aws

Guide de l’utilisateur

EC2 Image Builder

EC2 Image Builder: Guide de l’utilisateur

Table of Contents

Qu'est-ce qu'EC2 Image Builder ?

EC2 Image Builder est une solution entièrement Service AWS gérée qui vous permet d'automatiser la création, la gestion et le déploiement d'images personnalisées, sécurisées up-to-date et de serveur. Vous pouvez utiliser les API AWS Management Console AWS Command Line Interface, ou pour créer des images personnalisées dans votre Compte AWS.

Vous êtes propriétaire des images personnalisées créées par Image Builder dans votre compte. Vous pouvez configurer des pipelines pour automatiser les mises à jour et l'application de correctifs au système pour les images que vous possédez. Vous pouvez également exécuter une commande autonome pour créer une image avec les ressources de configuration que vous avez définies.

L'assistant de pipeline Image Builder peut vous guider tout au long des étapes de création d'une image personnalisée, comme suit :

1. Choisissez une image de base pour vos personnalisations.
2. Ajoutez ou supprimez des logiciels de votre image de base.
3. Personnalisez les paramètres et les scripts avec des composants de build.
4. Exécutez des tests sélectionnés ou créez des composants de test personnalisés.
5. Distribuez des AMI à Régions AWS et Comptes AWS.
6. Si votre pipeline Image Builder crée une Amazon Machine Image (AMI) personnalisée à distribuer, vous pouvez autoriser d'autres Comptes AWS organisations et unités d'organisation à la lancer depuis votre compte. Votre compte est facturé pour les frais associés à l'AMI.

Image Builder s'intègre aux éléments suivants Services AWS pour fournir des mesures détaillées sur les événements, la journalisation et la surveillance. Ces informations vous aident à suivre votre activité, à résoudre les problèmes liés à la création d'images et à créer des automatisations basées sur les notifications d'événements.

Contenu de la section

- Tarification
- Connexe Services AWS

Caractéristiques d'EC2 Image Builder

EC2 Image Builder fournit les fonctionnalités suivantes :

Augmentez la productivité et réduisez les opérations liées à la mise en conformité et à la création up-to-date d'images

Image Builder réduit la quantité de travail nécessaire à la création et à la gestion d'images à grande échelle en automatisant vos pipelines de création. Vous pouvez automatiser vos builds en indiquant vos préférences en matière de calendrier d'exécution des builds. L'automatisation réduit les coûts opérationnels liés à la maintenance de vos logiciels avec les derniers correctifs du système d'exploitation.

Augmenter le temps de disponibilité des services

Image Builder donne accès à des composants de test que vous pouvez utiliser pour tester vos images avant le déploiement. Vous pouvez également créer des composants de test personnalisés avec AWS Task Orchestrator and Executor (AWSTOE) et les utiliser. Image Builder distribue votre image uniquement si tous les tests configurés ont réussi.

Rehaussez la barre de sécurité pour les déploiements

Image Builder vous permet de créer des images qui éliminent toute exposition inutile aux vulnérabilités de sécurité des composants. Vous pouvez appliquer des paramètres de AWS sécurité pour créer des out-of-the-box images sécurisées répondant aux critères de sécurité internes et industriels. Image Builder fournit également des ensembles de paramètres pour les entreprises des secteurs réglementés. Vous pouvez utiliser ces paramètres pour créer rapidement et facilement des images conformes aux normes STIG. Pour une liste complète des composants STIG disponibles via Image Builder, consultezAmazon a géré les composants de renforcement STIG pour EC2 Image Builder.

Application centralisée et suivi du lignage

Grâce aux intégrations intégrées à Image Builder AWS Organizations, vous pouvez appliquer des politiques qui empêchent les comptes d'exécuter des instances uniquement à partir d'AMI approuvées.

Partage simplifié des ressources entre Comptes AWS

EC2 Image Builder s' AWS Resource Access Manager intègre àAWS RAM() pour vous permettre de partager certaines ressources avec Compte AWS n'importe qui ou AWS Organizations via. Les ressources EC2 Image Builder qui peuvent être partagées sont les suivantes :

- Composants
- Images
- Recettes d'images
- Recettes de contenants

Pour plus d'informations, consultez Partagez les ressources d'EC2 Image Builder.

Systèmes d'exploitation pris en charge

Image Builder prend en charge les versions de système d'exploitation suivantes :

Système d'exploitation/distribution	Versions prises en charge
Amazon Linux	2 et 2023
CentOS	7 et 8
CentOS Stream	8
Red Hat Enterprise Linux (RHEL)	7 et 8
Serveur SUSE Linux Enterprise (SUSE)	12 et 15
Ubuntu	18,04 LTS, 20,04 LTS et 22,04 LTS
Windows Server	2012 R2, 2016, 2019 et 2022

Formats d'image pris en charge

Pour vos images AMI personnalisées, vous pouvez choisir une AMI existante comme point de départ. Pour les images de conteneur Docker, vous pouvez choisir entre des images publiques hébergées

sur DockerHub, des images de conteneur existantes dans Amazon ECR ou des images de conteneur gérées par Amazon.

Concepts

Les termes et concepts suivants sont essentiels à votre compréhension et à votre utilisation d'EC2 Image Builder.

AMI

L'Amazon Machine Image (AMI) est l'unité de base du déploiement dans Amazon EC2 et constitue l'un des types d'images que vous pouvez créer avec Image Builder. Une AMI est une image de machine virtuelle préconfigurée qui contient le système d'exploitation (OS) et le logiciel préinstallé pour déployer les instances EC2. Pour plus d'informations, consultez Amazon Machine Images (AMI).

Pipeline d'images

Un pipeline d'images fournit un cadre d'automatisation permettant de créer des AMI sécurisées et des images de conteneur AWS. Le pipeline d'images Image Builder est associé à une recette d'image ou à une recette de conteneur qui définit les phases de création, de validation et de test du cycle de vie d'une création d'image.

Un pipeline d'image peut être associé à une configuration d'infrastructure qui définit l'emplacement de création de votre image. Vous pouvez définir des attributs, tels que le type d'instance, les sous-réseaux, les groupes de sécurité, la journalisation et d'autres configurations liées à l'infrastructure. Vous pouvez également associer votre pipeline d'images à une configuration de distribution pour définir la manière dont vous souhaitez déployer votre image.

Image gérée

Une image gérée est une ressource d'Image Builder composée d'une AMI ou d'une image de conteneur, ainsi que de métadonnées, telles que la version et la plate-forme. L'image gérée est utilisée par les pipelines Image Builder pour déterminer l'image de base à utiliser pour la génération. Dans ce guide, les images gérées sont parfois appelées « images », mais une image n'est pas la même chose qu'une AMI.

Recette d'images

Une recette d'image Image Builder est un document qui définit l'image de base et les composants qui sont appliqués à l'image de base afin de produire la configuration souhaitée pour l'image AMI

de sortie. Vous pouvez utiliser une recette d'image pour dupliquer les builds. Les recettes d'images Image Builder peuvent être partagées, ramifiées et modifiées à l'aide de l'assistant de console, de l' AWS CLI API ou de l'API. Vous pouvez utiliser des recettes d'images avec votre logiciel de contrôle de version pour conserver des recettes d'images versionnées partageables.

Recette de contenant

Une recette de conteneur Image Builder est un document qui définit l'image de base et les composants qui sont appliqués à l'image de base afin de produire la configuration souhaitée pour l'image de conteneur de sortie. Vous pouvez utiliser une recette de conteneur pour dupliquer des builds. Vous pouvez partager, associer et modifier des recettes d'images Image Builder à l'aide de l'assistant de console, de l' AWS CLI API ou de l'API. Vous pouvez utiliser des recettes de conteneurs avec votre logiciel de contrôle de version pour gérer des recettes de conteneurs versionnées et partageables.

