, V-230287, V-230288, V-230290, V-230291, V-230296, V-230330, V-230382, V-230526, V-230527, V-230555, V-230556, V-244526, V-244528, V-237642, V-237643, V-251711, V-230238, V-230239, V-230273, V-230275, V-230478, V-230488, V-230489, V-230489 230559, V-230560, V-230561, V-237640 e V-256974

Ubuntu 18.04 STIG versão 2 versão 13

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-219188, V-219190, V-219191, 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-219303, V-222303, V-222V-219326, V-219328, V-219330, V-219342, V-219189, V-219192, V-219193, V-219194, V-219315, V-219195, V-219196, V-219197, V-219213, V-219214, V-219215, V-219216, V-219216, V-219216, V-2216, V-2216, V-2216, V-2216, V-2216, V-2216, V-2216, V-2216, V-2216, V-2216, V-2216, V-2216, V-2216, V-2216, V-2216, V-2216 V-219217, V-219218, V-219220, V-219221, V-219222, V-219223, V-219224, V-219227, V-219228, V-219229, V-219230, V-219231, V-219232, V-219233, V-219234, V-219234, V-219234, V-219234 235, V-219236, V-219238, V-219239, V-219240, V-219241, V-219242, V-219243, V-219244, V-219250, V-219254, V-219257, V-219263, V-219264, V-219265, V-219266, V-219267, V-219268, V-219269, V-219270, V-219271, V-219272, V-219273, V-219274, V-219275, V-219275 219276, V-219277, V-219279, V-219281, V-219287, V-219291, V-219297, V-219299, V-219300, V-219309, V-219310, V-219311, V-219312, V-233779, V-233780, V-255906, V-219336, V-219338, V-219344, V-219181, V-219184, V-219186, V-219155, V-219156, V-219160, V-219306, V-219149, V-219166, V-219176, V-219339, V-219331, V-219331 37 e V-219335

Ubuntu 20.04 STIG Versão 1 Versão 11

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-238205, V-238207, V-238329, V-238337, V-238339, V-238340, V-238344, V-238345, V-238346, V-238347, V-238348, V-238349, V-238350, V-238351, V-238352, V-238376, V-238377, V-238378, V-238209, V-238325, V-238330, V-238333, V-238369, V-238338, V-238341, V-238342, V-238343, V-238324, V-238353, V-238228, V-238225, V-238227, V-238299, V-238238, V-238238, V-238239, V-238240, V-238241, V-238242, V-238244, V-238245, V-238246, V-238247, V-238248, V-238249, V-238250, V-238251, V-238252, V-238253, V-238254, V-238255, V-238255, V-238255 8256, 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-238300, V-238301, V-238302, V-238304, V-238309, V-238310, V-238315, V-238316, V-238317, V-238318, V-238319, V-238320, V-251505, V-238360, V-238211, V-238212, V-238213, V-238216, V-238220, V-255912, V-238355, V-238236, V-238303, V-238358, V-238356, V-238359, V-238370 e V-238334

STIG-Build-Linux-High versão 2024.1.x

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

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

RHEL 7 STIG Versão 3 Versão 14

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

  V-204425, V-204594, V-204455, V-204424, V-204442, V-204443, V-204447, V-204448, V-204502, V-204620 e V-204621
- AL2:

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

RHEL 8 STIG Versão 1 Versão 13

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/AL 2023

V-230265, V-230529, V-230531, V-230264, V-230487, V-230492, V-230533 e V-230558

Ubuntu 18.04 STIG versão 2 versão 13

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 e V-251507

Ubuntu 20.04 STIG Versão 1 Versão 11

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

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 primeiro trimestre de 2024 - 02/06/2024:

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

STIG-Build-Linux-low versão 2024.1.x

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

STIG-Build-Linux versão média 2024.1.x

- RHEL 7 STIG Versão 3 Versão 14
- RHEL 8 STIG Versão 1 Versão 13

- Ubuntu 18.04 STIG versão 2 versão 13
- Ubuntu 20.04 STIG Versão 1 Versão 11

STIG-Build-Linux-High versão 2024.1.x

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

Alterações no quarto trimestre de 2023 - 12/07/2023:

Versões atualizadas do STIG e aplicou o STIGS para a versão do quarto trimestre de 2023 da seguinte forma:

STIG-Build-Linux-low versão 2023.4.x

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

STIG-Build-Linux-Medium versão 2023.4.x

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

STIG-Build-Linux-High versão 2023.4.x

- RHEL 7 STIG Versão 3 Versão 13
- RHEL 8 STIG Versão 1 Versão 12
- Ubuntu 18.04 STIG versão 2 versão 12

- Ubuntu 20.04 STIG Versão 1 Versão 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. Script AL2 anterior combinado 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-linux versão 2021.04.0

O `scap-compliance-checker-linux` componente é executado nas instâncias de criação e teste do pipeline do Image Builder. 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, 2016 e 2019.

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

SCC Versão: 5.4.2

Benchmarks do quarto trimestre de 2021:

- U_MS_ DotNet _Framework_4-0_V2R1_STIG_SCAP_1-2_Benchmark
- U_MS_IE11_V2R1_STIG_SCAP_1-2_Benchmark
- U_MS_Windows_2012_and_2012_R2_MS_V3R2_STIG_SCAP_1-2_Benchmark
- U_MS_Windows_Defender_AV_V2R2_STIG_SCAP_1-2_Benchmark
- U_MS_Windows_Server_2016_V2R1_STIG_SCAP_1-2_Benchmark
- U_MS_Windows_Server_2019_V2R1_STIG_SCAP_1-2_Benchmark
- U_MS_Windows_Firewall_V2R1_STIG_SCAP_1-2_Benchmark
- U_CAN_Ubuntu_18-04_V2R4_STIG_SCAP_1-2_Benchmark
- U_RHEL_7_V3R5_STIG_SCAP_1-2_Benchmark
- U_RHEL_8_V1R3_STIG_SCAP_1-2_Benchmark

scap-compliance-checker-linux versão 2021.04.0

O `scap-compliance-checker-linux` componente é executado nas instâncias de criação e teste do pipeline do Image Builder. 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 oferece suporte à validação de conformidade SCAP para RHEL 7/8 e Ubuntu 18. Atualmente, o aplicativo SCC é compatível com a arquitetura para validação x86.

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

SCC Versão: 5.4.2

Benchmarks do quarto trimestre de 2021:

- U_CAN_Ubuntu_18-04_V2R4_STIG_SCAP_1-2_Benchmark
- U_RHEL_7_V3R5_STIG_SCAP_1-2_Benchmark
- U_RHEL_8_V1R3_STIG_SCAP_1-2_Benchmark
- U_MS_ DotNet _Framework_4-0_V2R1_STIG_SCAP_1-2_Benchmark
- U_MS_IE11_V2R1_STIG_SCAP_1-2_Benchmark
- U_MS_Windows_2012_and_2012_R2_MS_V3R2_STIG_SCAP_1-2_Benchmark
- U_MS_Windows_Defender_AV_V2R2_STIG_SCAP_1-2_Benchmark
- U_MS_Windows_Server_2016_V2R1_STIG_SCAP_1-2_Benchmark
- U_MS_Windows_Server_2019_V2R1_STIG_SCAP_1-2_Benchmark
- U_MS_Windows_Firewall_V2R1_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
Componentes SCAP adicionados	Os componentes SCAP a seguir foram introduzidos: • scap-compliance-checker-linux Versão criada 2021.04.0 (versão SCC: 5.4.2) • scap-compliance-checker-linux Versão criada 2021.04.0 (versão SCC: 5.4.2)	20 de dezembro de 2021

AWSTOE referência de comando

AWSTOE é um aplicativo de gerenciamento de componentes executado no AWS CLI.

> **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 sudo su comando uma vez ao fazer login antes de executar os comandos vinculados abaixo. Para obter mais informações sobre módulos de AWSTOE ação, consulteMódulos de ação suportados pelo 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 `--config` parâmetro ou na lista de documentos do componente especificada pelo `--documents` parâmetro.

> **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.

--documents ***./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).*
- URIs do S3 ()`s3://`*bucket/key*.
- *ARNs da versão de compilação do componente Image Builder (arn:aws:imagebuilder:us-west- 2:123456789012/ /2021.12.02/1). 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 nome1 = valor1, nome2 = valor2...

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

Quando você executa esse comando, ele valida a sintaxe do documento YAML para cada um dos documentos componentes 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.

--documents *./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*)
- URIs do S3 (s3://*bucket/key*)
- *ARNs da versão de compilação do componente Image Builder (arn:aws:imagebuilder:us-west- 2:123456789012/ /2021.12.02/1) 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.

Gerenciar recursos do EC2 Image Builder

Os recursos são os blocos de construção que compõem os 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

Gerencie componentes com o 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 compilação e teste do Amazon EC2. 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 criação (pré-snapshot) — Durante o estágio de criação, você faz alterações na instância de compilação do Amazon EC2 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 componentes são executadas durante a fase de construção:

   - 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-snapshot): durante o estágio de teste, há algumas diferenças entre imagens que criam AMIs e imagens de contêiner. Para fluxos de trabalho da AMI, o Image Builder inicia uma instância do EC2 com base no snapshot que ele criou como etapa final do estágio de compilação. 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 de componentes é executada para cada componente incluído na fórmula durante a fase de teste:

   - 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 AWSTOE 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 construção, 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, o documento do componente deve definir a `test` fase e não outras fases.
> 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, consulteGerencie os parâmetros AWSTOE do componente com o EC2 Image Builder.

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 documentos de componentes em AWSTOE.

> **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 do AWS CLI.

Conteúdo

Crie um documento do componente do YAML

Para criar um componente, forneça um documento do componente do aplicativo YAML. Isso representa as fases e etapas necessárias para criar o componente.

Os exemplos nesta seção criam um componente de construção que chama o módulo de `Update0S` ação no aplicativo de gerenciamento de AWSTOE componentes. O módulo atualiza o sistema operacional. Para obter mais informações sobre o módulo de ação `Update0S`, consulte UpdateOS. Para obter mais informações sobre as fases, etapas e sintaxe dos documentos do componente do aplicativo AWSTOE YAML, consulte Usar documentos em. AWSTOE

> **Note**
>
> O Image Builder determina os tipos de componentes no fluxo de trabalho do pipeline. Esse fluxo de trabalho corresponde ao estágio de Compilação e ao estágio de Teste no processo de compilação. O Image Builder determina o tipo de componente da seguinte forma:
>
> - 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. Se esse componente de construção tiver uma fase `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 Gerencie componentes com o 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

Crie um arquivo de componente YAML

Use uma ferramenta de edição de arquivos para criar um arquivo chamado *`update-linux-os.yaml`*. Inclua 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

Crie um arquivo de componente YAML

Use uma ferramenta de edição de arquivos para criar um arquivo chamado *update-windows-os.yaml*. Inclua 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.

Gerencie os parâmetros AWSTOE do componente com o EC2 Image Builder

Você pode gerenciar AWSTOE componentes, incluindo criar e definir parâmetros de componentes, diretamente do console do EC2 Image Builder ou AWS CLI usando comandos ou um dos SDKs do Image Builder. 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.

> **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 Guia do usuário do AWS Secrets Manager . Para obter mais informações sobre o AWS Systems Manager Repositório de parâmetros, consulte AWS Systems Manager Repositório de parâmetros no AWS Systems Manager Guia do usuário.

Use parâmetros no documento do componente do YAML

Para criar um componente, forneça um documento do componente do aplicativo YAML. Isso representa as fases e etapas necessárias para criar o componente. 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 chamado *hello-world-test.yaml*. Inclua 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 chamado *hello-world-test.yaml*. Inclua 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.

Para obter mais informações sobre as fases, etapas e sintaxe dos documentos do componente do aplicativo AWSTOE YAML, consulte Usar documentos em. 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 usar o console do Image Builder para criar um componente diretamente a partir do seu documento do YAML, consulte Crie um componente usando o console do Image Builder.
- Para usar os comandos do Image Builder no AWS CLI para criar seu componente, consulteCrie AWSTOE componentes com o Image Builder com o 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 (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 nesta receita 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.

Listar e visualizar detalhes do componente

Esta seção descreve como você pode encontrar informações e visualizar detalhes dos componentes AWS Task Orchestrator and Executor (AWSTOE) que você usa em suas receitas do EC2 Image Builder.

Detalhes do componente

- Listar AWSTOE componentes
- Listar versões de compilação do componente (AWS CLI)
- Obter detalhes do componente (AWS CLI)
- Obter detalhes da política de componentes (AWS CLI)

Listar AWSTOE componentes

Você pode usar um dos métodos a seguir para listar e filtrar AWSTOE componentes.

AWS Management Console

Para exibir uma lista de componentes no AWS Management Console, 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:
   - Início rápido (gerenciado pela Amazon) — componentes publicamente disponíveis que a Amazon cria e mantém.
   - De minha propriedade — Componentes que você criou. Esta é a ação 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 list-components comando para retornar uma lista dos AWSTOE componentes 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

- Self
- Amazon
- ThirdParty
- Compartilhada

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 versões de compilação do componente (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.

```
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"
            }
        }
    ]
}
```

Obter detalhes do componente (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": {}
    }
}
```

Obter detalhes da política de componentes (AWS CLI)

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

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

Crie um componente usando o console do Image Builder

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. Crie um documento do componente do YAML
5. Depois de inserir os detalhes do componente, selecione Criar componente.

> **Note**
>
> Para ver seu novo componente ao criar ou atualizar uma receita, 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.

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.

Crie um componente com o AWS CLI

Esta seção descreve como usar os comandos do Image Builder para criar componentes AWS Task Orchestrator and Executor (AWSTOE) a partir do AWS Command Line Interface. Para criar um componente, forneça um documento do componente do aplicativo YAML. Isso representa as fases e etapas necessárias para criar o componente. Para criar um novo documento do componente do YAML, consulte Crie um documento do componente do YAML.

Crie AWSTOE componentes com o Image Builder com o 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://my-s3-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 nas etapas anteriores. 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 ação da API Image Builder. Para revisar os parâmetros de solicitação de comando da API, consulte o CreateComponentcomando na Referência da API do 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://my-s3-bucket/my-path/update-linux-os.yaml",
 "kmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/98765432-
b123-456b-7f89-0123456f789c",
 "tags": {
  "MyTagKey-purpose": "security-updates"
 }
}
```

   Note

   - 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 e o Linux usa 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-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 e o Linux usa 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://my-s3-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 nas etapas anteriores. 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 ação da API Image Builder. Para revisar os parâmetros de solicitação de comando da API, consulte o CreateComponentcomando na Referência da API do 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://my-s3-bucket/my-path/update-windows-os.yaml",
 "kmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/98765432-
b123-456b-7f89-0123456f789c",
 "tags": {
  "MyTagKey-purpose": "security-updates"
 }
}
```

   Note

   - 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 e o Linux usa 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 e o Linux usa a barra (/).

AWSTOE controle de versão de componentes para atualizações ()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.

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.

Importar um componente (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*my-s3-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://my-s3-bucket/AdminConfig.ps1",
"kmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/60763706-
b131-418b-8f85-3420912f020c"
}
```

Para importar o componente, execute o seguinte comando.

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

Limpeza de 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 Exclua recursos do EC2 Image Builder.

Gerenciar fórmulas

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. O número de componentes que você pode aplicar a uma receita é limitado a 20 componentes no total. Isso inclui componentes de compilação e teste.

Depois de criar uma receita, não é possível modificá-la ou substituí-la. Para atualizar os componentes depois de criar uma receita, você deve criar uma nova receita ou versão da receita. Você sempre pode aplicar tags às receitas 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 com o aplicativo AWS Task Orchestrator and Executor (AWSTOE). Para começar, consulte o Comece com AWSTOE.

Esta seção aborda como listar, visualizar e criar receitas.

Conteúdo

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 fórmulas de imagem do EC2 Image Builder.