Image de base

L'image de base est l'image sélectionnée et le système d'exploitation utilisés dans votre image ou votre document de recette de conteneur, ainsi que les composants. L'image de base et les définitions de composants combinées produisent la configuration souhaitée pour l'image de sortie.

Composants

Un composant définit la séquence d'étapes requises pour personnaliser une instance avant la création de l'image (composant de génération) ou pour tester une instance lancée à partir de l'image créée (composant de test).

Un composant est créé à partir d'un document YAML ou JSON déclaratif en texte brut qui décrit la configuration d'exécution pour créer et valider, ou tester une instance produite par votre pipeline. Les composants s'exécutent sur l'instance à l'aide d'une application de gestion des composants. L'application de gestion des composants analyse les documents et exécute les étapes souhaitées.

Après leur création, un ou plusieurs composants sont regroupés à l'aide d'une recette d'image ou d'une recette de conteneur afin de définir le plan de création et de test d'une machine virtuelle ou d'une image de conteneur. Vous pouvez utiliser des composants publics détenus et gérés par AWS, ou vous pouvez créer les vôtres. Pour plus d'informations sur les composants, consultezAWS Task Orchestrator and Executor gestionnaire de composants.

Document sur les composants

Document YAML ou JSON déclaratif en texte brut qui décrit la configuration d'une personnalisation que vous pouvez appliquer à votre image. Le document est utilisé pour créer un composant de construction ou de test.

Étapes d'exécution

EC2 Image Builder comporte deux étapes d'exécution : construction et test. Chaque phase d'exécution comporte une ou plusieurs phases dont la configuration est définie par le document du composant.

Phases de configuration

La liste suivante indique les phases exécutées pendant les phases de construction et de test :

Étape de construction :

Phase de génération

Un pipeline d'images commence par la phase de génération de l'étape de génération lorsqu'il s'exécute. L'image de base est téléchargée et la configuration spécifiée pour la phase de génération du composant est appliquée pour créer et lancer une instance.

Phase de validation

Une fois qu'Image Builder a lancé l'instance et appliqué toutes les personnalisations de la phase de création, la phase de validation commence. Au cours de cette phase, Image Builder s'assure que toutes les personnalisations fonctionnent comme prévu, en fonction de la configuration spécifiée par le composant pour la phase de validation. Si la validation de l'instance aboutit, Image Builder arrête l'instance, crée une image, puis passe à la phase de test.

Étape de test :

Phase de test

Au cours de cette phase, Image Builder lance une instance à partir de l'image qu'il a créée une fois la phase de validation terminée avec succès. Image Builder exécute des composants de test au cours de cette phase pour vérifier que l'instance est saine et fonctionne comme prévu.

Phase de test de l'hôte du conteneur

Une fois qu'Image Builder a exécuté la phase de test pour tous les composants que vous avez sélectionnés dans la recette du conteneur, Image Builder exécute cette phase pour

les flux de travail du conteneur. La phase de test de l'hôte du conteneur peut exécuter des tests supplémentaires qui valident la gestion du conteneur et les configurations d'exécution personnalisées.

Flux de travail

Les flux de travail définissent la séquence d'étapes qu'Image Builder exécute lorsqu'il crée une nouvelle image. Toutes les images ont des flux de travail de création et de test. Les conteneurs disposent d'un flux de travail supplémentaire pour la distribution.

Types de flux de travail

`BUILD`

Couvre la configuration de la phase de construction pour chaque image créée.

`TEST`

Couvre la configuration de la phase de test pour chaque image créée.

`DISTRIBUTION`

Couvre le flux de travail de distribution des images de conteneurs.

Tarification

L'utilisation d'EC2 Image Builder pour créer des images d'AMI ou de conteneur personnalisées est gratuite. Cependant, la tarification standard s'applique aux autres services utilisés dans le processus. La liste suivante inclut l'utilisation de certaines images Services AWS qui peuvent entraîner des coûts lors de la création, du développement, du stockage et de la distribution de votre AMI ou de vos images de conteneur personnalisées, en fonction de votre configuration.

- Lancement d'une instance EC2
- Stockage des journaux sur Amazon S3
- Validation d'images avec Amazon Inspector
- Stockage des instantanés Amazon EBS pour vos AMI
- Stockage d'images de conteneurs dans Amazon ECR
- Transférer et extraire des images de conteneurs vers et depuis Amazon ECR

- Si Systems Manager Advanced Tier est activé et que les instances Amazon EC2 s'exécutent avec une activation sur site, des ressources peuvent vous être facturées via Systems Manager

Connexe Services AWS

EC2 Image Builder utilise d' Services AWS autres outils pour créer des images. Selon la configuration de votre recette d'image ou de recette de conteneur Image Builder, les services suivants peuvent être utilisés.

AWS License Manager

AWS License Manager vous permet de créer et d'appliquer des configurations de licence à partir d'un magasin de configuration de licences de compte. Pour chaque AMI, vous pouvez utiliser Image Builder pour l'associer à une configuration de licence préexistante à laquelle vous Compte AWS avez accès dans le cadre du flux de travail Image Builder. Les configurations de licence ne peuvent être appliquées qu'aux AMI. Image Builder ne peut utiliser que des configurations de licence préexistantes et ne peut pas créer ou modifier directement des configurations de licence. Les paramètres du License Manager ne seront pas répliqués entre ceux Régions AWS qui doivent être activés dans votre compte, par exemple entre les régions `ap-east-1` (Asie-Pacifique : Hong Kong) et `me-south-1` (Moyen-Orient : Bahreïn).

AWS Organizations

AWS Organizations vous permet d'appliquer des politiques de contrôle des services (SCP) aux comptes de votre organisation. Vous pouvez créer, gérer, activer et désactiver des politiques individuelles. Comme tous les autres AWS artefacts et services, Image Builder respecte les politiques définies dans AWS Organizations. AWS fournit des modèles de SCP pour les scénarios courants, tels que l'imposition de contraintes aux comptes membres afin de lancer des instances avec uniquement des AMI approuvées.

Amazon Inspector

Image Builder utilise Amazon Inspector comme agent d'analyse des vulnérabilités par défaut pour établir des bases de sécurité pour Amazon Linux 2, Windows Server 2012 et Windows Server 2016. Pour plus d'informations, consultez Qu'est-ce qu'Amazon Inspector ?

AWS Resource Access Manager

AWS Resource Access Manager (AWS RAM) vous permet de partager vos ressources avec n'importe qui Compte AWS ou via AWS Organizations. Si vous en avez plusieurs Comptes AWS,

vous pouvez créer des ressources de manière centralisée et les AWS RAM partager avec d'autres comptes. EC2 Image Builder permet de partager les ressources suivantes : composants, images et recettes d'images. Pour plus d'informations AWS RAM, consultez le guide de AWS Resource Access Manager l'utilisateur. Pour plus d'informations sur le partage des ressources Image Builder, consultezPartagez les ressources d'EC2 Image Builder.

Amazon CloudWatch Logs

Vous pouvez utiliser Amazon CloudWatch Logs pour surveiller, stocker et accéder à vos fichiers journaux à partir d'instances EC2 AWS CloudTrail, d'Amazon Route 53 et d'autres sources.

Amazon Elastic Container Registry (Amazon ECR)

Amazon ECR est un service de registre d'images de AWS conteneurs géré qui est sécurisé, évolutif et fiable. Les images de conteneur que vous créez avec Image Builder sont stockées dans Amazon ECR dans votre région source (où s'exécute votre build) et dans toutes les régions où vous distribuez l'image de conteneur. Pour plus d'informations sur Amazon ECR, consultez le guide de l'utilisateur d'Amazon Elastic Container Registry.