Detalhes da fórmula de imagem

- Listar receitas de imagens (console)
- Listar fórmulas de imagem (AWS CLI)
- Exibir detalhes da fórmula da imagem (console)
- Obtenha detalhes da fórmula da imagem (AWS CLI)
- Obtenha detalhes da política de fórmula de imagem (AWS CLI)

Listar receitas de imagens (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 Recipe name, depois escolher Exibir detalhes.

Listar fórmulas de imagem (AWS CLI)

O exemplo a seguir mostra como listar todas as fórmulas de imagens, usando a AWS CLI.

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

Exibir detalhes da fórmula da imagem (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 receitas de imagens (console).

Na página de detalhes da fórmula, você pode:

- Excluir a fórmula. Para obter mais informações sobre exclusão de recursos no Image Builder, consulte Exclua recursos do EC2 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 Criar um pipeline de imagem usando o assistente do console do EC2 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 da imagem (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 fórmula de imagem (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 de suas fórmulas de contêiner 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 Exclua recursos do EC2 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 Criar um pipeline de imagem usando o assistente do console do EC2 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 nova versão de uma fórmula de imagem (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. Esse detalhe de base 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.
- 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.

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

- ID da AMI – Obrigatório. No entanto, esta configuração não é pré-preenchida com sua entrada original. Você deve inserir o ID da AMI para sua imagem de base.

- 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.

> **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 do EC2 Image Builder.

- Diretório de trabalho – Pré-selecionado, mas você pode editá-lo.
- Componentes — Os componentes que já estão incluídos na receita 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 fortalecimento do CIS não seguem as regras padrão de ordenação de componentes nas fórmulas do Image Builder. 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.

  > **Note**
  >
  > As listas de componentes de compilação e teste exibem os componentes disponíveis com base no tipo de proprietário do componente. Para adicionar ou atualizar componentes para sua fórmula, selecione o tipo de proprietário do componente que você está procurando. Por exemplo, se você quiser adicionar um componente associado a uma imagem base na qual você se inscreveu AWS Marketplace, selecione na lista `Third party managed` de tipos de proprietário, ao lado da barra de pesquisa.

  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 nesta receita 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 receita se houver algum parâmetro obrigatório ausente e não tiver 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 Guia do usuário do AWS Secrets Manager . 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 para 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 de imagem.

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 fórmula de imagem a partir de AWS CLI, você deve criar os componentes que a fórmula 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 com o AWS CLI deste guia.

Depois de criar seus componentes, ou se você estiver usando componentes existentes, anote os ARNs que você deseja incluir na fórmula.

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 ação da API Image Builder. Para revisar os parâmetros de solicitação de comando da API, consulte o CreateImageRecipecomando na Referência da API do 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. O valor pode ser o ARN de imagem de base ou um ID de AMI.

  > **Note**
  >
  > O exemplo do Linux usa uma AMI do Image Builder, e o exemplo do Windows usa um ARN.

- semanticVersion (string, obrigatório) – A versão semântica da fórmula de imagem, expressa no seguinte formato, com valores numéricos em cada posição para indicar uma versão específica: <major>.<minor>.<patch>. Por exemplo, um valor pode ser `1.0.0`. Para saber mais sobre o versionamento semântico dos recursos do Image Builder, consulte Versionamento semântico.
- components (matriz, obrigatório) – Contém uma variedade de objetos `ComponentConfiguration`. 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 um dos exemplos para criar sua própria fórmula de imagem, você deve substituir os ARNs de exemplo pelos ARNs dos componentes que você está usando para sua fórmula.

- 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 receita se houver algum parâmetro obrigatório ausente e não tiver 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 Guia do usuário do AWS Secrets Manager . 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-0a01b234c5de6fabc",
```

```
"semanticVersion": "1.0.0",
"components": [
 {
  "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012: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 no SKU 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-2:123456789012:component/my-example-component/2019.12.02/1"
 },
 {
  "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/my-imported-component/1.0.0/1"
 }
]
}
```

> **Note**
>
> Para saber mais sobre o versionamento semântico dos recursos do Image Builder, consulte Versionamento semântico.

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 e o Linux usa a barra (/).

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 Criar um pipeline de imagem (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 como importar 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.

IAM role (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 configurações de licença criadas com AWS License Manager à sua imagem base, selecione na lista Nome da configuração de 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 Add tag. Para remover uma tag, selecione Remover tag.

Criar uma nova versão de receita de contêiner

Esta seção mostra como criar uma nova versão de receita de contêiner.

Conteúdo

Crie uma nova versão de receita de contêiner com o console

Criar uma nova versão de uma receita do contêiner é praticamente o mesmo que criar uma nova receita. A diferença é que certos detalhes são pré-selecionados para corresponder à receita 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 receita

- 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 base

- Selecione a opção de imagem — Pré-selecionada, mas editável. Se você alterar sua escolha para a fonte da imagem 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 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 auto-versionamento — Não corresponde à sua receita base. As opções de auto-versionamento usam 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.

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

- ID AMI — Pré-preenchido, mas editável.
- 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 receita 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 fortalecimento do CIS não seguem as regras padrão de ordenação de componentes nas fórmulas do Image Builder. 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.

  > **Note**
  >
  > As listas de componentes de compilação e teste exibem os componentes disponíveis com base no tipo de proprietário do componente. Para adicionar ou atualizar componentes para sua fórmula, selecione o tipo de proprietário do componente que você está procurando. Por exemplo, se você quiser adicionar um componente associado a uma imagem base na qual você se inscreveu AWS Marketplace, selecione na lista `Third party managed` de tipos de proprietário, ao lado da barra de pesquisa.

  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 nesta receita 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 receita se houver algum parâmetro obrigatório ausente e não tiver 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 Guia do usuário do AWS Secrets Manager . 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 de Dockerfile

- Modelo Dockerfile — Pré-preenchido, mas editável. Você pode especificar qualquer uma das seguintes variáveis contextuais que o Image Builder substitui por informações de compilação em tempo de execução.

  #### ParentImage (obrigatório)

  No momento da construção, essa variável é resolvida na imagem base da sua receita.

Exemplo:

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

ambientes (necessários se os componentes forem especificados)

Essa variável será resolvida em um script que executa componentes.

Exemplo:

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

componentes (opcional)

O Image Builder resolve scripts de componentes de criação e teste para os componentes que a receita do contêiner inclui. Essa variável pode ser colocada em qualquer lugar no Dockerfile, depois da variável de ambientes.

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 receita de contêiner

1. Na parte superior da página de detalhes da receita do contêiner, escolha Criar nova versão. Você será direcionado para a página Criar receita para fórmulas de contêiner.
2. Para criar a nova versão, faça suas alterações e depois escolha Criar receita.

Para obter mais informações sobre como criar uma receita de contêiner ao criar um pipeline de imagens, consulte Etapa 2: Escolher fórmula a seção Introdução deste guia.

Criar uma receita 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 receita de contêiner na etapa a seguir se refere a exemplos de componentes que são criados na seção Crie um componente com o AWS CLI deste guia.

Depois de criar seus componentes, ou se você estiver usando componentes existentes, anote os ARNs que você deseja incluir na fórmula.

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 receita 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 ação da API Image Builder. Para revisar os parâmetros de solicitação de comando da API, consulte o CreateContainerRecipecomando na Referência da API do 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 comandos.

   Aqui está um resumo dos parâmetros neste exemplo:

   - componentes (matriz de objetos, obrigatório) — Contém uma matriz de `ComponentConfiguration` objetos. Pelo menos um componente de 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 os ARNs de exemplo pelos ARNs dos componentes que você está usando para sua receita. Isso inclui o Região da AWS, nome, e o número da versão de cada um.

- parâmetros (matriz de objetos) — Contém uma matriz de `ComponentParameter` objetos. 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 receita se houver algum parâmetro obrigatório ausente e não tiver 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 Guia do usuário do AWS Secrets Manager . 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 receita do contêiner.
- description (string) — A descrição da receita do contêiner.
- parentImage (string, obrigatório) — A imagem que a receita de imagem usa como base para sua imagem personalizada. O valor pode ser o ARN de imagem de base ou um ID de AMI.
- platformOverride (string) - Especifica a plataforma do sistema operacional quando você usa uma imagem base personalizada.
- SemanticVersion (string, obrigatório) — A versão semântica da receita 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.
- tags (string map) - Tags que estão anexadas à receita 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) - O ID da AMI a ser usado como imagem base para uma instância de compilação e teste de contêineres. Se você não especificar esse valor, o Image Builder usará a AMI otimizada do Amazon ECS apropriada 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-east-1:123456789012:component/
helloworldal2/x.x.x"
   }
 ],
 "containerType": "DOCKER",
 "description": "My Linux Docker container image",
 "dockerfileTemplateData": "FROM
 {{{ imagebuilder:parentImage }}}\n{{{ imagebuilder:environments }}}\n{{{ imagebuilder:comp
 "name": "amazonlinux-container-recipe",
 "parentImage": "amazonlinux:latest",
 "platformOverride": "Linux",
 "semanticVersion": "1.0.2",
 "tags": {
   "sometag" : "Tag detail"
 },
 "instanceConfiguration": {
   "image": "ami-1234567890",
   "blockDeviceMappings": [
   {
   "deviceName": "/dev/xvda",
   "ebs": {
    "deleteOnTermination": true,
    "encrypted": false,
    "volumeSize": 8,
    "volumeType": "gp2"
    }
    }
   ]
```

```
 },
 "targetRepository": {
   "repositoryName": "myrepo",
   "service": "ECR"
 },
 "workingDirectory": "/tmp"
}
```

2. Criar a receita

   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 e o Linux usa a barra (/).

Limpeza de 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 Exclua recursos do EC2 Image Builder.

Gerenciar imagens do EC2 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 EC2 Image Builder.

Conteúdo

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

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 a ListImagesReferência 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

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 ou Linux).
- É possível filtrar imagens com base no tipo de saída que elas produzem (AMI ou imagem de contêiner).
- É possível usar a fonte de filtro para encontrar imagens que foram importadas de uma máquina virtual com o VMIE.

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 da versão do recurso de imagem, por exemplo, “Windows” ou “Linux”.

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).

Creation time

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 ou Linux).
- É possível filtrar imagens com base no tipo de saída que elas produzem (AMI ou imagem de contêiner).

- É possível usar a fonte de filtro para encontrar imagens que foram importadas de uma máquina virtual com o VMIE.

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).

Version (Versão)

A plataforma do sistema operacional da versão do recurso de imagem, por exemplo, “Windows” ou “Linux”.

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 da versão do recurso de imagem, por exemplo, “Windows” ou “Linux”.

Creation time

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 ou Linux).
- É possível filtrar imagens com base no tipo de saída que elas produzem (AMI ou imagem de contêiner).
- É possível usar a fonte de filtro para encontrar imagens que foram importadas de uma máquina virtual com o VMIE.

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).

Version (Versão)

A plataforma do sistema operacional da versão do recurso de imagem, por exemplo, “Windows” ou “Linux”.

Plataforma

A plataforma do sistema operacional da versão do recurso de imagem, por exemplo, “Windows” ou “Linux”.

Creation time

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"
```

```
  }
 ]
}
```

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 a ListWaitingWorkflowStepsReferência 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, SDKs ou AWS CLI para listar versões de criação de imagens.

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 a ListImageBuildVersionsReferência 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.

Ver os detalhes da 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 o Image Builder API, SDKs ou AWS CLI para obter os detalhes da imagem.

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

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 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`.

Date created (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`).

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 Imagens de máquina da Amazon (AMIs) no console Amazon EC2.

> **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 ter mais informações, consulte Gerencie as configurações de distribuição do EC2 Image Builder.

Guia de configuração de infraestrutura

A guia Configuração de infraestrutura exibe as configurações de infraestrutura do Amazon EC2 que o Image Builder usou para compilar e testar a imagem atualmente exibida. 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 chaves do Amazon EC2 para solução de problemas
- Um tópico do Amazon SNS para notificações de evento

Para ter mais informações, consulte Gerencie a configuração da infraestrutura do EC2 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 (OUs). AWS Organizations

> **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 o Amazon EC2 executa a partir da sua imagem.

- Contas de destino para uma configuração de inicialização mais rápida –
- Configurações de licença associadas – Os ARNs de configuração de licença do License Manager a serem associados à AMI na região especificada.
- Configuração do modelo de inicialização –
- Define a versão padrão do modelo de inicialização –

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 ter mais informações, consulte Gerencie as configurações de distribuição do EC2 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 compilação e teste. Os contêineres têm um fluxo de trabalho adicional para 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 compilação 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 visualizar um fluxo de trabalho diferente, selecione na lista, da seguinte forma:

Fluxos de trabalho da imagem (saída de AMI)

- `build-image`
- `test-image`

Fluxos de trabalho do contêiner (saída de contêiner)

- `build-container`
- `test-container`
- `distribute-container`

> **Note**
>
> Se o fluxo de trabalho ainda não tiver sido iniciado, ele não aparecerá na lista. Por exemplo, se a compilação da sua imagem acabou de ser iniciada, `build-image` é o único tipo de fluxo de trabalho que aparece na lista. Quando o próximo fluxo de trabalho iniciar, `test-image` nesse caso, 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
- Executando
- Concluído

- Com 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 mais informações, consulte Configure escaneamentos de segurança para imagens do Image Builder no AWS Management Console.

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
- Alta
- 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 ter mais informações, consulte Gerencie descobertas de segurança para imagens do Image Builder no AWS Management Console.

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.

Obter detalhes da política de imagens (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

Esta seção mostra como criar imagens do Image Builder e cancelar uma compilação em andamento.

Conteúdo

Criar uma imagem

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 AWS Management Console ou AWS CLI. Você também pode usar a ação CreateImageda API. Para a solicitação de SDK associada, você pode consultar o link Consulte também desse comando na Referência da API EC2 Image Builder.

AWS Management Console

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 Criar um pipeline de imagem (AMI) ou Criar um pipeline de imagens (Docker), 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 do

- Fórmula — Você deve especificar exatamente uma fórmula para sua imagem, da seguinte forma:

Receita 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 à sua 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 AWS CLI create-image. 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 de imagem (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 imagens de VM

O Image Builder se integra à API Amazon EC2 VM Import/Export 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.

Importar uma VM (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. Escolha Importar imagem.
4. Forneça detalhes para as seções a seguir na página Import image. Em seguida, selecione Import image ao concluir.

Geral

1. Especifique um Nome exclusivo para a imagem de base.
2. Especifique uma Versão para a imagem de base. Use o formato a seguir: *major*.*minor*.*patch*.
3. Você também pode inserir uma Descrição opcional para a imagem de base.

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

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 como importar 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.

IAM role (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.

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 Add tag. Para remover uma tag, selecione Remover tag.

Importar uma VM (AWS CLI)

Para importar uma VM de discos para uma AMI e criar um recurso de imagem do Image Builder que você possa referenciar imediatamente, siga estas etapas no AWS CLI:

1. Inicie uma importação de VM, com o comando Amazon EC2 VM import-image Import/Export no AWS CLI. Anote o ID da tarefa retornado pelo comando. Você precisará dele para a próxima etapa. Para obter mais informações, consulte Como importar uma VM como uma imagem usando o VM Import/Export no Guia do usuário de 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 ação da API Image

Builder. Para revisar os parâmetros de solicitação de comando da API, consulte o ImportVmImagecomando na Referência da API do 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, required) — 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.
- description (string) — A descrição da fórmula de imagem.
- platform (string, required) — A plataforma do sistema operacional para a VM importada.
- vmImportTaskId (string, obrigatório) — O `ImportTaskId` (AWS CLI) do processo de importação da VM do Amazon EC2. 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.
- clientToken (string, obrigatório) – Um identificador exclusivo e que diferencia maiúsculas e minúsculas que você fornece para garantir a idempotência da solicitação. Para obter mais informações, consulte Garantir idempotência) na Referência da API do Amazon EC2.
- 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",
    "clientToken": "asz1231231234cs3z",
    "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 e o Linux usa a barra (/).

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
- Alta
- 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 AWS Management Console

Se você ativou o Amazon Inspector para sua conta, o Amazon Inspector digitaliza automaticamente as instâncias EC2 que o Image Builder executa para compilar 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: Ative as verificações 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 imagens.
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 AWS Management Console

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 ter mais informações, consulte Ver os detalhes da 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
- Alta
- 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.

Limpeza de 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 Exclua recursos do EC2 Image Builder.

Gerencie a configuração da infraestrutura do EC2 Image Builder

Você pode usar configurações de infraestrutura para especificar a infraestrutura do Amazon EC2 que o Image Builder usa para criar e testar sua imagem do EC2 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.
- Um perfil de instância que fornece às suas instâncias de compilação e teste com 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 de instância também exige um conjunto mínimo de permissões para que o EC2 Image Builder se comunique com sucesso com a instância. Para ter mais informações, consulte Pré-requisitos.
- 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 chaves do Amazon EC2 que permite que você faça login em sua instância para solucionar problemas caso a criação falhe e você defina `terminateInstanceOnFailure` como `false`.
- 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

> **Tip**
>
> Quando você tem muitos recursos do mesmo tipo; é possível identificar um recurso específico com base nas tags que você lhe atribuiu. 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

- Liste as configurações de infraestrutura (AWS CLI)
- Obter detalhes da configuração de infraestrutura (AWS CLI)

Liste as configurações de infraestrutura (AWS CLI)

O exemplo a seguir mostra como listar todas as suas configurações de infraestrutura, usando o comando list-infrastructure-configurations no AWS CLI.