Comment fonctionne EC2 Image Builder

Lorsque vous utilisez l'assistant de la console de pipeline EC2 Image Builder pour créer une image personnalisée, un assistant vous guide tout au long des étapes suivantes.

1. Spécifiez les détails du pipeline : entrez les informations relatives à votre pipeline, telles que le nom, la description, les balises et le calendrier d'exécution des builds automatisés. Vous pouvez choisir des versions manuelles, si vous préférez.
2. Choisir une recette : choisissez entre créer une AMI ou créer une image de conteneur. Pour les deux types d'images de sortie, vous entrez le nom et la version de votre recette, sélectionnez une image de base et choisissez les composants à ajouter pour la création et les tests. Vous pouvez également choisir la gestion automatique des versions, afin de toujours utiliser la dernière version du système d'exploitation (OS) disponible pour votre image de base. Les recettes de conteneur définissent également Dockerfiles et le référentiel Amazon ECR cible pour votre image de conteneur Docker en sortie.

   > **Note**
   >
   > Les composants sont les éléments de base utilisés par une recette d'image ou une recette de conteneur. Par exemple, des packages pour l'installation, des étapes de renforcement de la sécurité et des tests. L'image de base et les composants sélectionnés constituent une recette d'image.

3. Définissez la configuration de l'infrastructure : Image Builder lance des instances EC2 dans votre compte pour personnaliser les images et exécuter des tests de validation. Les paramètres de configuration de l'infrastructure spécifient les détails de l'infrastructure pour les instances qui s'exécuteront dans votre Compte AWS système pendant le processus de génération.
4. Définissez les paramètres de distribution : choisissez les AWS régions dans lesquelles distribuer votre image une fois la génération terminée et tous les tests réussis. Le pipeline distribue automatiquement votre image dans la région où il exécute la génération, et vous pouvez ajouter une distribution d'image pour d'autres régions.

Les images que vous créez à partir de votre image de base personnalisée se trouvent dans votre Compte AWS. Vous pouvez configurer votre pipeline d'images pour produire des versions mises à jour et corrigées de votre image en saisissant un calendrier de génération. Lorsque le build est terminé, vous pouvez recevoir une notification via Amazon Simple Notification Service (SNS). Outre

la production d'une image finale, l'assistant de console Image Builder génère une recette qui peut être utilisée avec les systèmes de contrôle de version existants et les pipelines d'intégration continue/déploiement continu (CI/CD) pour une automatisation reproductible. Vous pouvez partager et créer de nouvelles versions de votre recette.

Contenu de la section

Éléments de l'AMI

Une Amazon Machine Image (AMI) est une image de machine virtuelle (VM) préconfigurée qui contient le système d'exploitation et le logiciel nécessaires au déploiement des instances EC2.

Une AMI inclut les éléments suivants :

- Modèle pour le volume racine de la machine virtuelle. Lorsque vous lancez une machine virtuelle Amazon EC2, le volume du périphérique racine contient l'image permettant de démarrer l'instance. Lorsque le stockage d'instance est utilisé, le périphérique racine est un volume de stockage d'instance créé à partir d'un modèle dans Amazon S3. Pour plus d'informations, consultez Amazon EC2 Root Device Volume.
- Lorsque Amazon EBS est utilisé, le périphérique racine est un volume EBS créé à partir d'un instantané EBS.
- Autorisations de lancement qui déterminent celles Comptes AWS qui peuvent lancer des machines virtuelles avec l'AMI.
- Bloquez les données de mappage des périphériques qui spécifient les volumes à associer à l'instance après le lancement.
- Un identifiant de ressource unique pour chaque région, pour chaque compte.

- Charges utiles de métadonnées, telles que les balises, et propriétés, telles que la région, le système d'exploitation, l'architecture, le type de périphérique racine, le fournisseur, les autorisations de lancement, le stockage pour le périphérique racine et le statut de signature.
- Signature AMI pour les images Windows afin de les protéger contre toute altération non autorisée. Pour plus d'informations, consultez la section Documents d'identité de l'instance.

Quotas par défaut

Pour consulter les quotas par défaut pour Image Builder, consultez la section Points de terminaison et quotas d'Image Builder.

AWS Régions et points de terminaison

Pour consulter les points de terminaison de service pour Image Builder, consultez la section Points de terminaison et quotas d'Image Builder.

Gestion des composants

EC2 Image Builder utilise une AWS Task Orchestrator and Executor application de gestion des composants AWSTOE() qui vous aide à orchestrer des flux de travail complexes, à modifier les configurations des systèmes et à tester vos systèmes à l'aide de composants de script basés sur YAML. Comme AWSTOE il s'agit d'une application autonome, elle ne nécessite aucune configuration supplémentaire. Il peut fonctionner sur n'importe quelle infrastructure cloud et sur site. Pour commencer à l'utiliser en AWSTOE tant qu'application autonome, voirCommencez avec AWSTOE.

Image Builder permet AWSTOE d'effectuer toutes les activités sur instance. Il s'agit notamment de créer et de valider votre image avant de prendre un instantané, et de tester l'instantané pour vous assurer qu'il fonctionne comme prévu avant de créer l'image finale. Pour plus d'informations sur la façon dont Image AWSTOE Builder gère ses composants, consultezGérez les composants avec Image Builder. Pour plus d'informations sur la création de composants avec AWSTOE, consultezAWS Task Orchestrator and Executor gestionnaire de composants.

Test d'image

Vous pouvez utiliser des composants de AWSTOE test pour valider votre image et vous assurer qu'elle fonctionne comme prévu avant de créer l'image finale.

En général, chaque composant de test consiste en un document YAML contenant un script de test, un binaire de test et des métadonnées de test. Le script de test contient les commandes d'orchestration permettant de démarrer le binaire de test, qui peut être écrit dans n'importe quelle langue prise en charge par le système d'exploitation. Les codes d'état de sortie indiquent le résultat du test. Les métadonnées du test décrivent le test et son comportement ; par exemple, le nom, la description, les chemins vers le binaire de test et la durée prévue.

Gestion des versions sémantique

Image Builder utilise le versionnement sémantique pour organiser les ressources et s'assurer qu'elles possèdent des identifiants uniques. La version sémantique comporte quatre nœuds :

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

Vous pouvez attribuer des valeurs aux trois premiers et filtrer en les appliquant tous.

Le versionnement sémantique est inclus dans le nom de ressource Amazon (ARN) de chaque objet, au niveau qui s'applique à cet objet comme suit :

1. Les ARN sans version et les ARN de nom n'incluent de valeurs spécifiques dans aucun des nœuds. Les nœuds sont soit complètement omis, soit spécifiés sous forme de caractères génériques, par exemple : x.x.x.
2. <major>Les ARN de version n'ont que les trois premiers nœuds :. <minor>. <patch>
3. Les ARN de version de build possèdent les quatre nœuds et pointent vers une version spécifique pour une version spécifique d'un objet.

Affectation : pour les trois premiers nœuds, vous pouvez attribuer n'importe quelle valeur entière positive, y compris zéro, avec une limite supérieure de 2^30-1 ou 1 073 741 823 pour chaque nœud. Image Builder attribue automatiquement le numéro de compilation au quatrième nœud.

Modèles : vous pouvez utiliser n'importe quel modèle numérique qui respecte les exigences d'attribution pour les nœuds que vous pouvez attribuer. Par exemple, vous pouvez choisir un modèle de version de logiciel, tel que 1.0.0, ou une date, telle que 2021.01.01.