```
aws imagebuilder list-infrastructure-configurations
```

Obter detalhes da configuração de infraestrutura (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 de AWS infraestrutura, 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.
   - 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 EC2 Image Builder e endpoints de interface da VPC ()AWS PrivateLink.
   - 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 instância do EC2 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 instância do EC2 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 os 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 instâncias do EC2 que o Image Builder usa para criar e testar sua imagem:
  - Selecione a versão de metadados para determinar se o EC2 exige um cabeçalho de token assinado para solicitações de recuperação de metadados de instância.
    - V1 e V2 (token opcional) — Valor padrão se você não selecionar nada.
    - V2 (token obrigatório)

    > **Note**
    >
    > Recomendamos que você configure todas as instâncias do EC2 que o Image Builder executa a partir de um pipeline criado para usar o IMDSv2, de 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.

6. Na seção Tags de infraestrutura (opcional), você pode atribuir metatags à instância do Amazon EC2 que o Image Builder executa durante o processo de criação. As tags são inseridas como pares de valores-chave.
7. Na seção Tags (opcional), você pode atribuir metatags ao recurso de configuração de infraestrutura que o Image Builder cria como saída. As tags são inseridas como pares de valores-chave.

AWS CLI

O exemplo a seguir mostra como configurar a infraestrutura da sua imagem com o create-infrastructure-configuration comando Image Builder no AWS CLI.

1. Criar um arquivo JSON de entrada da CLI

   Este exemplo de configuração de infraestrutura 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 de instância também exige um conjunto mínimo de permissões para que o EC2 Image Builder se comunique com sucesso com a instância. Para ter mais informações, consulte Pré-requisitos.

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 `create-infrastructure-configuration.json`:

```
{
    "name": "MyExampleInfrastructure",
    "description": "An example that will retain instances of failed builds",
    "instanceTypes": [
        "m5.large", "m5.xlarge"
    ],
    "instanceProfileName": "myIAMInstanceProfileName",
    "securityGroupIds": [
        "sg-12345678"
    ],
    "subnetId": "sub-12345678",
    "logging": {
        "s3Logs": {
            "s3BucketName": "my-logging-bucket",
            "s3KeyPrefix": "my-path"
        }
    },
    "keyPair": "myKeyPairName",
    "terminateInstanceOnFailure": false,
    "snsTopicArn": "arn:aws:sns:us-west-2:123456789012:MyTopic"
}
```

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.

Console

Você pode editar os seguintes detalhes de configuração da infraestrutura no console do Image Builder:

- A descrição da configuração da infraestrutura.
- 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, a 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 no 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
```

EC2 Image Builder e endpoints de interface da VPC ()AWS PrivateLink

É possível estabelecer uma conexão privada entre sua VPC e o EC2 Image Builder criando um endpoint da VPC de interface. Os endpoints de interface são alimentados por AWS PrivateLinkuma tecnologia que permite acessar de forma privada as APIs do Image Builder 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 a comunicação com APIs do Image Builder. 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 ter 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 Interface AWS CLI Para 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*.imagebuilder

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 executar ações.
- As ações que podem ser executadas.
- Os recursos sobre os quais as ações podem ser realizadas.

Se você estiver usando componentes gerenciados pela Amazon em sua receita, o endpoint da VCP 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 o EC2 Image Builder, certas solicitações de API com falha, como aquelas que

`RequestLimitExceeded` falham, podem não ser 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 (*`região`*) e pelo ambiente do aplicativo (*`ambiente`*). Image Builder e dê AWSTOE 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 VPC endpoint

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 executar todas as outras ações do EC2 Image Builder.

```
{
```

```
    "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 fornecer acesso aos componentes gerenciados pela Amazon AWSTOE . Substitua a *região principal-org-id*,, e *resource-org-id*pelos valores da sua organização.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRequestsByOrgsIdentitiesToOrgsResources",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
```

```
          "aws:PrincipalOrgID": "principal-org-id",
          "aws:ResourceOrgID": "resource-org-id"
        }
      }
    },
    {
      "Sid": "AllowAccessToEC2ImageBuilderComponents",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "imagebuilder:GetComponent"
      ],
      "Resource": [
        "arn:aws:imagebuilder:region:aws:component/*"
      ]
    }
  ]
}
```

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. Substitua a *região* e o *meio ambiente* pelos valores da sua organização. Adicione outras permissões necessárias à política com base nos requisitos do seu aplicativo.

> **Note**
>
> Para imagens do Linux, se você não especificar dados do usuário em sua receita de imagem, o Image Builder adiciona um script para baixar e instalar o agente do Systems Manager nas instâncias de criação e teste da sua imagem. Para baixar o agente, o Image Builder acessa o bucket do S3 para sua região de compilação.
>
> Para garantir que o Image Builder possa inicializar as instâncias de compilação e teste, adicione o seguinte recurso adicional à sua política de endpoint do S3: `"arn:aws:s3:::amazon-ssm-region/*"`

```
{
```

```
  "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/
*"
      ]
    }
  ]
}
```

Gerencie as configurações de distribuição do EC2 Image Builder

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 OUs a iniciar 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`. Consulte os exemplos para tornar uma AMI pública em ModifyImageAttribute do EC2.

- Crie uma cópia da AMI de saída para cada uma das contas, organizações e OUs de destino especificadas na região de destino. As contas, organizações e OUs de destino possuem suas

cópias da AMI e são cobradas por quaisquer cobranças associadas. Para obter mais informações sobre como distribuir sua AMI 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 ter mais informações, consulte Crie configurações de distribuição para discos VM de saída (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, organizações e OUs especificadas, use as configurações de distribuição com um pipeline do Image Builder que é executado de acordo com uma programação.
- Para criar uma nova imagem e entregá-la às regiões, contas, organizações e OUs especificadas, use as configurações de distribuição com um pipeline do Image Builder que você executa uma vez no console do Image Builder, usando Executar pipeline no menu Ações.
- Para criar uma nova imagem e entregá-la às regiões, contas, organizações e OUs especificadas, use as configurações de distribuição com a seguinte ação da API ou comando do Image Builder na 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.

Este tópico aborda como listar, visualizar e criar configurações de distribuição.

Conteúdo

Listar e visualizar detalhes das configurações de distribuição

Esta seção descreve as várias maneiras pelas quais você pode encontrar informações e visualizar detalhes de suas configurações de distribuição do EC2 Image Builder.

Detalhes das configurações de distribuição

Listar configurações de distribuição (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 (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 (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 Exclua recursos do EC2 Image Builder.
- Editar detalhes de distribuição.

Listar distribuições (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
```

Obter detalhes da configuração de distribuição (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

Crie uma configuração de distribuição da AMI (console)

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.

Criar uma nova configuração de distribuição da AMI:

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 do `EC2ImageBuilderDistributionCrossAccountRole` em todas as contas de destino nas regiões de destino e anexar a política gerenciada do Política do Ec2ImageBuilderCrossAccountDistributionAccess à função.

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 secundárias incluem OUs e AWS contas.
- Organização — conceda acesso à sua AWS Organizations entidade e a todas as suas entidades secundárias. As entidades secundárias incluem OUs e 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 de diferentes tipos.

ii. Você pode expandir a seção Configuração de licença para anexar as configurações de licença criadas com AWS License Manager às 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 de configuração do modelo de inicialização para especificar um modelo de inicialização do EC2 a ser usado para iniciar instâncias da AMI que você criou.

  Se você estiver usando um modelo de execução do EC2, poderá instruir o Image Builder para criar uma nova versão do seu modelo de inicialização 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.

b. Para adicionar configurações de distribuição para outra região, escolha Adicionar região.

7. Escolha Configurações de criação depois de concluir.

Crie configurações de distribuição para AMIs de saída (AWS CLI)

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 configurar o EC2 Fast Launch para AMIs de saída do Windows. Para tornar uma AMI pública, defina as contas autorizadas com permissão de execução como `all`. Consulte os exemplos para tornar uma AMI pública em ModifyImageAttribute do EC2.

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:

   Accounts

   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"
                    ]
                }
            }
        }
    ]
}
```

Organizations and 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"
          ]
        }
      }
    }
  ]
}
```

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 e o Linux usa a barra (/).

Para obter mais informações detalhadas, consulte create-distribution-configuration na Referência de comandos da AWS CLI .

Atualizar as configurações de distribuição da AMI (console)

Você pode alterar as configurações de distribuição da AMI usando o console do Image Builder. As configurações de distribuição atualizadas são usadas para todas as implantações automáticas e manuais do pipeline daqui para frente. 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.

Atualizar configuração de distribuição da AMI

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 Crie uma configuração de distribuição da AMI (console).
5. Escolha Save changes (Salvar alterações) quando terminar.

Criar configurações de distribuição para uma AMI do Windows com o EC2 Fast Launch habilitado (AWS CLI).

O exemplo a seguir mostra como usar o comando create-distribution-configuration para criar configurações de distribuição que tem EC2 Fast Launch configurado para a sua 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 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 with
 cross-account distribution.",
            "amiTags": {
                "KeyName": "Some Value"
            }
        },
        "fastLaunchConfigurations": [{
            "enabled": true,
            "snapshotConfiguration": {
                "targetResourceCount": 5
            },
            "maxParallelLaunches": 6,
            "launchTemplate": {
                "launchTemplateId": "lt-0ab1234c56d789012",
                "launchTemplateVersion": "1"
             },
             "accountId": "123456789012"
        }],
        "launchTemplateConfigurations": [{
                    "launchTemplateId": "lt-0ab1234c56d789012",
                    "setDefaultVersion": true
          }]
    }]
}
```

> **Note**
>
> 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 e o Linux usa a barra (/).

Para obter mais informações detalhadas, consulte create-distribution-configuration na Referência de comandos da AWS CLI .

Crie configurações de distribuição para discos VM de saída (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 ação da API Image Builder. Para revisar os parâmetros de solicitação de comando da API, consulte o comando CreateDistributionConfiguration na Referência EC2 Image Builder API.
   > 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, obrigatório) — O nome da função que concede permissão ao VM Import/Export 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.
  - Stream-optimized ESX Virtual Machine Disk (VMDK) – Compatível com VMware ESX e VMware vSphere versões 4, 5 e 6.
  - 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 e o Linux usa a barra (/).

Para obter mais informações detalhadas, consulte create-distribution-configuration na Referência de comandos da AWS CLI .

Atualizar as configurações de distribuição da AMI (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 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-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 e o Linux usa 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.

Criar e atualizar configurações de distribuição para imagens de contêiner

Esta seção aborda a criação e a atualização de configurações de distribuição imagens de contêiner do Image Builder.

Conteúdo

Criar configurações de distribuição para imagens de contêiner do Image Builder (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 e o Linux usa a barra (/).

Para obter mais informações detalhadas, consulte create-distribution-configuration na Referência de comandos da AWS CLI .

Atualize as configurações de distribuição da sua imagem de contêiner (AWS CLI)

O exemplo a seguir mostra como usar o update-distribution-configuration comando para atualizar as configurações de distribuição da sua 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 e o Linux usa 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 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 AMIs criptografadas, a imagem na conta de origem é descriptografada e enviada para a região de destino, onde é recriptografada usando a chave designada para aquela 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 as AMIs do Image Builder que são distribuídas entre contas, o proprietário da conta de destino deve criar um novo perfil do IAM em sua conta chamada `EC2ImageBuilderDistributionCrossAccountRole`.
2. Ele deve anexar o Política do 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. 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.

```
{
 "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 inicialização do Amazon EC2, você também deve adicionar a seguinte política ao seu `EC2ImageBuilderDistributionCrossAccountRole` em cada conta de destino.

```
{
    "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"
                }
            }
        }
    ]
```

```
}
```

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 ser compatível com as AMIs de virtualização PV. Para obter mais informações, consulte Tipos de virtualização da AMI em Linux.
- 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 a CreateDistributionConfigurationReferência da API EC2 Image Builder.

Configurar a distribuição entre contas para uma AMI do Image Builder (console)

Esta seção descreve como criar e definir configurações de distribuição para distribuição entre contas de suas AMIs do Image Builder usando o AWS Management Console. A configuração da distribuição entre contas requer permissões específicas do IAM. Você deve preencher o Pré-requisitos 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.

Configurar a distribuição entre contas para uma AMI do Image Builder (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 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 os IDs da conta de destino 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 e o Linux usa a barra (/).

Você também pode fornecer o JSON diretamente no comando, usando o parâmetro `--distributions`.

Defina as configurações de distribuição da AMI para usar um modelo de execução do Amazon EC2

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 lançamento do Amazon EC2 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 a execução de uma instância do EC2 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 o Windows Fast Launch em sua imagem, o modelo de execução deve incluir a seguinte tag para que o Image Builder possa habilitar o Windows Fast Launch em seu nome.
> `CreatedBy: EC2 Image Builder`

Adicione um modelo de execução da Amazon EC2 às suas configurações da distribuição AMI (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 o tipo de saída da Amazon Machine Image (AMI). 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, selecione o nome de um modelo de execução do EC2 na lista. Se não houver modelos de lançamento em sua conta, escolha Criar novo modelo de lançamento, que abre os modelos de lançamento no painel do EC2.

   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 execução do Amazon EC2 às configurações de distribuição da AMI ()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 um `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 e o Linux usa a barra (/).

Você também pode fornecer o JSON diretamente no comando, usando o parâmetro `--distributions`.

Gerenciar políticas de ciclo de vida de imagens do EC2 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 AMIs ou imagens de contêiner personalizadas. 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 recursos de imagem de AMI que você exclui, também é possível optar por remover AMIs distribuídas e seus snapshots 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 EC2 Image Builder

Antes que possa 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 IAM em https://console.aws.amazon.com/iam/.
2. Escolha Funções no painel de navegação.
3. Selecione Criar função. 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.

```
{
    "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 Role name.
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 IAM em https://console.aws.amazon.com/iam/.
2. Escolha Funções no painel de navegação.
3. Selecione Criar função. 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.

Políticas de gerenciamento do ciclo de vida para recursos de imagem do EC2 Image Builder

Com as políticas de ciclo de vida de imagem, você pode definir sua estratégia de gerenciamento de recursos para retirar imagens desatualizadas e seus recursos associados por meio de um processo de descontinuação, desabilitação e exclusão de imagens desatualizadas e seus recursos associados. Esta seção mostra como listar políticas, visualizar detalhes da política e criar novas políticas para AMI e imagens de contêiner.

Conteúdo

Listar políticas de gerenciamento do ciclo de vida para recursos de imagem do Image Builder

Você pode obter uma lista de suas políticas de gerenciamento do ciclo de vida de imagens que inclui colunas de detalhes importantes na página de listagem de políticas de ciclo de vida no AWS Management Console, ou com comandos ou ações na API Image Builder, SDKs ou. 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 a ListLifecyclePoliciesReferência da API do EC2 Image Builder. Para ver a solicitação de SDK associada, consulte o link Consulte também na mesma página.

AWS Management Console

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 AWS Management Console, 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, os SDKs ou a AWS CLI do Image Builder para obter os detalhes da política.

Conteúdo

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).

Date created (Data de 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 de ciclo 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: ***`tipo de filtro`***: 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 dependerá 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 AWS Management Console 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.

AWS Management Console

Para criar uma política de ciclo de vida para recursos de imagem da AMI no AWS Management Console, 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 aplicá-las aos recursos selecionados pela política de ciclo de vida. 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 das AMIs associadas 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 as AMIs associadas do Amazon EC2 com uma data de descontinuação de uso. As AMIs permanecerão disponíveis e você ainda poderá iniciar novas instâncias com base nelas.

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`.
- Desabilitar AMIs: marque a caixa de seleção para desabilitar as AMIs associadas do Amazon EC2. Você não poderá mais usar as AMIs nem iniciar novas instâncias com base nelas.

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 poderá cancelar o registro das AMIs associadas ou excluir

os snapshots dessas 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 o registro de AMIs: marque a caixa de seleção para cancelar o registro de AMIs associadas do Amazon EC2. Você não poderá mais usar as AMIs nem iniciar novas instâncias com base nelas.
- Reter imagens, AMIs 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. As tags se aplicam aos recursos de imagem e às AMIs do Amazon EC2. É possível adicionar até 50 pares de chave e valor.

Tags (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 AWS Management Console 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.

AWS Management Console

Para criar uma política de ciclo de vida para recursos de imagem de contêiner no AWS Management Console, 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.

Tags (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 EC2 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 AMIs de saída ou armazenamento de repositórios do 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 das AMIs associadas 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 describe-images comando Amazon EC2 no AWS CLI, ele não retornará AMIs obsoletas no conjunto de resultados. Contudo, você ainda poderá encontrar AMIs obsoletas com suas IDs de AMI.

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 poderá cancelar o registro das AMIs associadas ou excluir os snapshots dessas 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 (API/SDK/CLI)

As regras de exclusão a seguir definem exceções às regras de ciclo de vida das AMIs. As AMIs que atenderem aos critérios especificados pelas regras de exclusão serão excluídas das ações do ciclo de vida. As regras de exclusão não estão disponíveis no AWS Management Console.

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 as AMIs públicas estão excluídas 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.

regiões

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 das AMIs marcadas com elas.

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.

Gerenciar fluxos de trabalho de compilação e teste para imagens do EC2 Image Builder

Um fluxo de trabalho de imagem define a sequência de etapas que o EC2 Image Builder executa durante os estágios de compilação e teste 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. É possível fazer o mesmo para criar fluxos de trabalho somente de compilação.

> **Note**
>
> Você não pode modificar um fluxo de trabalho existente, mas pode cloná-lo ou criar uma nova versão.

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.

O framework do fluxo de trabalho de criação de imagens inclui os dois estágios distintos a seguir.

1. Estágio de criação (pré-snapshot) — Durante o estágio de criação, você faz alterações na instância de compilação do Amazon EC2 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-snapshot): durante o estágio de teste, há algumas diferenças entre imagens que criam AMIs e imagens de contêiner. Para fluxos de trabalho da AMI, o Image Builder inicia uma instância do EC2 com base no snapshot que ele criou como etapa final do estágio de compilação. 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.

O framework do fluxo de trabalho também inclui um estágio de distribuição. No entanto, o Image Builder manipula os fluxos de trabalho desse estágio.

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.

Conteúdo

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, SDKs ou listar fluxos AWS CLI de trabalho de imagens em sua conta.

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 a ListWorkflowsReferência da API do EC2 Image Builder. Para ver a solicitação de SDK associada, consulte o link Consulte também na mesma página.

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: `BUILD` ou `TEST`.
- 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

`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"
  }
 ]
}
```

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 Criptografia e gerenciamento de chaves no EC2 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.

AWS Management Console

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 ter 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 ter 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 do YAML que o Image Builder usa para realizar ações de criação e teste de imagens está estruturado da seguinte forma.

- Identificação
- Parâmetros de entrada
- Etapas
- Outputs

Identificação

Identifica exclusivamente o fluxo de trabalho. Essa seção pode incluir os atributos a seguir.

Campo	Descrição	Tipo	Obrigatório
name		String	Não

Campo	Descrição	Tipo	Obrigatório
	O nome do documento do fluxo de trabalho.		
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

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	Descrição	Tipo	Obrigatório	Restrições
name	O nome do parâmetro.	String	Sim	
descrição	A descrição do parâmetro.	String	Não	

Campo	Descrição	Tipo	Obrigatório	Restrições
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 válido 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

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 seletor JSONPath. Por exemplo: .

```
$.stepOutputs.step-name.output-name
```

Para ter 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	Descrição	Tipo	Obrigatório	Valor padrão	Restrições
ação	A ação do fluxo de trabalho que	String	Sim		Deve ser uma ação de etapa compatíve

Campo	Descrição	Tipo	Obrigatório	Valor padrão	Restrições
	essa etapa executa.				l com documentos de fluxo de trabalho do Image Builder.

Campo	Descrição	Tipo	Obrigatório	Valor padrão	Restrições
`if`, seguido por um conjunto de declaraçõ es condicion ais que modificam o operador `if`.	As declaraçõ es condicion ais adicionam pontos de decisão de fluxo de controle ao corpo das etapas do fluxo de trabalho.	Dict	Não		O Image Builder é compatíve l com as seguintes declaraçõ es condicion ais como modificad ores do operador `if`: • Condições de ramificação e modificad ores: `if`, `and`, `or`, `not`. As condições de ramificaç ão são especific adas individua lmente em uma linha. • Operadore s de comparaçã o:

Campo	Descrição	Tipo	Obrigatório	Valor padrão	Restrições
					`booleanEquals`, `numberEquals`, `numberGreaterThan`, `numberGreaterThanEquals`, `numberLessThan`, `numberLessThanEquals`, `stringEquals`.
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.

Campo	Descrição	Tipo	Obrigatório	Valor padrão	Restrições
inputs	Contém os parâmetros de que a ação de etapa precisa para funcionar. Você pode especificar os valores de chave como valores estáticos ou com uma variável JSONPath que seja resolvida para o tipo de dados correto.	Dict	Sim		
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 alfanuméricos e _. Sem espaços.

Campo	Descrição	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	Descrição	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	Descrição	Tipo	Obrigatório	Valor padrão	Restrições
rollbackEnabled	Configura se a etapa será revertida se ocorrer uma falha. Você pode usar um valor booliano estático ou uma variável dinâmica JSONPath com resolução para um valor booliano.	Booleano	Não	`true`	`true` | `false` | ou uma variável JSONPath com resolução como verdadeiro ou falso.
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.	Entre 1 e 86.400 segundos (máximo de 24 horas)

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"
```

Outputs

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	Descrição	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

Campo	Descrição	Tipo	Obrigatório
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 ter 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

Imagens de máquina da Amazon

ARN

Nome do recurso da Amazon

Ações compatíveis

- BootstrapInstanceForContainer
- CollectImageMetadata
- CollectImageScanFindings
- CreateImage
- ExecuteComponents
- LaunchInstance
- RunCommand
- RunSysPrep
- SanitizeInstance
- TerminateInstance
- WaitForAction

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

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	Descrição	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	Descrição	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
output	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.

O EC2 Image Builder executa o AWS Systems Manager (Systems Manager) Agent nas instâncias do EC2 que ele executa para compilar 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 que criam AMIs.

Tempo limite padrão: 30 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	Descrição	Tipo	Obrigatório	Padrão	Restrições
instanceId	A instância de compilação à qual aplicar as	String	Sim		Deve ser o ID da instância de saída da etapa

Nome da entrada	Descrição	Tipo	Obrigatório	Padrão	Restrições
	configurações de metadados.				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	Descrição	Tipo
osVersion	O nome e a versão do sistema operacional coletados da instância de compilação.	String
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

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	Descrição	Tipo	Obrigatório	Padrão	Restrições
instanceId	O ID da instância na qual a varredura foi executada.	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	Descrição	Tipo
runCommandId	O ID do sendCommand do Systems Manager que executou o script para coletar as descobertas.	String

Nome da saída	Descrição	Tipo
status	O status retornado do sendCommand do Systems Manager.	String
output	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 com base em uma instância em execução com a API `CreateImage` do Amazon EC2. 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

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	Descrição	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	Descrição	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
```

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

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	Descrição	Tipo	Obrigatório	Padrão	Restrições
instanceId	O ID da instância na qual os componentes devem ser executados.	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	Descrição	Tipo
runCommandId	O ID do sendCommand do Systems Manager que	String

Nome da saída	Descrição	Tipo
	executou os componentes na instância.	
status	O status retornado do sendCommand do Systems Manager.	String
output	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
```

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: 60 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	Descrição	Tipo	Obrigatório	Padrão	Restrições
waitFor	A condição a ser aguardada antes de concluir a etapa do fluxo de trabalho e passar para a próxima etapa.	String	Sim		No momento, o Image Builder é compatível com `ssmAgent`.

Saídas: a tabela a seguir inclui saídas para essa ação de etapa.

Nome da saída	Descrição	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
```

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: 12 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	Descrição	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	String	Sim		

Nome da entrada	Descrição	Tipo	Obrigatório	Padrão	Restrições
	a ser executado.				
parâmetros	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	Descrição	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
output	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

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	Descrição	Tipo	Obrigatório	Padrão	Restrições
instanceId	O ID da instância na qual executar o documento	String	Sim		Deve ser o ID da instância de saída da etapa do fluxo de

Nome da entrada	Descrição	Tipo	Obrigatório	Padrão	Restrições
	`AWSEC2-RunSysprep` .				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	Descrição	Tipo
runCommandId	O ID do sendCommand do Systems Manager que executou o documento `AWSEC2-RunSysprep` na instância.	String
status	O status retornado do sendCommand do Systems Manager.	String
output	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

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	Descrição	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	Descrição	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
output	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

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	Descrição	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. Essa 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.

Tempo limite padrão: 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	Descrição	Tipo	Obrigatório	Padrão	Restrições
snsTopicArn	Um ARN opcional de	String	Não		

Nome da entrada	Descrição	Tipo	Obrigatório	Padrão	Restrições
	tópico do SNS para enviar uma notificação quando a etapa do fluxo de trabalho estiver pendente.				

Saídas: a tabela a seguir inclui saídas para essa ação de etapa.

Nome da saída	Descrição	Tipo
ação	A ação retornada pela ação de API SendWorkflowStepAction.	String (`RESUME` ou `STOP`)
razão	O motivo da ação retornada.	String

Exemplo

Especifique a ação de etapa no documento de fluxo de trabalho.

```
- name: SendEventAndWait
  action: WaitForAction
  onFailure: Abort
  inputs:
    snsTopicArn: arn:aws:sns:us-west-2:111122223333:ExampleTopic
```

Use a saída do valor da ação da etapa no documento do fluxo de trabalho.

```
$.stepOutputs.SendEventAndWait.reason
```

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. Os valores das variáveis dinâmicas são representados como seletores JSONPath com nós estruturais que identificam a variável de destino de modo exclusivo.

Estrutura variável de fluxo de trabalho dinâmico JSONPath

```
$.<document structure>.[<step name>].<variable name>
```

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 os nós compatíveis de estrutura de documentos de fluxo de trabalho JSONPath.

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 indicar a saída de qualquer ação da etapa anterior com essas variáveis dinâmicas JSONPath. 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
```

Exemplo

```
  - name: ApplyTestComponents
    action: ExecuteComponents
    onFailure: Abort
```

```
    inputs:
      instanceId.$: "$.stepOutputs.LaunchTestInstance.instanceId"
```

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	Descrição	Tipo	Valor de exemplo
cloudWatchLogGrupo	O nome do grupo CloudWatch Logs para registros de saída. Formato: `/aws/imagebuilder/<recipe-name>`	String	`/aws/imagebuilder/sampleImageRecipe`
cloudWatchLogTransmitir	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 resultado	Booleano	`true` | `false`

Nome da variável	Descrição	Tipo	Valor de exemplo
	s de varredura de imagens.		
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`

Nome da variável	Descrição	Tipo	Valor de exemplo
s3Logs	Um objeto JSON que contém a configuração dos logs do S3 que o Image Builder grava.	Objeto JSON	*{'S3Logs': {'s3':BucketName' repositório de amostras ', 's3KeyPrefix': 'ib-logs'}}*
securityGroups	Os IDs do grupo de segurança aplicáveis à criação e ao teste de instâncias.	List [String]	*[sg-1234567890abcdef1, sg-11112222333344445]*
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:imagebuilder:*us-east-1* :*111122223333* :image/*sampleImage* /*1.0.0/1*
subnetId	O ID da sub-rede na qual inicializar as instâncias de compilação e teste.	String	*subnet-1234567890abcdef1*

Nome da variável	Descrição	Tipo	Valor de exemplo
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`
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 as seguintes declarações condicionais que realizam comparações de valores (igual, menor que, maior que) com base nos tipos de dados que ela compara (string ou número).

Declarações condicionais 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.
- 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

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

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

A declaração condicional `booleanEquals` compara os valores boolianos e retorna verdadeiro se os valores boolianos tiverem correspondência exata.