Sélection : avec le versionnement sémantique, vous pouvez utiliser des caractères génériques (x) pour spécifier les versions ou les nœuds les plus récents lors de la sélection de l'image de base ou des composants de votre recette. Lorsque vous utilisez un caractère générique dans un nœud, tous

les nœuds situés à droite du premier caractère générique doivent également être des caractères génériques.

Par exemple, étant donné les versions récentes suivantes : 2.2.4, 1.7.8 et 1.6.8, la sélection de version à l'aide de caractères génériques produit les résultats suivants :

- `x.x.x`= 2,2,4
- `1.x.x`= 1,7.8
- `1.6.x`= 1,6.8
- `x.2.x`n'est pas valide et produit une erreur
- `1.x.8`n'est pas valide et produit une erreur

Ressources créées

Lorsque vous créez un pipeline, aucune ressource externe à Image Builder n'est créée, sauf si ce qui suit est vrai :

- Lorsqu'une image est créée dans le cadre du planning du pipeline
- Lorsque vous choisissez Run Pipeline dans le menu Actions de la console Image Builder
- Lorsque vous exécutez l'une de ces commandes depuis l'API ou AWS CLI : StartImagePipelineExecution ou CreateImage

Les ressources suivantes sont créées au cours du processus de création de l'image :

Pipelines d'images AMI

- Instance EC2 (temporaire)
- Association d'inventaire de Systems Manager (via le gestionnaire d'état de Systems Manager s'`EnhancedImageMetadata`il est activé) sur l'instance EC2
- AMI Amazon EC2
- Le snapshot Amazon EBS associé à l'AMI Amazon EC2

Pipelines d'images de conteneurs

- Conteneur Docker exécuté sur une instance EC2 (temporaire)

- L'association d'inventaire de Systems Manager (via Systems Manager State Manager) `EnhancedImageMetadata` est activée) sur l'instance EC2
- Image de conteneur Docker
- Dockerfile

Une fois l'image créée, toutes les ressources temporaires sont supprimées.

Distribution

EC2 Image Builder peut distribuer des AMI ou des images de conteneur dans AWS n'importe quelle région. L'image est copiée dans chaque région que vous spécifiez dans le compte utilisé pour créer l'image.

Pour les images de sortie d'AMI, vous pouvez définir des autorisations de lancement d'AMI afin de contrôler Comptes AWS les personnes autorisées à lancer des instances EC2 avec l'AMI créée. Par exemple, vous pouvez rendre l'image privée, publique ou partager avec des comptes spécifiques. Si vous distribuez l'AMI à d'autres régions et que vous définissez des autorisations de lancement pour d'autres comptes, les autorisations de lancement sont propagées aux AMI de toutes les régions dans lesquelles l'AMI est distribuée.

Vous pouvez également utiliser votre AWS Organizations compte pour imposer des restrictions aux comptes membres afin de lancer des instances uniquement avec des AMI approuvées et conformes. Pour plus d'informations, consultez la section Gérer le Comptes AWS au sein de votre organisation.

Pour mettre à jour vos paramètres de distribution à l'aide de la console Image Builder, suivez les étapes de Création d'une nouvelle version de recette imagée (console) ouCréez une nouvelle version de recette de conteneur avec la console.

Partage de ressources

Pour partager des composants, des recettes ou des images avec d'autres comptes ou au sein d'autres comptes AWS Organizations, voirPartagez les ressources d'EC2 Image Builder.

Conformité d'

Pour CIS, EC2 Image Builder utilise Amazon Inspector pour évaluer l'exposition, les vulnérabilités et les écarts par rapport aux meilleures pratiques et aux normes de conformité. Par exemple,

Image Builder évalue l'accessibilité involontaire au réseau, les CVE non corrigés, la connectivité Internet publique et l'activation de la connexion root à distance. Amazon Inspector est proposé en tant que composant de test que vous pouvez choisir d'ajouter à votre recette d'images. Pour plus d'informations sur Amazon Inspector, consultez le guide de l'utilisateur d'Amazon Inspector. Pour le renforcement, EC2 Image Builder valide avec STIG. Pour une liste complète des composants STIG disponibles via Image Builder, consultezAmazon a géré les composants de renforcement STIG pour EC2 Image Builder. Pour plus d'informations, consultez le Center for Internet Security (CIS) Benchmarks.

Commencez avec EC2 Image Builder

Ce chapitre vous aide à configurer votre environnement et à créer pour la première fois un pipeline d'images ou un pipeline de conteneurs automatisé à l'aide de l'assistant de console EC2 Image Builder Create Image Pipeline.

Table des matières

Prérequis

Vérifiez les conditions préalables suivantes pour créer un pipeline d'images avec EC2 Image Builder. Sauf indication contraire, des conditions préalables sont requises pour tous les types de pipelines.

Rôle lié au service EC2 Image Builder

EC2 Image Builder utilise un rôle lié à un service pour accorder des autorisations à AWS d'autres services en votre nom. Vous n'avez pas besoin de créer manuellement un rôle lié à un service. Lorsque vous créez votre première ressource Image Builder dans la console AWS de gestion AWS CLI, ou dans l' AWS API, Image Builder crée pour vous le rôle lié au service. Pour plus d'informations sur le rôle lié à un service créé par Image Builder dans votre compte, consultez. Utilisation de rôles liés à un service pour EC2 Image Builder

Exigences de configuration

- Image Builder prend en charge AWS PrivateLink. Pour plus d'informations sur la configuration des points de terminaison VPC pour Image Builder, consultez. EC2 Image Builder et points de terminaison VPC d'interface ()AWS PrivateLink
- Image Builder est compatible avec EC2-Classic.
- Les instances utilisées par Image Builder pour créer des images de conteneur doivent disposer d'un accès Internet pour les télécharger AWS CLI depuis Amazon S3 et pour télécharger une image de base depuis le référentiel Docker Hub, le cas échéant. Image Builder utilise le AWS

CLI pour obtenir le Dockerfile à partir de la recette du conteneur, où il est stocké sous forme de données.

- Les instances utilisées par Image Builder pour créer des images et exécuter des tests doivent avoir accès au service Systems Manager. Les exigences d'installation dépendent de votre système d'exploitation.

  Pour connaître les exigences d'installation de votre image de base, choisissez l'onglet correspondant au système d'exploitation de votre image de base.

  Linux

  Pour les instances Linux Amazon EC2, Image Builder installe l'agent Systems Manager sur l'instance de build s'il n'est pas déjà présent, et le supprime avant de créer l'image.

  Windows

  Image Builder n'installe pas l'agent Systems Manager sur les instances de build Windows Server d'Amazon EC2. Si votre image de base n'est pas préinstallée avec l'agent Systems Manager, vous devez lancer une instance à partir de votre image source, installer manuellement Systems Manager sur l'instance et créer une nouvelle image de base à partir de votre instance.

  Pour installer manuellement l'agent Systems Manager sur votre instance Amazon EC2 Windows Server, consultez la section Installation manuelle de l'agent Systems Manager sur les instances EC2 pour Windows Server dans le guide de l'AWS Systems Manager utilisateur.

Référentiel de conteneurs (pipelines d'images de conteneurs)

Pour les pipelines d'images de conteneurs, la recette définit la configuration des images Docker produites et stockées dans le référentiel de conteneurs cible. Vous devez créer le référentiel cible avant de créer la recette de conteneur pour votre image Docker.

Image Builder utilise Amazon ECR comme référentiel cible pour les images de conteneurs. Pour créer un référentiel Amazon ECR, suivez les étapes décrites dans la section Création d'un référentiel dans le guide de l'utilisateur d'Amazon Elastic Container Registry.

AWS Identity and Access Management (JE SUIS)

Le rôle IAM que vous associez à votre profil d'instance doit être autorisé à exécuter les composants de génération et de test inclus dans votre image. Les politiques de rôle IAM suivantes doivent être associées au rôle IAM associé au profil d'instance :

- EC2InstanceProfileForImageBuilder
- EC2InstanceProfileForImageBuilderECRContainerBuilds
- Amazon SMS ManagedInstanceCore

Si vous configurez la journalisation, le profil d'instance spécifié dans la configuration de votre infrastructure doit disposer d'`s3:PutObject`autorisations pour le compartiment cible (`arn:aws:s3:::BucketName/*`). Par exemple :

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

Attacher une stratégie

Les étapes suivantes vous guident tout au long du processus d'attachement des politiques IAM à un rôle IAM pour accorder les autorisations précédentes.

1. Connectez-vous à la console de AWS gestion et ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation de gauche, choisissez Politiques.
3. Filtrer la liste des politiques avec EC2InstanceProfileForImageBuilder
4. Sélectionnez la bullet à côté de la politique, puis dans la liste déroulante des actions de stratégie, sélectionnez Joindre.

5. Sélectionnez le nom du rôle IAM auquel vous souhaitez associer la politique.
6. Choisissez Attach policy (Attacher une politique).
7. Répétez les étapes 3 à 6 pour les politiques EC2InstanceProfileForImageBuilderECRContainerBuildset AmazonSSM ManagedInstanceCore.

> **Note**
>
> Si vous souhaitez copier une image créée avec Image Builder vers un autre compte, vous devez créer le `EC2ImageBuilderDistributionCrossAccountRole` rôle dans tous les comptes cibles et associer la politique Stratégie Ec2ImageBuilderCrossAccountDistributionAccess gérée au rôle. Pour plus d'informations, consultez Partagez les ressources d'EC2 Image Builder.

Accédez à EC2 Image Builder

Vous pouvez gérer EC2 Image Builder à partir de l'une des interfaces suivantes.

- Page d'accueil de la console EC2 Image Builder. Depuis la console EC2 Image Builder.
- AWS Command Line Interface (AWS CLI). Vous pouvez utiliser le AWS CLI pour accéder aux opérations de AWS l'API. Pour plus d'informations, consultez la section Installation de l'interface de ligne de AWS commande dans le guide de AWS Command Line Interface l'utilisateur.
- AWS Outils pour les SDK. Vous pouvez utiliser AWS les SDK et les outils pour accéder à Image Builder et le gérer dans la langue de votre choix.

Créez un pipeline d'images à l'aide de l'assistant de console EC2 Image Builder

Ce didacticiel explique comment créer un pipeline automatique pour créer et gérer une image EC2 Image Builder personnalisée à l'aide de l'assistant de console Create image pipeline. Pour vous aider à suivre les étapes de manière efficace, les paramètres par défaut sont utilisés lorsqu'ils sont disponibles et les sections facultatives sont ignorées.

Créer un flux de travail de pipeline d'images

- Étape 1 : Spécifier les détails du pipeline

Étape 1 : Spécifier les détails du pipeline

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Pour commencer à créer votre pipeline, choisissez Create image pipeline.
3. Dans la section Général, entrez le nom de votre pipeline (obligatoire).

   > **Tip**
   >
   > La collecte de métadonnées améliorée est activée par défaut. Pour garantir la compatibilité entre les composants et les images de base, maintenez-le activé.

4. Dans la section Créer un calendrier, vous pouvez conserver les valeurs par défaut des options de planification. Notez que le fuseau horaire indiqué pour le calendrier par défaut est le temps universel coordonné (UTC). Pour plus d'informations sur l'heure UTC et pour connaître le décalage correspondant à votre fuseau horaire, consultez la section Abréviations des fuseaux horaires — Liste mondiale.

   Pour les paramètres de mise à jour des dépendances, choisissez l'option Exécuter le pipeline à l'heure planifiée s'il existe des mises à jour des dépendances. Ce paramètre oblige votre pipeline à vérifier les mises à jour avant de démarrer la génération. S'il n'y a aucune mise à jour, la construction planifiée du pipeline est ignorée.

   > **Note**
   >
   > Pour vous assurer que votre pipeline reconnaît les mises à jour et les builds des dépendances comme prévu, vous devez utiliser le versionnement sémantique (x.x.x) pour votre image de base et vos composants. Pour en savoir plus sur le versionnement sémantique des ressources Image Builder, consultez. Gestion des versions sémantique

5. Choisissez Next pour passer à l'étape suivante.

Étape 2 : Choisissez la recette

1. Image Builder utilise par défaut la recette existante dans la section Recette. Pour la première fois, choisissez l'option Créer une nouvelle recette.
2. Dans la section Type d'image, choisissez l'option Amazon Machine Image (AMI) pour créer un pipeline d'images qui produira et distribuera une AMI.
3. Dans la section Général, entrez les cases obligatoires suivantes :

   - Nom — le nom de votre recette
   - Version : version de votre recette (utilisez le format) <major>. <minor>. <patch>, où major, minor et patch sont des valeurs entières). Les nouvelles recettes commencent généralement par`1.0.0`.
4. Dans la section Image source, conservez les valeurs par défaut pour Sélectionner une image, Système d'exploitation de l'image (OS) et Origine de l'image. Il en résulte une liste d'AMI Amazon Linux 2, gérées par Amazon, parmi lesquelles vous pouvez choisir pour votre image de base.

   a. Dans le menu déroulant Nom de l'image, choisissez une image.

   b. Conservez la valeur par défaut pour les options de gestion automatique des versions (utilisez la dernière version du système d'exploitation disponible).

   > **Note**
   >
   > Ce paramètre garantit que votre pipeline utilise le versionnement sémantique pour l'image de base, afin de détecter les mises à jour des dépendances pour les tâches planifiées automatiquement. Pour en savoir plus sur le versionnement sémantique des ressources Image Builder, consultez. Gestion des versions sémantique
5. Dans la section Configuration de l'instance, conservez les valeurs par défaut pour l'agent Systems Manager. Image Builder conserve donc l'agent Systems Manager une fois la compilation et les tests terminés, afin d'inclure l'agent Systems Manager dans votre nouvelle image.

   Laissez les données utilisateur vides pour ce didacticiel. Vous pouvez utiliser cette zone à d'autres moments pour fournir des commandes ou un script de commande à exécuter lorsque vous lancez votre instance de build. Toutefois, il remplace toutes les commandes qu'Image Builder aurait pu ajouter pour garantir l'installation de Systems Manager. Lorsque vous l'utilisez,

assurez-vous que l'agent Systems Manager est préinstallé sur votre image de base ou que vous incluez l'installation dans vos données utilisateur.

6. Dans la section Composants, vous devez sélectionner au moins un composant de construction.

   Dans le panneau Build components — Amazon Linux, vous pouvez parcourir les composants répertoriés sur la page. Utilisez le contrôle de pagination situé dans le coin supérieur droit pour parcourir les composants supplémentaires disponibles pour votre système d'exploitation d'image de base. Vous pouvez également rechercher des composants spécifiques ou créer votre propre composant de construction à l'aide du gestionnaire de composants.

   Pour ce didacticiel, choisissez un composant qui met à jour Linux avec les dernières mises à jour de sécurité, comme suit :

   a. Filtrez les résultats en saisissant le mot `update` dans la barre de recherche située en haut du panneau.

   b. Cochez la case correspondant au composant de `update-linux` construction.

   c. Faites défiler la page vers le bas, puis dans le coin supérieur droit de la liste des composants sélectionnés, sélectionnez Tout développer.

   d. Conservez la valeur par défaut pour les options de version (utilisez la dernière version de composant disponible).