O exemplo a seguir determina se `collectImageScanFindings` está habilitado.

```
if:
  - booleanEquals: true
    value: '$.imagebuilder.collectImageScanFindings'
```

Declaração condicional: stringEquals

A declaração condicional `stringEquals` compara duas cadeias de caracteres e retorna verdadeiro se as cadeias forem 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

A declaração condicional `numberEquals` compara dois números e retorna verdadeiro se os números forem 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
```

Importar e exportar imagens de máquina virtual (VM) do com o EC2 Image Builder

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 e usá-la como imagem base para suas fórmulas de imagens.

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)

- Raw

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 de usuários, consulte Importar uma VM (console). Para obter mais informações sobre a importação como parte de sua fórmula de imagem, consulteConfiguração de importação de VM.

Importar uma VM para o Image Builder (AWS CLI)

Para importar uma VM de discos para uma AMI e criar um recurso de imagem do Image Builder que você possa referenciar imediatamente, siga estas etapas no AWS CLI:

1. Inicie uma importação de VM, com o comando Amazon EC2 VM import-image Import/Export no AWS CLI. Anote o ID da tarefa retornado pelo comando. Você precisará dele para a próxima etapa. Para obter mais informações, consulte Como importar uma VM como uma imagem usando o VM Import/Export no Guia do usuário de 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 ação da API Image Builder. Para revisar os parâmetros de solicitação de comando da API, consulte o ImportVmImagecomando na Referência da API do 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, required) — 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.
- description (string) — A descrição da fórmula de imagem.
- platform (string, required) — A plataforma do sistema operacional para a VM importada.
- vmImportTaskId (string, obrigatório) — O `ImportTaskId` (AWS CLI) do processo de importação da VM do Amazon EC2. 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.
- clientToken (string, obrigatório) – Um identificador exclusivo e que diferencia maiúsculas e minúsculas que você fornece para garantir a idempotência da solicitação. Para obter mais informações, consulte Garantir idempotência) na Referência da API do Amazon EC2.
- 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",
    "clientToken": "asz1231231234cs3z",
    "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 e o Linux usa a barra (/).

Distribua discos VM a partir da sua imagem build ()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 ter mais informações, consulte Crie configurações de distribuição para discos VM de saída (AWS CLI).

Compartilhar recursos do EC2 Image Builder

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

- Componentes
- Imagens
- Fórmulas

Esta seção fornece informações para ajudar você a compartilhar esses recursos do EC2 Image Builder.

Conteúdo da seção

Como trabalhar com componentes, imagens e fórmulas compartilhados no EC2 Image Builder

O compartilhamento de componentes, imagens e receitas permite que os proprietários de recursos compartilhem configurações de software com outras pessoas Contas da AWS ou dentro de uma AWS organização. Você pode gerenciar o compartilhamento de recursos de forma centralizada e definir um conjunto de contas com as quais a configuração pode ser compartilhada.

Nesse modelo, o Conta da AWS proprietário do componente, imagem ou receita (proprietários) o compartilha com outros Contas da AWS (consumidores). Os consumidores podem associar um componente compartilhado a seus pipelines de imagem para consumir automaticamente as atualizações do componente, imagem ou fórmula compartilhados.

O proprietário de um componente, imagem ou fórmula pode compartilhar esses recursos com:

- 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.

Pré-requisitos para compartilhar componentes, imagens e fórmulas

Para compartilhar um componente, imagem ou fórmula do Image Builder:

- Você deve possuir o componente, a imagem ou a fórmula em seu Conta da AWS. 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 de destino.
- Para compartilhar seus recursos do Image Builder AWS Organizations e usar OUs 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 máquina 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.

Serviços relacionados

AWS Resource Access Manager

O compartilhamento de componentes, imagens e receitas se integra com AWS Resource Access Manager (AWS RAM). AWS RAM é um serviço que permite que você compartilhe seus AWS recursos com qualquer AWS conta ou por meio de AWS Organizations. Com AWS RAM, você compartilha recursos de sua propriedade criando 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.

Para obter mais informações sobre AWS RAM, consulte o Guia AWS RAM do usuário.

Compartilhar entre regiões

Componentes, imagens e fórmulas compartilhados só podem ser compartilhados em uma região específica da AWS . Quando você compartilha esses recursos, eles não se replicam em todas as regiões.

Compartilhar um componente, uma imagem ou uma fórmula

Para compartilhar um componente, uma imagem ou uma fórmula do Image Builder, você deve adicioná-la a um compartilhamento de recursos. Um compartilhamento de recursos é um AWS RAM recurso que permite que você compartilhe seus recursos entre AWS contas. Um compartilhamento de recursos especifica os recursos a serem compartilhados, e os consumidores com os quais compartilhá-los. Para adicionar o componente, a imagem ou a receita a um novo compartilhamento de recursos, primeiro você deve criar o compartilhamento de recursos usando o AWS RAM console.

Se você faz parte de uma organização AWS Organizations e o compartilhamento dentro de sua organização está ativado, os consumidores em sua organização recebem automaticamente acesso ao componente, imagem ou receita compartilhados. Caso contrário, os consumidores receberão um convite para participar do compartilhamento de recursos e acesso ao recurso compartilhado depois de aceitar o convite.

Os recursos a seguir estão disponíveis para compartilhar seus 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 de RAM

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 você 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 compnentes selcionados.

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 compnentes selcionados.

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 componente, uma imagem ou uma fórmula

Para cancelar o compartilhamento de um componente, uma imagem ou uma fórmula de sua propriedade, é necessário removê-la do compartilhamento de recursos. Você pode fazer isso usando o AWS Resource Access Manager console ou AWS CLI o.

> **Note**
>
> Para cancelar o compartilhamento de um componente, de uma imagem ou de uma fórmula, o consumidor não pode ter dependências dela. O consumidor deve remover todas as

dependências dos recursos compartilhados antes que o proprietário possa cancelar o compartilhamento.

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 parar de compartilhar o recurso.

Como identificar um componente, imagem ou fórmula compartilhada

Proprietários e consumidores podem identificar imagens e fórmulas de imagem compartilhadas usando os comandos do Image Builder no AWS CLI.

Identificar um componente compartilhado

Execute o comando list-components para obter uma lista dos componentes que você possui e dos componentes que são compartilhados com você. O comando get-component mostra o Conta da AWS ID do proprietário do componente.

Identificar uma imagem compartilhada

Execute o comando list-images para obter uma lista das imagens que você possui e das imagens que são compartilhadas com você. O comando get-image mostra o Conta da AWS ID do proprietário da imagem.

Identificar uma imagem de contêiner compartilhada

Execute o comando list-images para obter uma lista das imagens que você possui e das imagens que são compartilhadas com você. O comando get-image mostra o ID Conta da AWS do proprietário da imagem.

Identificar uma fórmula compartilhada

Execute o list-image-recipescomando para obter uma lista das receitas de imagens que você possui e das receitas de imagens que são compartilhadas com você. O get-image-recipecomando mostra o Conta da AWS ID do proprietário da receita da imagem.

Identificar uma fórmula de contêiner compartilhada

Execute o list-container-recipescomando para obter uma lista das receitas de contêiner que você possui e das receitas de contêiner que são compartilhadas com você. O get-container-recipecomando mostra o Conta da AWS ID do proprietário da receita do contêiner.

Permissões compartilhadas de componentes, imagens e fórmulas

Permissões para proprietários

Os proprietários não podem excluir um componente, imagem ou fórmula de imagem compartilhada até que deixem de ser compartilhados. Um proprietário não pode deixar de compartilhar esses recursos até que nenhum dos consumidores dependa deles.

Permissões para consumidores

Os consumidores podem ler um componente, imagem ou fórmula de imagem, mas não podem modificá-los de forma alguma. Eles não podem visualizar ou modificar esses recursos se forem de propriedade de outros consumidores ou do proprietário do recurso. Os consumidores podem usar componentes e imagens compartilhados em fórmulas de imagens para criar imagens personalizadas. Os consumidores podem usar fórmulas de imagens compartilhadas em fórmulas de imagens para suas próprias imagens personalizadas.

Faturamento e medição

O uso do EC2 Image Builder é gratuito.

Limites de recurso

Componentes, imagens e fórmulas de imagens compartilhadas contam somente para os limites de recursos correspondentes do proprietário. Os limites de recursos dos consumidores não são afetados por recursos que foram compartilhados com eles.

Marcar recursos do EC2 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 para às seguintes tags dinâmicas:

- `- {{imagebuilder:buildDate}}`

  Será resolvido como data/hora da compilação no momento da compilação.

- `- {{imagebuilder:buildVersion}}`

  Resolve para uma versão de compilação, que é um número localizado no final de um Image Builder Amazon Resource Name (ARN). 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 às suas AMIs de saída.

- `"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 Image Builder que foi usado para criar a AMI.

Conteúdo

Marcar um recurso (AWS CLI)

O exemplo a seguir mostra como usar um comando CLI do imagebuilder 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
```

Desmarcar um recurso (AWS CLI)

O exemplo a seguir 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
```

Indique todas as tags de um recurso específico (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
```

Exclua recursos do EC2 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 se preocupar com a 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. Até lá, talvez não esteja claro se você pode se livrar deles com segurança.

A exclusão de recursos não exclui nenhuma AMIs do Amazon EC2 ou imagens de contêiner do Amazon ECR criadas 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 os comandos. AWS CLI

> **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

Excluir recursos usando o AWS Management Console

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 do nome do pipeline para selecionar o pipeline do 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 do nome da fórmula para selecionar a fórmula do 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 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 usando o 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

Excluir um pipeline de imagens (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
```

Excluir fórmula de imagem (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 pipelines do EC2 Image Builder usando o console

Os pipelines de imagem do Image Builder fornecem uma estrutura de automação para criar e manter AMIs e imagens de contêiner personalizadas. Os pipelines oferecem as seguintes funcionalidades:

- Montar a imagem base, os componentes para construção e teste, a configuração da infraestrutura e as 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.

Conteúdo

Indicar e visualizar detalhes do pipeline

Esta seção descreve as várias maneiras de encontrar informações e visualizar detalhes de seus pipelines de imagem do EC2 Image Builder.

Detalhes do pipeline

- Indicar pipelines de imagens (AWS CLI)
- Obter detalhes do pipeline de imagens (AWS CLI)

Indicar pipelines de imagens (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
```

Obter detalhes do pipeline de imagens (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
```

Crie e atualize pipelines de imagens da AMI

Você pode instalar, configurar e gerenciar pipelines de imagem de AMI usando o console do Image Builder usando a API do Image Builder ou com comandos imagebuilder na AWS CLI. É possível usar o assistente do console Criar pipeline de imagem para obter orientação nas seguintes etapas:

- Especifique os detalhes do pipeline, como nome, descrição e tags de recursos.
- Selecione uma fórmula de imagem de AMI que inclua uma imagem de base proveniente de imagens gerenciadas com o início rápido ou imagens que você criou ou que foram compartilhadas com você. A fórmula também inclui componentes que realizam as seguintes tarefas nas instâncias do EC2 que o Image Builder usa para compilar 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 do KMS para criptografia, configurar o compartilhamento de AMI ou a configuração de licenças ou configurar um modelo de execução para as 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, consulteCriar um pipeline de imagem usando o assistente do console do EC2 Image Builder.

Conteúdo

Criar um pipeline de imagem da AMI (AWS CLI)

Você pode criar um pipeline de imagem da AMI com um arquivo JSON que contém detalhes de configuração como entrada para o comando create-image-pipeline no AWS CLI.

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 Use expressões cron no EC2 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 da AMI:

- ARN da fórmula de imagem
- ARN de configuração de infraestrutura

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,
    "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 e o Linux usa 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 pipelines de imagem da AMI (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 Atualizar pipelines de imagem da AMI (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 Image pipelines 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 Nome do pipeline. Isso 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 ter mais informações, consulte Gerenciar fórmulas.

- 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 essa 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.
- 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, a 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 estiver pronto para salvar as atualizações feitas na configuração da sua 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 AWS nas 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 de 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.
- Editar 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 como programar sua compilação usando expressões cron, consulte Use expressões cron no EC2 Image Builder.

Atualizar pipelines de imagem da AMI (AWS CLI)

Você pode atualizar um pipeline de imagem da AMI usando um arquivo JSON como entrada para o comando update-image-pipeline no AWS CLI. Para configurar o arquivo JSON, você deve ter nomes do recurso da Amazon (ARNs) para referenciar os seguintes recursos existentes:

- Pipeline de imagens a ser atualizado
- Fórmula de 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 e o Linux usa 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, por meio da API Image Builder ou com os comandos de imagebuilder no 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, consulteCriar um pipeline de imagens de contêiner usando o assistente do console do EC2 Image Builder.

Conteúdo

Criar um pipeline de imagem de contêiner (AWS CLI)

Você pode criar um pipeline de imagens de contêiner usando um arquivo JSON como entrada para o comando create-image-pipeline na AWS CLI.

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 Use expressões cron no EC2 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 e o Linux usa 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 imagem de contêiner (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 Atualizar pipelines de imagens de contêiner (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 Image pipelines 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 Nome do pipeline. Isso 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 ter mais informações, consulte Criar uma nova versão de receita 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 essa 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, a 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 estiver pronto para salvar as atualizações feitas na configuração da sua 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 Use expressões cron no EC2 Image Builder.

Atualizar pipelines de imagens de contêiner (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 nomes do recurso da Amazon (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 e o Linux usa 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 de imagem para seu pipeline do EC2 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 EC2 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 (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

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.

Executar seu pipeline de imagem

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.

Use expressões cron no EC2 Image Builder

Use expressões cron para o 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 do Tempo Coordenado Universal (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

O EC2 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 dos ay-of-week campos Dia e D, 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 ay-of-wee k posição D 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 ay-of-month campo D especific a um dia da semana. No ay-of-month campo D, se você inserir um número antes do`W`, isso significa que você deseja segmentar o

Curinga	Descrição
	dia 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

- Você não pode especificar os ay-of-week campos D ay-of-month e D 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 EC2 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 de classificação no EC2 Image Builder

Uma expressão rate começa quando você cria a regra de evento programado e, em seguida, e a executa em sua programação definida.

As expressões rate tem 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.

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 que você cria para responder a eventos no barramento ou em uma programação.

> **Note**
>
> Os barramentos de eventos são específicos para uma região. A regra e o alvo devem estar localizados na mesma região.

Conteúdo

EventBridge termos

Esta seção contém um resumo dos termos para ajudar você a entender como se EventBridge integra aos seus pipelines do Image Builder.

Evento

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.

Destino

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.

Schedule

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 Image pipelines 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 Nome do pipeline. Isso 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 a opção 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 Add target.
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.

Integre produtos e serviços no EC2 Image Builder

O EC2 Image Builder se AWS Marketplace integra a Serviços da AWS outros aplicativos para ajudá-lo a criar imagens de máquina personalizadas robustas 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 a partir da API, CLI ou SDK do Image Builder, você pode especificar um produto de imagem de nome do recurso da Amazon (ARN) para usar como sua imagem base.
- AWSTOE componentes — Os componentes que você especifica em suas receitas podem realizar ações de criação e teste, por exemplo, para instalar software ou realizar validação de conformidade. Alguns produtos de imagem dos quais você AWS Marketplace assina podem incluir um componente complementar que você pode usar em suas fórmula. 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 o ajudam a rastrear sua atividade, solucionar problemas de compilação de imagem e criar automações com base em notificações de eventos.

- AWS CloudTrail— Monitore os eventos do Image Builder enviados para CloudTrail o. Para obter mais informações sobre CloudTrail, consulte O que é AWS CloudTrail? no Guia do AWS CloudTrail usuário.
- Amazon CloudWatch Logs — Monitore, armazene e acesse seus arquivos de log do Image Builder. Como opção, você pode salvar seus registros em um bucket do S3. Para obter mais informações sobre CloudWatch registros, consulte O que é Amazon CloudWatch Logs? no Guia do usuário do Amazon CloudWatch Logs.
- 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 da instância de teste do EC2 que o Image Builder inicia quando você cria 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 escaneamentos e definição de preço, consulte O que é o 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, consulte Verificação de imagens de contêiner do Amazon ECR no Guia do usuário do Amazon Inspector.

  > **Note**
  >
  > O Amazon Inspector é um atributo pago.

- AWS Marketplace— Veja uma lista de suas assinaturas de AWS Marketplace produtos atuais 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 como gerenciar AWS Marketplace assinaturas, consulte o Guia do AWS Marketplace comprador.
- 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

AWS CloudTrail integração no Image Builder

Este serviço oferece suporte AWS CloudTrail. CloudTrail é um serviço que registra AWS chamadas para você Conta da AWS e entrega arquivos de log para um bucket do Amazon S3. Usando as informações coletadas por CloudTrail, você pode determinar quais solicitações foram feitas com sucesso Serviços da AWS, quem fez a solicitação, quando ela foi feita e assim por diante. Para obter mais informações sobre a CloudTrail integração com o Image Builder, consulteRegistrando chamadas da API do EC2 Image Builder usando AWS CloudTrail.

Para saber mais sobre CloudTrail, inclusive como ativá-lo e encontrar seus arquivos de log, consulte o Guia AWS CloudTrail do usuário.

Integração do Amazon CloudWatch Logs no Image Builder

CloudWatch O suporte a registros está ativado por padrão. Os registros são retidos na instância durante o processo de criação e transmitidos para CloudWatch o Logs. Os registros da instância são removidos da instância antes da criação da imagem.

Os registros de compilação são transmitidos para o seguinte grupo e stream do Image Builder CloudWatch Logs:

LogGroup:

```
/aws/imagebuilder/ImageName
```

LogStream (x.x.x/x):

```
ImageVersion/ImageBuildVersion
```

Você pode desativar o streaming de CloudWatch registros removendo as seguintes permissões associadas ao perfil da instância.

```
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "logs:CreateLogStream",
            "logs:CreateLogGroup",
            "logs:PutLogEvents"
        ],
        "Resource": "arn:aws:logs:*:*:log-group:/aws/imagebuilder/*"
    }
]
```

Para solucionar problemas avançados, você pode executar comandos e scripts predefinidos usando AWS Systems Manager Executar comando. Para ter mais informações, consulte Solução de problemas do EC2 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 instância do EC2 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

- Invocar o Run Command do Amazon EC2
- 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 se integra aos pipelines de imagem do Image Builder, consulte. Use EventBridge regras com pipelines do Image Builder

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
  - Alta
  - 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 para seu pipeline, consulteConfigure escaneamentos de segurança para imagens do Image Builder no AWS Management Console.

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 ter mais informações, consulte Gerencie descobertas de segurança para imagens do Image Builder no AWS Management Console.

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?

> **Note**
>
> Os fornecedores de produtos de dados devem atender aos requisitos de elegibilidade do AWS Data Exchange. Para obter mais informações, consulte Fornecimento de produtos de dados no Data Exchange do AWS no Guia do usuário do Data Exchange do AWS .

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.
- Veja uma lista de suas assinaturas atuais de AWS Marketplace produtos.
- Use um produto de AWS Marketplace imagem como imagem base para uma receita do Image Builder.

Para produtos que incluem componentes associados AWS Task Orchestrator and Executor (AWSTOE), você pode filtrar pelo proprietário do produto no console e na API, SDK e CLI. Para ter mais informações, consulte Listar AWSTOE componentes.

Encontre produtos de AWS Marketplace imagem no console do Image Builder

O Image Builder se AWS Marketplace integra para mostrar suas assinaturas de produtos de imagem diretamente da AWS Marketplaceseção no console do Image Builder. Você também pode pesquisar produtos de AWS Marketplace imagem na página Produtos de imagem sem sair do console do Image Builder.

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 Produtos de imagem na AWS Marketplaceseção.
3. A página Produtos de imagem mostra um resumo dos produtos de imagem nos quais você se inscreveu na guia Assinaturas, ou você pode pesquisar produtos de imagem na guia. AWS Marketplace

   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.

   AWS Marketplace

   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 machine learning.

- 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.

Subscriptions

Essa guia mostra uma lista dos produtos de imagem nos quais você se inscreveu. AWS Marketplace 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.
- 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 que você assinou.
- Se houver um componente associado incluído em seu produto inscrito, o Image Builder exibirá um link para os detalhes do AWSTOE 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 inscrito como imagem base para uma nova fórmula, selecione um produto inscrito e escolha Criar nova fórmula. 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.

Use um produto de AWS Marketplace imagem nas receitas do Image Builder

No console do Image Builder, há duas maneiras de criar uma nova fórmula de imagem com base em um de seus produtos de imagem inscritos.

1. Você pode começar na página Produtos de imagem da seguinte forma:
    1. Abra o console do EC2 Image Builder em https://console.aws.amazon.com/imagebuilder/.
    2. No painel de navegação, escolha Produtos de imagem na AWS Marketplaceseção.
    3. Abra a guia Assinaturas.
    4. Selecione o produto de imagem inscrito para usar como imagem base em sua fórmula.
    5. Escolha Criar nova fórmula. Isso abre a página Criar fórmula com a opção de AWS Marketplace imagens e seu produto de imagem inscrito pré-selecionado.
    6. Definir configurações restantes da sua fórmula como faria normalmente. Para obter mais informações sobre fórmulas de imagem, consulte Criar uma nova versão de uma fórmula de imagem.
2. Você também pode abrir a página Criar receita e selecionar um produto de AWS Marketplace imagem para usar como imagem base.
    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 Create image recipe (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, consulteEncontre produtos de AWS Marketplace imagem no console do Image Builder.
6. Insira os detalhes restantes, como de costume, e escolha Criar fórmula.

> **Note**
>
> Se sua assinatura de produto de imagem incluir um componente de AWSTOE compilação, você poderá selecioná-lo na lista Componentes de criação. Selecione na lista `Third party managed` de tipos do proprietário do componente para vê-lo. Se a assinatura do produto incluir um componente de AWSTOE teste, siga o mesmo procedimento da lista de componentes de teste.

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 vinculados ao serviço para o EC2 Image Builder.

```
{
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::123456789012:role/aws-service-role/
imagebuilder.amazonaws.com/AWSServiceRoleForImageBuilder"
    },
    "Action": [
      "kms:GenerateDataKey*",
      "kms:Decrypt"
    ],
    "Resource": "*"
  }]
```

```
}
```

É possível usar um dos seguintes métodos para obter um ARN.

AWS Management Console

Para obter o ARN da função vinculada ao serviço que o Image Builder criou em sua conta a partir do AWS Management Console, siga estas etapas:

1. Abra o console IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação à esquerda, escolha Roles.
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": 123794003928538027489912 4227,
  "arn": "arn:aws:imagebuilder:us-west-1:123456789012:image/example-linux-
image/1.0.0/3",
  "name": "example-linux-image",
  "version": "1.0.0",
  "type": "AMI",
  "buildVersion": 3,
  "state": {
    "status": "AVAILABLE"
  },
  "platform": "Linux",
  "imageRecipe": {
    "arn": "arn:aws:imagebuilder:us-west-1:123456789012:image-recipe/example-linux-
image/1.0.0",
    "name": "amjule-barebones-linux",
    "version": "1.0.0",
    "components": [
      {
        "componentArn": "arn:aws:imagebuilder:us-west-1:123456789012:component/update-
linux/1.0.2/1"
      }
    ],
    "platform": "Linux",
    "parentImage": "arn:aws:imagebuilder:us-west-1:987654321098:image/amazon-linux-2-
x86/2022.6.14/1",
    "blockDeviceMappings": [
      {
        "deviceName": "/dev/xvda",
        "ebs": {
          "encrypted": false,
          "deleteOnTermination": true,
          "volumeSize": 8,
          "volumeType": "gp2"
```

```
        }
      }
    ],
    "dateCreated": "Feb 24, 2021 12:31:54 AM",
    "tags": {
      "internalId": "1a234567-8901-2345-bcd6-ef7890123456",
      "resourceArn": "arn:aws:imagebuilder:us-west-1:123456789012:image-recipe/example-
linux-image/1.0.0"
    },
    "workingDirectory": "/tmp",
    "accountId": "462045008730"
  },
  "sourcePipelineArn": "arn:aws:imagebuilder:us-west-1:123456789012:image-pipeline/
example-linux-pipeline",
  "infrastructureConfiguration": {
    "arn": "arn:aws:imagebuilder:us-west-1:123456789012:infrastructure-configuration/
example-linux-infra-config-uswest1",
    "name": "example-linux-infra-config-uswest1",
    "instanceProfileName": "example-linux-ib-baseline-admin",
    "tags": {
      "internalId": "234abc56-d789-0123-a4e5-6b789d012c34",
      "resourceArn": "arn:aws:imagebuilder:us-west-1:123456789012:infrastructure-
configuration/example-linux-infra-config-uswest1"
    },
    "logging": {
      "s3Logs": {
        "s3BucketName": "12345-example-linux-testbucket-uswest1"
      }
    },
    "keyPair": "example-linux-key-pair-uswest1",
    "terminateInstanceOnFailure": true,
    "snsTopicArn": "arn:aws:sns:us-west-1:123456789012:example-linux-ibnotices-
uswest1",
    "dateCreated": "Feb 24, 2021 12:31:55 AM",
    "accountId": "123456789012"
  },
  "imageTestsConfigurationDocument": {
    "imageTestsEnabled": true,
    "timeoutMinutes": 720
  },
  "distributionConfiguration": {
    "arn": "arn:aws:imagebuilder:us-west-1:123456789012:distribution-configuration/
example-linux-distribution",
    "name": "example-linux-distribution",
```

```
    "dateCreated": "Feb 24, 2021 12:31:56 AM",
    "distributions": [
      {
        "region": "us-west-1",
        "amiDistributionConfiguration": {}
      }
    ],
    "tags": {
      "internalId": "345abc67-8910-12d3-4ef5-67a8b90c12de",
      "resourceArn": "arn:aws:imagebuilder:us-west-1:123456789012:distribution-
configuration/example-linux-distribution"
    },
    "accountId": "123456789012"
  },
  "dateCreated": "Jul 28, 2022 1:13:45 AM",
  "outputResources": {
    "amis": [
      {
        "region": "us-west-1",
        "image": "ami-01a23bc4def5a6789",
        "name": "example-linux-image 2022-07-28T01-14-17.416Z",
        "accountId": "123456789012"
      }
    ]
  },
  "buildExecutionId": "ab0cd12e-34fa-5678-b901-2c3456d789e0",
  "testExecutionId": "6a7b8901-cdef-234a-56b7-8cd89ef01234",
  "distributionJobId": "1f234567-8abc-9d0e-1234-fa56b7c890de",
  "integrationJobId": "432109b8-afe7-6dc5-4321-0ba98f7654e3",
  "accountId": "123456789012",
  "osVersion": "Amazon Linux 2",
  "enhancedImageMetadataEnabled": true,
  "buildType": "USER_INITIATED",
  "tags": {
    "internalId": "901e234f-a567-89bc-0123-d4e567f89a01",
    "resourceArn": "arn:aws:imagebuilder:us-west-1:123456789012:image/example-linux-
image/1.0.0/3"
  }
}
```

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 e componentes de AWS Marketplace conformidade. AWSTOE

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 fortalecidas CIS e componentes de fortalecimento, consulte os Guias de início rápido no portal de suporte do site 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 ter 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 fortalecimento do STIG gerenciados pela Amazon para o EC2 Image Builder.

Monitore eventos e logs no EC2 Image Builder

Para manter a confiabilidade, a disponibilidade e o desempenho de seus pipelines do EC2 Image Builder, é importante monitorar eventos e logs. 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

- Registrando chamadas da API do EC2 Image Builder usando AWS CloudTrail

Registrando chamadas da API do EC2 Image Builder usando AWS CloudTrail

O EC2 Image Builder está integrado AWS CloudTrail com, um serviço que fornece um registro das ações de todas as chamadas de API realizadas por um usuário, função ou AWS serviço por meio da API Image Builder. CloudTrail captura 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 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 sobre isso 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 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 S3 que você especificar. Além disso, você pode configurar outros Serviços da AWS para analisar e agir com base nos dados do evento coletados nos CloudTrail registros. Para 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.
- Recebendo arquivos de CloudTrail log de várias contas.

CloudTrail registra todas as ações do Image Builder que estão documentadas na Referência da API do 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:

- Se a solicitação foi feita com credenciais de usuário raiz ou do usuário do IAM.
- Se a solicitação foi feita com credenciais de segurança temporárias de um perfil ou de um usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para obter mais informações sobre como determinar quem solicitou um evento, consulte o elemento CloudTrail userIdentity.

Segurança no EC2 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 compatibilidade que se aplicam ao EC2 Image Builder, consulte Escopo da Serviços da AWS por programa de compatibilidade.
- 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 no EC2 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 seu conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para ter mais informações sobre a privacidade de dados, consulte as Perguntas frequentes sobre privacidade de dados. Para ter mais informações sobre a proteção de dados na Europa, consulte a postagem do blog AWS Shared Responsibility Model and GDPR 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 AWS IAM Identity Center 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 os recursos. AWS Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail.
- 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 sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para ter mais informações sobre endpoints do FIPS, consulte Federal Information Processing Standard (FIPS) 140-2.

É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de e-mail dos seus clientes, em marcações 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, a API ou AWS os SDKs. AWS CLI Quaisquer dados inseridos em tags ou campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de

diagnóstico. Se você fornecer um URL para um servidor externo, recomendamos fortemente que não sejam incluídas informações de credenciais no URL para validar a solicitação a esse servidor.

Criptografia e gerenciamento de chaves no EC2 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 .

```
{
 "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.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "Allow access to the workflow key",
   "Effect": "Allow",
   "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
   ],
   "Resource": "arn:aws:kms:us-west-2:111122223333:key/key_ID",
   "Condition": {
    "StringLike": {
     "kms:EncryptionContext:aws:imagebuilder:arn":
 "arn:aws:imagebuilder:*:111122223333:workflow/*"
```

```
    }
   }
  }
 ]
}
```

AWS CloudTrail 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 EC2 Image Builder

O Image Builder não armazena nenhum dos seus registros no serviço. Todos os registros são salvos na sua instância do Amazon EC2 que é usada para criar a imagem ou nos seus registros de automação do Systems Manager.

Privacidade de tráfego entre redes no EC2 Image Builder

As conexões são protegidas entre o Image Builder e locais locais, entre AZs em uma AWS região e entre AWS regiões por meio de HTTPS. Não há conexões diretas entre contas.

Gerenciamento de identidade e acesso para o EC2 Image Builder

Tópicos

Público

A forma como você usa AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz no Image Builder.

Usuário do serviço: se você usa o serviço Image Builder para fazer o trabalho, o administrador fornece as credenciais e as permissões necessárias. À medida que usar mais atributos do Image

Builder para fazer seu trabalho, você poderá precisar de permissões adicionais. Entender como o acesso é gerenciado pode ajudá-lo a solicitar as permissões corretas ao seu administrador. Se não for possível acessar um atributo no Image Builder, consulte Solução de problemas de identidade e acesso do EC2 Image Builder.

Administrador do serviço: se você for o responsável pelos recursos do Image Builder na sua empresa, provavelmente terá acesso total ao Image Builder. Cabe a você determinar quais atributos e recursos do Image Builder os usuários do seu serviço devem acessar. Assim, você deve enviar solicitações ao administrador do IAM para alterar as permissões dos usuários de seu serviço. Revise as informações nesta página para entender os Introdução ao IAM. Para saber mais sobre como a empresa pode usar o IAM com o Image Builder, consulte Como o EC2 Image Builder funciona com o IAM.

Administrador do IAM: se você for um administrador do IAM, talvez queira saber detalhes sobre como é possível criar políticas para gerenciar o acesso ao Image Builder. Para visualizar exemplos de políticas baseadas em identidade do Image Builder que podem ser usadas no IAM, consulte Políticas baseadas em identidade do Image Builder.

Autenticando 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 EC2 Image Builder funciona com o 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

É compatível com políticas baseadas em identidade	Sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. Essas

políticas controlam quais ações os usuários e funções podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Criar políticas do IAM no Guia do usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações ou recursos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas. Você não pode especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou função à qual ela está anexada. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte Referência de elementos da 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

Oferece suporte a políticas baseadas em recurso	Não

Políticas baseadas em recurso 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 recurso ao qual a política está anexada, a política define quais ações uma entidade principal especificada pode executar nesse recurso e em que condições. Você deve 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, você pode especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em recurso. Adicionar uma entidade principal entre contas à política baseada em recurso é apenas metade da tarefa de estabelecimento da relação de confiança. Quando o principal e o recurso são diferentes Contas da AWS, um administrador do IAM na conta confiável também deve conceder permissão à entidade principal (usuário ou função) para acessar o recurso. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em recurso

conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Para obter mais informações, consulte Como as funções do IAM diferem de políticas baseadas em recursos no Guia do usuário do IAM.

Ações de política para Image Builder

Oferece suporte a 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 você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da operação de AWS API associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Há também algumas operações que exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista das ações do Image Builder, consulte Ações definidas pelo EC2 Image Builder na Referência de autorização do serviço.