   > **Note**
   >
   > Ce paramètre garantit que votre pipeline utilise le versionnement sémantique pour le composant sélectionné, afin de détecter les mises à jour de dépendance pour les tâches planifiées automatiquement. Pour en savoir plus sur le versionnement sémantique des ressources Image Builder, consultez. Gestion des versions sémantique

   Si vous avez sélectionné un composant doté de paramètres d'entrée, vous verrez également les paramètres dans cette zone. Les paramètres ne sont pas abordés dans ce didacticiel. Pour plus d'informations sur l'utilisation des paramètres d'entrée dans vos composants et leur définition dans vos recettes, consultezGérez les paramètres des AWSTOE composants avec EC2 Image Builder.

   Réorganiser les composants (facultatif)

Si vous avez choisi plusieurs composants à inclure dans votre image, vous pouvez utiliser cette drag-and-drop action pour les réorganiser dans l'ordre dans lequel ils doivent s'exécuter pendant le processus de création.

> **Note**
>
> Les composants de durcissement CIS ne respectent pas les règles de classement des composants standard des recettes Image Builder. Les composants de renforcement CIS sont toujours exécutés en dernier pour garantir que les tests de référence s'exécutent par rapport à votre image de sortie.

1. Revenez à la liste des composants disponibles.
2. Cochez la case correspondant au composant de `update-linux-kernel-mainline` construction (ou à tout autre composant de votre choix).
3. Faites défiler la page jusqu'à la liste des composants sélectionnés pour voir qu'il y a au moins deux résultats.
4. Il est possible que les paramètres de version ou de paramètres d'entrée des composants récemment ajoutés ne soient pas étendus. Pour développer les paramètres des options de version ou des paramètres d'entrée, vous pouvez cliquer sur la flèche à côté du nom du paramètre. Pour étendre tous les paramètres de tous les composants sélectionnés, vous pouvez activer ou désactiver le bouton Tout étendre.
5. Choisissez l'un des composants et faites-le glisser vers le haut ou vers le bas pour modifier l'ordre dans lequel les composants seront exécutés.
6. Pour supprimer le `update-linux-kernel-mainline` composant, choisissez `X` dans le coin supérieur droit de la boîte du composant.
7. Répétez l'étape précédente pour supprimer tous les autres composants que vous avez éventuellement ajoutés, en ne laissant que le `update-linux` composant sélectionné.

7. Choisissez Next pour passer à l'étape suivante.

Étape 3 : définir la configuration de l'infrastructure (facultatif)

Image Builder lance des instances EC2 dans votre compte pour personnaliser les images et exécuter des tests de validation. Les paramètres de configuration de l'infrastructure spécifient les détails de

l'infrastructure pour les instances qui s'exécuteront dans votre Compte AWS système pendant le processus de génération.

Dans la section Configuration de l'infrastructure, les options de configuration sont définies par défaut sur`Create infrastructure configuration using service defaults`. Cela crée un rôle IAM et un profil d'instance associé pour les instances de build et de test EC2 utilisées pour configurer votre image. Pour plus d'informations sur les paramètres de configuration de l'infrastructure, consultez le CreateInfrastructureConfigurationmanuel de référence de l'API EC2 Image Builder.

Pour ce didacticiel, nous utilisons les paramètres par défaut.

> **Note**
>
> Pour spécifier un sous-réseau à utiliser pour un VPC privé, vous pouvez créer votre propre configuration d'infrastructure personnalisée ou utiliser des paramètres que vous avez déjà créés.

- Choisissez Next pour passer à l'étape suivante.

Étape 4 : définir les paramètres de distribution (facultatif)

Les configurations de distribution incluent le nom de l'AMI de sortie, les paramètres régionaux spécifiques pour le chiffrement, les autorisations de lancement Comptes AWS, les organisations et unités organisationnelles (UO) qui peuvent lancer l'AMI de sortie, ainsi que les configurations de licence.

Dans la section Paramètres de distribution, les options de configuration sont définies par défaut sur`Create distribution settings using service defaults`. Cette option distribuera l'AMI de sortie à la région actuelle. Pour plus d'informations sur la configuration de vos paramètres de distribution, consultezGérer les paramètres de distribution d'EC2 Image Builder.

Pour ce didacticiel, nous utilisons les paramètres par défaut.

- Choisissez Next pour passer à l'étape suivante.

Étape 5 : Vérifier

La section Révision affiche tous les paramètres que vous avez configurés. Pour modifier les informations d'une section donnée, cliquez sur le bouton Modifier situé dans le coin supérieur droit de la section d'étape. Par exemple, si vous souhaitez modifier le nom de votre pipeline, cliquez sur le bouton Modifier dans le coin supérieur droit de la section Étape 1 : Détails du pipeline.

1. Lorsque vous avez revu vos paramètres, choisissez Create pipeline pour créer votre pipeline.
2. Vous pouvez voir les messages de réussite ou d'échec en haut de la page, lorsque vos ressources sont créées pour les paramètres de distribution, la configuration de l'infrastructure, votre nouvelle recette et le pipeline. Pour voir les détails d'une ressource, y compris son identifiant, choisissez Afficher les détails.
3. Après avoir consulté les détails d'une ressource, vous pouvez consulter les détails des autres ressources en choisissant le type de ressource dans le volet de navigation. Par exemple, pour voir les détails de votre nouveau pipeline, choisissez Image pipelines dans le volet de navigation. Si votre construction a réussi, votre nouveau pipeline est affiché dans la liste des pipelines d'images.

Étape 6 : Nettoyer

Votre environnement Image Builder, tout comme votre maison, a besoin d'un entretien régulier pour vous aider à trouver ce dont vous avez besoin et à accomplir vos tâches sans vous encombrer. Assurez-vous de nettoyer régulièrement les ressources temporaires que vous avez créées pour les tests. Sinon, vous pourriez oublier ces ressources et, plus tard, ne plus vous souvenir de leur utilisation. D'ici là, il se peut que vous ne sachiez pas si vous pouvez vous en débarrasser en toute sécurité.

> **Tip**
>
> Pour éviter les erreurs de dépendance lorsque vous supprimez des ressources, veillez à supprimer vos ressources dans l'ordre suivant :
>
> 1. Pipeline d'images
> 2. Recette d'images
> 3. Toutes les ressources restantes

Pour nettoyer les ressources que vous avez créées pour ce didacticiel, procédez comme suit :

Supprimer le pipeline

1. Pour voir la liste des pipelines de génération créés sous votre compte, choisissez Image pipelines dans le volet de navigation.
2. Cochez la case à côté du nom du pipeline pour sélectionner le pipeline que vous souhaitez supprimer.
3. En haut du panneau Pipelines d'images, dans le menu Actions, choisissez Supprimer.
4. Pour confirmer la suppression, entrez `Delete` dans le champ et choisissez Supprimer.

Supprimer la recette

1. Pour voir la liste des recettes créées sous votre compte, choisissez Image recipes dans le volet de navigation.
2. Cochez la case à côté du nom de la recette pour sélectionner la recette que vous souhaitez supprimer.
3. En haut du panneau Recettes d'images, dans le menu Actions, choisissez Supprimer la recette.
4. Pour confirmer la suppression, entrez `Delete` dans le champ et choisissez Supprimer.

Supprimer la configuration de l'infrastructure

1. Pour consulter la liste des configurations d'infrastructure créées sous votre compte, choisissez Configuration de l'infrastructure dans le volet de navigation.
2. Cochez la case à côté du nom de la configuration pour sélectionner la configuration d'infrastructure que vous souhaitez supprimer.
3. En haut du panneau de configuration de l'infrastructure, choisissez Supprimer.
4. Pour confirmer la suppression, entrez `Delete` dans le champ et choisissez Supprimer.

Supprimer les paramètres de distribution

1. Pour voir la liste des paramètres de distribution créés sous votre compte, choisissez Paramètres de distribution dans le volet de navigation.
2. Cochez la case à côté du nom de la configuration pour sélectionner les paramètres de distribution que vous avez créés pour ce didacticiel.

3. En haut du panneau des paramètres de distribution, choisissez Supprimer.
4. Pour confirmer la suppression, entrez `Delete` dans le champ et choisissez Supprimer.

Supprimer l'image

Suivez ces étapes pour vérifier que vous avez supprimé toutes les images créées à partir du pipeline du didacticiel. Il est peu probable que ce didacticiel crée une image à moins que suffisamment de temps ne se soit écoulé depuis que vous avez créé le pipeline qu'il exécute, conformément au calendrier de génération.

1. Pour voir la liste des images créées sous votre compte, choisissez Images dans le volet de navigation.
2. Choisissez la version de l'image que vous souhaitez supprimer. Cela ouvre la page des versions de génération d'images.
3. Cochez la case à côté de la version pour toute image que vous souhaitez supprimer. Vous pouvez sélectionner plusieurs versions d'image à la fois.
4. En haut du panneau des versions de génération d'images, choisissez Supprimer la version.
5. Pour confirmer la suppression, entrez `Delete` dans le champ et choisissez Supprimer.

Créez un pipeline d'images de conteneur à l'aide de l'assistant de console EC2 Image Builder

Ce didacticiel explique comment créer un pipeline automatique pour créer et gérer une image Docker EC2 Image Builder personnalisée à l'aide de l'assistant de console Create image pipeline. Pour vous aider à suivre les étapes de manière efficace, les paramètres par défaut sont utilisés lorsqu'ils sont disponibles et les sections facultatives sont ignorées.

Créer un flux de travail de pipeline d'images

- Étape 1 : Spécifier les détails du pipeline
- Étape 2 : Choisissez la recette
- Étape 3 : définir la configuration de l'infrastructure (facultatif)
- Étape 4 : définir les paramètres de distribution (facultatif)
- Étape 5 : Vérifier
- Étape 6 : Nettoyer

Étape 1 : Spécifier les détails du pipeline

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Pour commencer à créer votre pipeline, choisissez Create image pipeline.
3. Dans la section Général, entrez le nom de votre pipeline (obligatoire).
4. Dans la section Créer un calendrier, vous pouvez conserver les valeurs par défaut des options de planification. Notez que le fuseau horaire indiqué pour le calendrier par défaut est le temps universel coordonné (UTC). Pour plus d'informations sur l'heure UTC et pour connaître le décalage correspondant à votre fuseau horaire, consultez la section Abréviations des fuseaux horaires — Liste mondiale.

   Pour les paramètres de mise à jour des dépendances, choisissez l'option Exécuter le pipeline à l'heure planifiée s'il existe des mises à jour des dépendances. Ce paramètre oblige votre pipeline à vérifier les mises à jour avant de démarrer la génération. S'il n'y a aucune mise à jour, la construction planifiée du pipeline est ignorée.

   > **Note**
   >
   > Pour vous assurer que votre pipeline reconnaît les mises à jour et les builds des dépendances comme prévu, vous devez utiliser le versionnement sémantique (x.x.x) pour votre image de base et vos composants. Pour en savoir plus sur le versionnement sémantique des ressources Image Builder, consultez. Gestion des versions sémantique

5. Choisissez Next pour passer à l'étape suivante.

Étape 2 : Choisissez la recette

1. Image Builder utilise par défaut la recette existante dans la section Recette. Pour la première fois, choisissez l'option Créer une nouvelle recette.
2. Dans la section Type d'image, choisissez l'option image Docker pour créer un pipeline de conteneurs qui produira une image Docker et la distribuera aux référentiels Amazon ECR dans les régions cibles.
3. Dans la section Général, entrez les cases obligatoires suivantes :

   - Nom — le nom de votre recette

- Version : version de votre recette (utilisez le format) <major>. <minor>. <patch>, où major, minor et patch sont des valeurs entières). Les nouvelles recettes commencent généralement par`1.0.0`.

4. Dans la section Image source, conservez les valeurs par défaut pour Sélectionner une image, Système d'exploitation de l'image (OS) et Origine de l'image. Il en résulte une liste d'images de conteneur Amazon Linux 2, gérées par Amazon, parmi lesquelles vous pouvez choisir pour votre image de base.

   a. Dans le menu déroulant Nom de l'image, choisissez une image.

   b. Conservez la valeur par défaut pour les options de gestion automatique des versions (utilisez la dernière version du système d'exploitation disponible).

   > **Note**
   >
   > Ce paramètre garantit que votre pipeline utilise le versionnement sémantique pour l'image de base, afin de détecter les mises à jour des dépendances pour les tâches planifiées automatiquement. Pour en savoir plus sur le versionnement sémantique des ressources Image Builder, consultez. Gestion des versions sémantique

5. Dans la section Composants, vous devez sélectionner au moins un composant de construction.

   Dans le panneau Build components — Amazon Linux, vous pouvez parcourir les composants répertoriés sur la page. Utilisez le contrôle de pagination situé dans le coin supérieur droit pour parcourir les composants supplémentaires disponibles pour votre système d'exploitation d'image de base. Vous pouvez également rechercher des composants spécifiques ou créer votre propre composant de construction à l'aide du gestionnaire de composants.

   Pour ce didacticiel, choisissez un composant qui met à jour Linux avec les dernières mises à jour de sécurité, comme suit :

   a. Filtrez les résultats en saisissant le mot `update` dans la barre de recherche située en haut du panneau.

   b. Cochez la case correspondant au composant de `update-linux` construction.

   c. Faites défiler la page vers le bas, puis dans le coin supérieur droit de la liste des composants sélectionnés, sélectionnez Tout développer.

   d. Conservez la valeur par défaut pour les options de version (utilisez la dernière version de composant disponible).

> **Note**
>
> Ce paramètre garantit que votre pipeline utilise le versionnement sémantique pour le composant sélectionné, afin de détecter les mises à jour de dépendance pour les tâches planifiées automatiquement. Pour en savoir plus sur le versionnement sémantique des ressources Image Builder, consultez. Gestion des versions sémantique

Si vous aviez sélectionné un composant doté de paramètres d'entrée, les paramètres s'afficheraient également dans cette zone. Les paramètres ne sont pas abordés dans ce didacticiel. Pour plus d'informations sur l'utilisation des paramètres d'entrée dans vos composants et leur définition dans vos recettes, consultezGérez les paramètres des AWSTOE composants avec EC2 Image Builder.

Réorganiser les composants (facultatif)

Si vous avez choisi plusieurs composants à inclure dans votre image, vous pouvez utiliser cette drag-and-drop action pour les réorganiser dans l'ordre dans lequel ils doivent s'exécuter pendant le processus de génération.

> **Note**
>
> Les composants de durcissement CIS ne respectent pas les règles de classement des composants standard des recettes Image Builder. Les composants de renforcement CIS sont toujours exécutés en dernier pour garantir que les tests de référence s'exécutent par rapport à votre image de sortie.