As ações de políticas no Image Builder usam o seguinte prefixo antes da ação:

```
imagebuilder
```

Para especificar várias ações em uma única instrução, 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

Oferece suporte a 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 `Resource` JSON especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou um elemento `NotResource`. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Isso pode ser feito para ações que oferecem suporte a um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem suporte a permissões em nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

Para obter uma lista dos tipos de recursos do Image Builder e seus ARNs, consulte Recursos definidos pelo EC2 Image Builder na Referência de autorização do serviço. Para saber com quais ações é possível 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 do 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` (ou bloco de `Condition`) permite que você especifique condições nas quais uma instrução está em vigor. O elemento `Condition` é opcional. É possível criar expressões

condicionais que usam atendentes de condição, como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único elemento `Condition`, a AWS os avaliará usando uma operação lógica `AND`. Se você especificar vários valores para uma única chave de condição, AWS avalia a condição usando uma `OR` operação lógica. Todas as condições devem ser atendidas para que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar as condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um recurso somente se ele estiver marcado com seu nome de usuário do IAM. Para mais informações, consulte Elementos de política do IAM: variáveis e tags no Guia do usuário do IAM.

AWS suporta chaves de condição globais e chaves de condição específicas do serviço. 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 de chaves de condição do Image Builder, consulte Chaves de condição do EC2 Image Builder na Referência de autorização do serviço. Para saber com quais ações e recursos é possível 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 no construtor de imagens

Oferece suporte a ACLs	Não

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou perfis da conta) têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

ABAC com Image Builder

Oferece suporte a ABAC (tags em políticas)	Parcial

O controle de acesso baseado em recurso (ABAC) é uma estratégia de autorização que define permissões com base em recursos. Em AWS, esses atributos são chamados de tags. Você pode anexar tags a entidades do IAM (usuários ou funções) e a vários AWS recursos. A marcação de entidades e recursos é a primeira etapa do ABAC. Em seguida, você cria políticas de ABAC para permitir operações quando a tag da entidade principal corresponder à tag do recurso que ela está tentando acessar.

O ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna um problema.

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 `aws:TagKeys` chaves de condição.

Se um serviço oferecer suporte às três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço oferecer suporte às três chaves de condição somente para alguns tipos de recursos, o valor será Parcial.

Para mais informações sobre o ABAC, consulte O que é ABAC? no Guia do Usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte Usar Controle de Acesso Baseado em recursos (ABAC) no Guia do Usuário do IAM.

Usar credenciais temporárias com o Image Builder

Oferece suporte a credenciais temporárias	Sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte Serviços da AWS “Trabalhe com o IAM” no Guia do usuário do IAM.

Você está usando credenciais temporárias se fizer login AWS Management Console usando qualquer método, exceto um nome de usuário e senha. Por exemplo, quando você acessa AWS usando o link de login único (SSO) da sua empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna perfis. Para mais informações sobre como alternar funções, consulte Alternar para uma função (console) no Guia do usuário do IAM.

Você pode criar manualmente credenciais temporárias usando a AWS API AWS CLI ou. Em seguida, você pode usar essas credenciais temporárias para acessar AWS. AWS recomenda que você gere

credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para mais informações, consulte Credenciais de segurança temporárias no IAM.

Permissões de entidade principal entre serviços para o Image Builder

Suporte para o recurso Encaminhamento de sessões de acesso (FAS)	Sim

Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado um principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Encaminhar sessões de acesso.

Funções de serviço para Image Builder

Oferece suporte a 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 mais informações, 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

Oferece suporte a perfis vinculados ao serviço	Não

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 para 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 pode editar as permissões para perfis vinculados ao serviço.

Para detalhes sobre a função vinculada ao serviço para o Image Builder, consulte Usar perfis vinculados ao serviço para o EC2 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 usados em uma política JSON, consulte Ações, recursos e chaves de condição do 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.

Gerenciamento do 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 `Resource` JSON especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou um elemento `NotResource`. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Isso pode ser feito para ações que oferecem suporte a um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem suporte a permissões em nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

O recurso de instância do Image Builder tem o nome do recurso da Amazon (ARN) a seguir.

```
arn:aws:imagebuilder:region:account-id:resource:resource-id
```

Para obter mais informações sobre o formato dos ARNs, consulte Amazon Resource Names (ARNs) e AWS Service Namespaces.

Por exemplo, para especificar a instância de `i-1234567890abcdef0` em sua instrução, use o ARN a seguir.

```
"Resource": "arn:aws:imagebuilder:us-east-1:123456789012:instance/i-1234567890abcdef0"
```

Para especificar todas as instâncias que pertencem a uma conta específica, use o caractere curinga (*).

```
"Resource": "arn:aws:imagebuilder:us-east-1:123456789012:instance/*"
```

Algumas ações do Image Builder, como as ações para a criação de recursos, não podem ser executadas em um recurso específico. Nesses casos, você deve utilizar o caractere curinga (*).

```
"Resource": "*"
```

Muitas ações da API do 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 as `CreateInfrastrucutreConfiguration` e `UpdateInfrastructureConfiguration` APIs.

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 as `CreateInfrastrucutreConfiguration` e `UpdateInfrastructureConfiguration` APIs.

Construtor de imagens: EC2 MetadataHttpTokens

Funciona com operadores de string.

Use esta chave para filtrar o acesso pelo Requisito de token HTTP de metadados de instâncias do EC2 especificado na solicitação.

Esse valor para essa chave pode ser `optional` ou`required`.

Disponibilidade — Essa chave está disponível somente para as `CreateInfrastrucutreConfiguration` e `UpdateInfrastructureConfiguration` APIs.

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 as `CreateInfrastrucutreConfiguration` e `UpdateInfrastructureConfiguration` APIs.

Exemplos

Para visualizar exemplos de políticas baseadas em identidade do Image Builder, consulte Políticas baseadas em identidade do EC2 Image Builder.

Políticas baseadas em recursos do construtor de imagens

As políticas baseadas em recursos especificam quais ações um diretor específico pode realizar no recurso 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, imagem ou fórmula de imagem, consulte. Compartilhar recursos do EC2 Image Builder

> **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 chaves de condição `imagebuilder:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`. Para obter mais informações sobre marcação de recursos no Image Builder, consulte Marcar um recurso (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 na 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 vinculados ao serviço para o EC2 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.

Políticas baseadas em identidade do EC2 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 incorrer em custos para a 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 mais informações, 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 mais informações 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: você pode adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, você pode 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 AWS

CloudFormation. Para obter mais informações, consulte Elementos de política JSON do IAM: Condition 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 acionáveis para ajudar você a criar políticas seguras e funcionais. Para mais informações, 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 a MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para mais informações, consulte Configuração de acesso à API protegido por MFA no Guia do usuário do IAM.

Para mais informações 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 do AWSImageBuilderReadOnlyAccess
- Política do AWSImageBuilderFullAccess

Para obter mais informações sobre políticas gerenciadas pelo Image Builder, consulte Como usar políticas gerenciadas para o 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:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sns:Publish"
            ],
            "Resource": "sns topic arn"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetInstanceProfile"
            ],
            "Resource": "instance profile role arn"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "instance profile role arn",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ec2.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": "bucket arn"
        }
```

```
    ]
}
```

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.

Políticas baseadas em recursos do EC2 Image Builder

Para informações sobre como criar um componente, consulte Gerencie componentes com o 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 instrução identifica o intervalo 54.240.143.* de endereços IP 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 valores IPv4 `aws:sourceIp` usam a notação CIDR padrão. Para obter mais informações, consulte Operadores de condição de endereço IP no Guia do usuário do IAM.

```
{
  "Version": "2012-10-17",
  "Id": "IBPolicyId1",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "imagebuilder.GetComponent:*",
      "Resource": "arn:aws:imagebuilder:::examplecomponent/*",
      "Condition": {
         "IpAddress": {"aws:SourceIp": "54.240.143.0/24"},
         "NotIpAddress": {"aws:SourceIp": "54.240.143.188/32"}
```

```
      }
    }
  ]
}
```

Como usar políticas gerenciadas para o 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 elas estão disponíveis para uso de todos os AWS clientes. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo cliente 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) é lançada ou novas operações de API são disponibilizadas para serviços existentes.

Para mais informações, consulte Políticas gerenciadas pela AWS no Manual do usuário do IAM.

Política do 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 AWS Management Console.

Detalhes da permissão

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 o Amazon EC2 descrever as ações 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”.

Exemplo de política

A seguir há um exemplo da política do AWSImageBuilderFullAccess.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "imagebuilder:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:ListTopics"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:Publish"
            ],
            "Resource": "arn:aws:sns:*:*:*imagebuilder*"
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "license-manager:ListLicenseConfigurations",
                "license-manager:ListLicenseSpecificationsForResource"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
imagebuilder.amazonaws.com/AWSServiceRoleForImageBuilder"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetInstanceProfile"
            ],
            "Resource": "arn:aws:iam::*:instance-profile/*imagebuilder*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:ListInstanceProfiles",
                "iam:ListRoles"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::*:instance-profile/*imagebuilder*",
                "arn:aws:iam::*:role/*imagebuilder*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ec2.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "s3:ListAllMyBuckets",
                "s3:GetBucketLocation"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3::*:*imagebuilder*"
        },
        {
            "Action": "iam:CreateServiceLinkedRole",
            "Effect": "Allow",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
imagebuilder.amazonaws.com/AWSServiceRoleForImageBuilder",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "imagebuilder.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeImages",
                "ec2:DescribeSnapshots",
                "ec2:DescribeVpcs",
                "ec2:DescribeRegions",
                "ec2:DescribeVolumes",
                "ec2:DescribeSubnets",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeInstanceTypeOfferings",
                "ec2:DescribeLaunchTemplates"
            ],
            "Resource": "*"
        }
    ]
}
```

Política do 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 de API `iam:GetRole`.

Detalhes da permissão

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`.

Exemplo de política

A seguir há um exemplo da política do AWSImageBuilderReadOnlyAccess.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "imagebuilder:Get*",
                "imagebuilder:List*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
imagebuilder.amazonaws.com/AWSServiceRoleForImageBuilder"
        }
    ]
}
```

Política do AWSServiceRoleForImageBuilder

A AWSServiceRoleForImageBuilderpolítica permite que o Image Builder ligue Serviços da AWS em seu nome.

Detalhes da permissão

Esta política é anexada à função vinculada ao serviço Image Builder quando a função é criada por meio do Systems Manager. Para analisar as permissões específicas concedidas, consulte o exemplo de política nesta seção. Para obter mais informações sobre a função vinculada ao serviço do Image Builder, consulte Usar perfis vinculados ao serviço para o EC2 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 imagens e iniciar instâncias do EC2 em sua conta, usando snapshots, volumes, interfaces de rede, sub-redes, grupos de segurança, configuração de licença 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 `CreatedBy: EC2 Image Builder` ou `CreatedBy: EC2 Fast Launch`.

  O Image Builder pode obter informações sobre imagens do Amazon EC2, atributos da instância, status da instância, os tipos de instância que estão disponíveis para sua conta, modelos de inicialização, sub-redes, hosts e tags em seus recursos do 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 o Amazon EC2 e para o 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 suas invocações, entradas de inventário, descrever informações de instância e status de execução de automação, além de obter detalhes de inovação do comando. 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`, e pode criar a função vinculada ao serviço Systems Manager em sua conta.
- 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.

Exemplo de política

A seguir há um exemplo da política do AWSServiceRoleForImageBuilder.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:*::image/*",
                "arn:aws:ec2:*::snapshot/*",
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:network-interface/*",
                "arn:aws:ec2:*:*:security-group/*",
                "arn:aws:ec2:*:*:key-pair/*",
                "arn:aws:ec2:*:*:launch-template/*",
                "arn:aws:license-manager:*:*:license-configuration:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:volume/*",
                "arn:aws:ec2:*:*:instance/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/CreatedBy": [
                        "EC2 Image Builder",
                        "EC2 Fast Launch"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
```

```
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com",
                        "ec2.amazonaws.com.cn",
                        "vmie.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StopInstances",
                "ec2:StartInstances",
                "ec2:TerminateInstances"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/CreatedBy": "EC2 Image Builder"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CopyImage",
                "ec2:CreateImage",
                "ec2:CreateLaunchTemplate",
                "ec2:DeregisterImage",
                "ec2:DescribeImages",
                "ec2:DescribeInstanceAttribute",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceTypeOfferings",
                "ec2:DescribeInstanceTypes",
                "ec2:DescribeSubnets",
                "ec2:DescribeTags",
                "ec2:ModifyImageAttribute",
                "ec2:DescribeImportImageTasks",
                "ec2:DescribeExportImageTasks",
                "ec2:DescribeSnapshots",
```

```
                "ec2:DescribeHosts"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:ModifySnapshotAttribute"
            ],
            "Resource": "arn:aws:ec2:*::snapshot/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/CreatedBy": "EC2 Image Builder"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": [
                        "RunInstances",
                        "CreateImage"
                    ],
                    "aws:RequestTag/CreatedBy": [
                        "EC2 Image Builder",
                        "EC2 Fast Launch"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*::image/*",
                "arn:aws:ec2:*:*:export-image-task/*"
            ]
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*::snapshot/*",
                "arn:aws:ec2:*:*:launch-template/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/CreatedBy": [
                        "EC2 Image Builder",
                        "EC2 Fast Launch"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "license-manager:UpdateLicenseSpecificationsForResource"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:Publish"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:ListCommands",
                "ssm:ListCommandInvocations",
                "ssm:AddTagsToResource",
                "ssm:DescribeInstanceInformation",
                "ssm:GetAutomationExecution",
                "ssm:StopAutomationExecution",
                "ssm:ListInventoryEntries",
                "ssm:SendAutomationSignal",
```

```
                "ssm:DescribeInstanceAssociationsStatus",
                "ssm:DescribeAssociationExecutions",
                "ssm:GetCommandInvocation"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ssm:SendCommand",
            "Resource": [
                "arn:aws:ssm:*:*:document/AWS-RunPowerShellScript",
                "arn:aws:ssm:*:*:document/AWS-RunShellScript",
                "arn:aws:ssm:*:*:document/AWSEC2-RunSysprep",
                "arn:aws:s3:::*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:instance/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ssm:resourceTag/CreatedBy": [
                        "EC2 Image Builder"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ssm:StartAutomationExecution",
            "Resource": "arn:aws:ssm:*:*:automation-definition/ImageBuilder*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:CreateAssociation",
                "ssm:DeleteAssociation"
            ],
            "Resource": [
```

```
                "arn:aws:ssm:*:*:document/AWS-GatherSoftwareInventory",
                "arn:aws:ssm:*:*:association/*",
                "arn:aws:ec2:*:*:instance/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncryptFrom",
                "kms:ReEncryptTo",
                "kms:GenerateDataKeyWithoutPlaintext"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "kms:EncryptionContextKeys": [
                        "aws:ebs:id"
                    ]
                },
                "StringLike": {
                    "kms:ViaService": [
                        "ec2.*.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:DescribeKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "ec2.*.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
```

```
            "Action": "kms:CreateGrant",
            "Resource": "*",
            "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": true
                },
                "StringLike": {
                    "kms:ViaService": [
                        "ec2.*.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "arn:aws:iam::*:role/
EC2ImageBuilderDistributionCrossAccountRole"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:CreateLogGroup",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/imagebuilder/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateLaunchTemplateVersion",
                "ec2:DescribeLaunchTemplates",
                "ec2:ModifyLaunchTemplate",
                "ec2:DescribeLaunchTemplateVersions"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:ExportImage"
            ],
            "Resource": "arn:aws:ec2:*::image/*",
```

```
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/CreatedBy": "EC2 Image Builder"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:ExportImage"
            ],
            "Resource": "arn:aws:ec2:*:*:export-image-task/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CancelExportTask"
            ],
            "Resource": "arn:aws:ec2:*:*:export-image-task/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/CreatedBy": "EC2 Image Builder"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "ssm.amazonaws.com",
                        "ec2fastlaunch.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:EnableFastLaunch"
            ],
            "Resource": [
```

```
                "arn:aws:ec2:*::image/*",
                "arn:aws:ec2:*:*:launch-template/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/CreatedBy": "EC2 Image Builder"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "inspector2:ListCoverage",
                "inspector2:ListFindings"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecr:CreateRepository"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/CreatedBy": "EC2 Image Builder"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecr:TagResource"
            ],
            "Resource": "arn:aws:ecr:*:*:repository/image-builder-*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/CreatedBy": "EC2 Image Builder"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "ecr:BatchDeleteImage"
            ],
            "Resource": "arn:aws:ecr:*:*:repository/image-builder-*",
            "Condition": {
                "StringEquals": {
                    "ecr:ResourceTag/CreatedBy": "EC2 Image Builder"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "events:DeleteRule",
                "events:DescribeRule",
                "events:PutRule",
                "events:PutTargets",
                "events:RemoveTargets"
            ],
            "Resource": [
                "arn:aws:events:*:*:rule/ImageBuilder-*"
            ]
        }
    ]
}
```

Política do 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 imagem do Amazon EC2 na conta. A política também concede a capacidade de modificar as permissões da AMI por meio da ação da API `ec2:ModifyImageAttribute`.

Detalhes da permissão

Esta política inclui as seguintes permissões:

- Amazon EC2 — O acesso é concedido ao Amazon EC2 para descrever, copiar e modificar atributos de uma imagem e criar tags para qualquer imagem do Amazon EC2 na conta.

Exemplo de política

A seguir há um exemplo da política do Ec2ImageBuilderCrossAccountDistributionAccess.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:*::image/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeImages",
                "ec2:CopyImage",
                "ec2:ModifyImageAttribute"
            ],
            "Resource": "*"
        }
    ]
}
```

Política do EC2ImageBuilderLifecycleExecutionPolicy

A política EC2ImageBuilderLifecycleExecutionPolicy concede permissões para que o Image Builder execute ações como descontinuar, desabilitar ou excluir recursos de imagem do Image Builder e seus recursos subjacentes (AMIs, instantâneos) a fim de viabilizar regras automatizadas para tarefas de gerenciamento do ciclo de vida de imagem.

Detalhes da permissão

Esta política inclui as seguintes permissões:

- Amazon EC2: o acesso é concedido ao Amazon EC2 para realizar as seguintes ações para imagens de máquina da Amazon (AMI) na conta 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.

Exemplo de política

A seguir há um exemplo da política do EC2ImageBuilderLifecycleExecutionPolicy.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Ec2ImagePermission",
            "Effect": "Allow",
            "Action": [
                "ec2:EnableImage",
                "ec2:DeregisterImage",
                "ec2:EnableImageDeprecation",
                "ec2:DescribeImageAttribute",
                "ec2:DisableImage",
                "ec2:DisableImageDeprecation"
            ],
            "Resource": "arn:aws:ec2:*::image/*",
            "Condition": {
```

```
                "StringEquals": {
                    "aws:ResourceTag/CreatedBy": "EC2 Image Builder"
                }
            }
        },
        {
            "Sid": "EC2DeleteSnapshotPermission",
            "Effect": "Allow",
            "Action": "ec2:DeleteSnapshot",
            "Resource": "arn:aws:ec2:*::snapshot/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/CreatedBy": "EC2 Image Builder"
                }
            }
        },
        {
            "Sid": "EC2TagsPermission",
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteTags",
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*::snapshot/*",
                "arn:aws:ec2:*::image/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/DeprecatedBy": "EC2 Image Builder",
                    "aws:ResourceTag/CreatedBy": "EC2 Image Builder"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "DeprecatedBy"
                }
            }
        },
        {
            "Sid": "ECRImagePermission",
            "Effect": "Allow",
            "Action": [
                "ecr:BatchGetImage",
                "ecr:BatchDeleteImage"
            ],
```

```
            "Resource": "arn:aws:ecr:*:*:repository/*",
            "Condition": {
                "StringEquals": {
                    "ecr:ResourceTag/LifecycleExecutionAccess": "EC2 Image Builder"
                }
            }
        },
        {
            "Sid": "ImageBuilderEC2TagServicePermission",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeImages",
                "tag:GetResources",
                "imagebuilder:DeleteImage"
            ],
            "Resource": "*"
        }
    ]
}
```

Política do EC2InstanceProfileForImageBuilder

A política do EC2InstanceProfileForImageBuilder concede as permissões mínimas necessárias para que uma instância do EC2 funcione com o Image Builder. Isso não inclui as permissões necessárias para usar o Systems Manager Agent.

Detalhes da permissão

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/`.
- Image Builder — O acesso é concedido para obter qualquer componente do Image Builder.
- 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 começa com `ec2imagebuilder-`.

Exemplo de política

A seguir há um exemplo da política do EC2InstanceProfileForImageBuilder.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "imagebuilder:GetComponent"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "kms:EncryptionContextKeys": "aws:imagebuilder:arn",
                    "aws:CalledVia": [
                        "imagebuilder.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": "arn:aws:s3:::ec2imagebuilder*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:CreateLogGroup",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/imagebuilder/*"
        }
    ]
}
```

Política do EC2InstanceProfileForImageBuilderECRContainerBuilds

A política do EC2InstanceProfileForImageBuilderECRContainerBuilds concede as permissões mínimas necessárias para uma instância do EC2 ao trabalhar com o Image Builder para criar imagens do Docker e, em seguida, registrar e armazenar as imagens em um repositório de contêiner do Amazon ECR. Isso não inclui as permissões necessárias para usar o Systems Manager Agent.

Detalhes da permissão

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-`.

Exemplo de política

A seguir há um exemplo da política do EC2InstanceProfileForImageBuilderECRContainerBuilds.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "imagebuilder:GetComponent",
                "imagebuilder:GetContainerRecipe",
                "ecr:GetAuthorizationToken",
                "ecr:BatchGetImage",
                "ecr:InitiateLayerUpload",
                "ecr:UploadLayerPart",
                "ecr:CompleteLayerUpload",
                "ecr:BatchCheckLayerAvailability",
```

```
                "ecr:GetDownloadUrlForLayer",
                "ecr:PutImage"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "kms:EncryptionContextKeys": "aws:imagebuilder:arn",
                    "aws:CalledVia": [
                        "imagebuilder.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": "arn:aws:s3:::ec2imagebuilder*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:CreateLogGroup",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/imagebuilder/*"
        }
    ]
}
```

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
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 seguintes alterações no perfil de serviço para oferecer compatibilidade com macOS. • 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 agosto de 2023
AWSServiceRoleForImageBuilder: atualização para uma política existente	O Image Builder fez as seguintes alterações no perfil de serviço para permitir que	30 de março de 2023

Alteração	Descrição	Data
	os fluxos de trabalho do Image Builder coletassem descobertas de vulnerabi lidade para compilações de imagens de contêiner AMI e ECR. As novas permissões são compatíveis ao 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 instância s de teste e colete resultado s para imagens que estão configuradas para permitir isso. • Foi adicionado ecr:Creat eRepository, com a exigência de que o Image Builder marque o repositór io com `CreatedBy: EC2 Image Builder` ()tag-on-create. Também foi adicionado ecr: TagResour ce (obrigatório para tag-on-create) com a mesma restrição de CreatedBy tag e uma restrição adicional que exige que o nome do	

Alteração	Descrição	Data
	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 AMIs de imagem básica 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: • Permissões adicionadas para a ação EnableFastLaunch da API EC2, para ativar e desativar o lançamento mais rápido de instâncias do Windows. • Escopo mais estreito 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 reflita com precisão os tipos de instância que estão 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 vinculados ao serviço para o EC2 Image Builder

O EC2 Image Builder AWS Identity and Access Management usa funções vinculadas a serviços (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 a 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 EC2 Image Builder, o Image Builder cria um perfil vinculado ao serviço para você novamente.

Para ver as permissões do AWSServiceRoleForImageBuilder, consulte a página Política do 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 Exclua recursos do EC2 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 EC2 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.

Solução de problemas de identidade e acesso do EC2 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 atributo *`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

Você pode criar uma função que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. Você pode 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 EC2 Image Builder funciona com o 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 usar perfis e políticas baseadas em recursos para acesso entre contas, consulte Como os perfis do IAM diferem de políticas baseadas em recursos no Guia do usuário do IAM.

Validação da conformidade do EC2 Image Builder

O EC2 Image Builder não está no escopo de AWS nenhum programa de 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: estes 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— 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 ter mais informações, consulte Produtos de conformidade para suas imagens do Image Builder.

Resiliência no EC2 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 mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de datacenter 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 instâncias do EC2 que o Image Builder usa para criar e testar imagens devem ter acesso AWS Systems Manager.

Gerenciamento de patches no EC2 Image Builder

O EC2 Image Builder fornece as mais recentes AMIs do Amazon Linux 2, Amazon Linux 2023, Red Hat Enterprise Linux (RHEL), CentOS, Ubuntu, SUSE Linux Enterprise Server e Windows 2012 R2 e versões posteriores como fontes de imagem gerenciadas. Você mantém a responsabilidade de aplicação de patches do sistema Amazon EC2, de acordo com o modelo de responsabilidade compartilhada. Se for possível substituir facilmente as instâncias do EC2 na workload da sua aplicação, talvez seja mais eficiente atualizar a AMI básica e reimplantar todos os nós de computação com base nessa imagem.

Veja a seguir duas maneiras de manter suas AMIs do Image Builder atualizadas.

- AWS-componentes de correção fornecidos — O EC2 Image Builder fornece dois componentes de compilação `update-linux` e `update-windows`, que instalam todas as atualizações pendentes do sistema operacional. Esses componentes usam o módulo de `UpdateOS` ação. Para ter 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 de AMIs compatíveis, 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 ter mais informações, consulte Módulos de ação suportados pelo gerenciador de componentes do AWSTOE.

  Componente que usa o módulo de `UpdateOS` ação (Linux e Windows)

  ```
  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 do EC2 Image Builder

O EC2 Image Builder oferece uma série de recursos de segurança a serem considerados no desenvolvimento e na implementação das suas próprias políticas de segurança. As melhores práticas a seguir são diretrizes gerais e não representam uma solução completa de segurança. Como essas melhores práticas 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.

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 seu pipeline do Image Builder é executado, ele envia solicitações HTTP para iniciar instâncias do EC2 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 instâncias do EC2 que o Image Builder executa a partir de um pipeline criado para usar o IMDSv2, de 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, consulteGerencie a configuração da infraestrutura do EC2 Image Builder. Para obter mais informações sobre opções de metadados de instância do EC2 para imagens do Linux, consulte Configurar as opções de metadados da instância no Guia do usuário do Amazon EC2 para instâncias do Linux. Para ter mais informações, consulte Configurar as opções de metadados da instância no Guia do usuário do Amazon EC2 para instâncias Windows.

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.

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.

Entrada

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 EC2 Image Builder

O EC2 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

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 GetWorkflowExecutione ListWorkflowStepExecutionsda 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:

  LogGroup:

  ```
  /aws/imagebuilder/ImageName
  ```

  LogStream (x.x.x/x):

  ```
  ImageVersion/ImageBuildVersion
  ```

  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 instância do Amazon EC2 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 logs que você envia para seu bucket do S3 mostram as etapas e as mensagens de erro da atividade na instância do EC2 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 compilação ou teste do Amazon EC2 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 de instância não tem as permissões necessárias para acessar APIs ou recursos de componentes.
- 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 Conceitos de endpoints de VPC ().AWS PrivateLink
- Permissões ausentes — Adicione as seguintes políticas gerenciadas à sua função vinculada ao serviço do IAM para o Image Builder:
  - EC2 InstanceProfileForImageBuilder
  - EC2 ECR InstanceProfileForImageBuilder ContainerBuilds
  - Amazon SMS ManagedInstanceCore

  Para obter mais informações sobre a função vinculada ao serviço do Image Builder, consulte Usar perfis vinculados ao serviço para o EC2 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 no 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 usuário do Amazon EC2 para instâncias do Windows.

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 instâncias do EC2. 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 imagem da AMI usando o console do EC2 Image Builder, consulte Atualizar pipelines de imagem da AMI (console). Para obter mais informações sobre como atualizar seu pipeline de imagem de contêiner usando o console do EC2 Image Builder, consulte Atualizar um pipeline de imagem de contêiner (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 propriedade `EnhancedImageMetadataEnabled` em seu arquivo JSON como `false` quando você cria seu pipeline usando a AWS CLI.

Histórico do documento para o Guia do usuário do EC2 Image Builder

A tabela a seguir descreve alterações importantes feitas na documentação por data. Para receber notificações sobre atualizações dessa documentação, você poderá se inscrever em um feed RSS.

- Versão da API: 2023-12-12

Alteração	Descrição	Data
Atualizações do STIG Q1	Versões atualizadas do Linux STIG e STIGS aplicado 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 Q4	Versões atualizadas do Linux STIG e aplicou o STIGS para a versão do quarto trimestre de 2023. Não houve alteraçõe s nas versões do Windows. Também atualizou o Linux e o Windows SCAP para novos	7 de dezembro de 2023

	componentes, softwares e números de benchmark.	
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
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 STIGs ignorados 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. Adicionado suporte para o AL2023.	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.	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	Adicionada documentação para o módulo de ação `ExecuteDocument` sob `General execution` .	28 de março de 2022

Release do atributo: oferecer suporte à inicialização mais rápida da AMI do Windows	Adicionadas definições de configuração de distribui ção para oferecer suporte à inicialização mais rápida de AMIs do Windows.	21 de fevereiro de 2022
Versão de manutençã o: 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 configura ção JSON como entrada para o AWSTOE run comando.	3 de fevereiro de 2022
Novas versões do STIG	Versões do STIG atualizad as e STIGS aplicados para o release do quarto trimestre de 2021. Também foi adicionad a uma seção para os novos componentes do SCAP Compliance Checker (SCC).	22 de dezembro de 2021
Release do atributo: integraçã o de VM Import/Export (VMIE)	Adicionado suporte para importação de VM por meio de todos os canais (console, API/ CLI etc.) e para exportação de VM via API/CLI. Atualment e, a exportação de VM não está disponível no console do Image Builder.	20 de dezembro de 2021
Lançamento de recursos: compartilhamento de AMI para AWS Organizations e OUs	Configuração de distribuição atualizada para adicionar suporte ao compartilhamento de AMIs de saída com AWS Organizations OUs.	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 instâncias do EC2 que são usadas para compilar e testar imagens, incluindo a capacidade de especificar comandos a serem executados na inicializ ação, e mais controle sobre a instalação e remoção do atendente do Systems Manager.	7 de julho de 2021
Novas versões do STIG	Versões do STIG atualizad as e STIGS aplicados para o release do segundo trimestre de 2021.	30 de junho de 2021
Aprimoramento: aprimoram entos de marcação	Mensagens aperfeiçoadas sobre marcação de recursos.	25 de junho de 2021
Release do atributo: inicializar a integração do modelo	Adicionado suporte para o uso de modelos de inicialização do Amazon EC2 para distribuição de AMI nas configurações de distribuição.	7 de abril de 2021
Release do atributo: aprimoramentos da compilaçã o de contêineres	Adicionado suporte para configurar mapeamentos de dispositivos de blocos e especificar AMIs a serem usadas como imagem de base para compilações de contêiner es.	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 gerenciam ento, 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 desenvolv imento local de AWSTOE componentes	Desenvolva e teste component es de imagem localmente com o AWSTOE aplicativo.	28 de julho de 2020

AMIs criptografadas	O EC2 Image Builder adiciona suporte para distribuição de AMI criptografada.	1º 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	É possível estabelecer uma conexão privada entre sua VPC e o EC2 Image Builder criando um endpoint da VPC. Os endpoints de interface são alimentados por AWS PrivateLink uma tecnologia que permite acessar de forma privada as APIs do Image Builder sem um gateway de internet, dispositivo NAT, conexão VPN ou conexão Direct AWS Connect. As instâncias em sua VPC não precisam de endereços IP públicos para a comunicação com APIs do Image Builder. 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.