1. Revenez à la liste des composants disponibles.
2. Cochez la case correspondant au composant de `update-linux-kernel-mainline` construction (ou à tout autre composant de votre choix).
3. Faites défiler la page jusqu'à la liste des composants sélectionnés pour voir qu'il y a au moins deux résultats.
4. Il est possible que le versionnement des composants récemment ajoutés ne soit pas étendu. Pour développer les options de gestion des versions, vous pouvez soit choisir la flèche

située à côté des options de gestion des versions, soit activer ou désactiver le bouton Tout développer pour étendre le contrôle de version à tous les composants sélectionnés.

5. Choisissez l'un des composants et faites-le glisser vers le haut ou vers le bas pour modifier l'ordre dans lequel les composants seront exécutés.
6. Pour supprimer le `update-linux-kernel-mainline` composant, choisissez `X` dans le coin supérieur droit de la boîte du composant.
7. Répétez l'étape précédente pour supprimer tous les autres composants que vous avez éventuellement ajoutés, en ne laissant que le `update-linux` composant sélectionné.

6. Dans la section Modèle Dockerfile, sélectionnez l'option Utiliser un exemple. Dans le panneau Contenu, notez les variables contextuelles dans lesquelles Image Builder place les informations de build ou les scripts, en fonction de la recette de votre image de conteneur.

   Par défaut, Image Builder utilise les variables contextuelles suivantes dans votre Dockerfile.

   ParentImage (obligatoire)

   Au moment de la création, cette variable devient l'image de base de votre recette.

   Exemple :

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

   environnements (obligatoire si des composants sont spécifiés)

   Cette variable sera convertie en un script qui exécute des composants.

   Exemple :

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

   composants (facultatif)

   Image Builder résout les scripts de création et de test des composants pour les composants inclus dans la recette du conteneur. Cette variable peut être placée n'importe où dans le Dockerfile, après la variable d'environnement.

Exemple :

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

7. Dans la section Référentiel cible, spécifiez le nom du référentiel Amazon ECR que vous avez créé comme condition préalable à ce didacticiel. Ce référentiel est utilisé comme paramètre par défaut pour la configuration de distribution dans la région où s'exécute le pipeline (région 1).

> **Note**
>
> Le référentiel cible doit exister dans Amazon ECR pour toutes les régions cibles avant sa distribution.

8. Choisissez Next pour passer à l'étape suivante.

Étape 3 : définir la configuration de l'infrastructure (facultatif)

Image Builder lance des instances EC2 dans votre compte pour personnaliser les images et exécuter des tests de validation. Les paramètres de configuration de l'infrastructure spécifient les détails de l'infrastructure pour les instances qui s'exécuteront dans votre Compte AWS système pendant le processus de génération.

Dans la section Configuration de l'infrastructure, les options de configuration sont définies par défaut sur`Create infrastructure configuration using service defaults`. Cela crée un rôle IAM et un profil d'instance associé qui sont utilisés par les instances de build pour configurer vos images de conteneur. Vous pouvez également créer votre propre configuration d'infrastructure personnalisée ou utiliser des paramètres que vous avez déjà créés. Pour plus d'informations sur les paramètres de configuration de l'infrastructure, consultez le CreateInfrastructureConfigurationmanuel de référence de l'API EC2 Image Builder.

Pour ce didacticiel, nous utilisons les paramètres par défaut.

- Choisissez Next pour passer à l'étape suivante.

Étape 4 : définir les paramètres de distribution (facultatif)

Les paramètres de distribution comprennent les régions cibles et le nom du référentiel Amazon ECR cible. Les images Docker en sortie sont déployées dans le référentiel Amazon ECR désigné dans chaque région.

Dans la section Paramètres de distribution, les options de configuration sont définies par défaut sur`Create distribution settings using service defaults`. Cette option distribuera l'image Docker de sortie au référentiel Amazon ECR spécifié dans votre recette de conteneur pour la région où s'étend votre pipeline (région 1). Si vous le souhaitez`Create new distribution settings`, vous pouvez remplacer le référentiel ECR pour la région actuelle et ajouter d'autres régions à distribuer.

Pour ce didacticiel, nous utilisons les paramètres par défaut.

- Choisissez Next pour passer à l'étape suivante.

Étape 5 : Vérifier

La section Révision affiche tous les paramètres que vous avez configurés. Pour modifier les informations d'une section donnée, cliquez sur le bouton Modifier situé dans le coin supérieur droit de la section d'étape. Par exemple, si vous souhaitez modifier le nom de votre pipeline, cliquez sur le bouton Modifier dans le coin supérieur droit de la section Étape 1 : Détails du pipeline.

1. Lorsque vous avez revu vos paramètres, choisissez Create pipeline pour créer votre pipeline.
2. Vous pouvez voir les messages de réussite ou d'échec en haut de la page, lorsque vos ressources sont créées pour les paramètres de distribution, la configuration de l'infrastructure, votre nouvelle recette et le pipeline. Pour voir les détails d'une ressource, y compris son identifiant, choisissez Afficher les détails.
3. Après avoir consulté les détails d'une ressource, vous pouvez consulter les détails des autres ressources en choisissant le type de ressource dans le volet de navigation. Par exemple, pour voir les détails de votre nouveau pipeline, choisissez Image pipelines dans le volet de navigation. Si votre construction a réussi, votre nouveau pipeline est affiché dans la liste des pipelines d'images.

Étape 6 : Nettoyer

Votre environnement Image Builder, tout comme votre maison, a besoin d'un entretien régulier pour vous aider à trouver ce dont vous avez besoin et à accomplir vos tâches sans vous encombrer. Assurez-vous de nettoyer régulièrement les ressources temporaires que vous avez créées pour les tests. Sinon, vous pourriez oublier ces ressources et, plus tard, ne plus vous souvenir de leur utilisation. D'ici là, il se peut que vous ne sachiez pas si vous pouvez vous en débarrasser en toute sécurité.

> **Tip**
>
> Pour éviter les erreurs de dépendance lorsque vous supprimez des ressources, veillez à supprimer vos ressources dans l'ordre suivant :
>
> 1. Pipeline d'images
> 2. Recette d'images
> 3. Toutes les ressources restantes

Pour nettoyer les ressources que vous avez créées pour ce didacticiel, procédez comme suit :

Supprimer le pipeline

1. Pour voir la liste des pipelines de génération créés sous votre compte, choisissez Image pipelines dans le volet de navigation.
2. Cochez la case à côté du nom du pipeline pour sélectionner le pipeline que vous souhaitez supprimer.
3. En haut du panneau Pipelines d'images, dans le menu Actions, choisissez Supprimer.
4. Pour confirmer la suppression, entrez `Delete` dans le champ et choisissez Supprimer.

Supprimer la recette du contenant

1. Pour voir la liste des recettes de conteneurs créées sous votre compte, choisissez Recettes de conteneurs dans le volet de navigation.
2. Cochez la case à côté du nom de la recette pour sélectionner la recette que vous souhaitez supprimer.
3. En haut du panneau Container recipes, dans le menu Actions, sélectionnez Supprimer la recette.

4. Pour confirmer la suppression, entrez `Delete` dans le champ et choisissez Supprimer.

Supprimer la configuration de l'infrastructure

1. Pour consulter la liste des configurations d'infrastructure créées sous votre compte, choisissez Configuration de l'infrastructure dans le volet de navigation.
2. Cochez la case à côté du nom de la configuration pour sélectionner la configuration d'infrastructure que vous souhaitez supprimer.
3. En haut du panneau de configuration de l'infrastructure, choisissez Supprimer.
4. Pour confirmer la suppression, entrez `Delete` dans le champ et choisissez Supprimer.

Supprimer les paramètres de distribution

1. Pour voir la liste des paramètres de distribution créés sous votre compte, choisissez Paramètres de distribution dans le volet de navigation.
2. Cochez la case à côté du nom de la configuration pour sélectionner les paramètres de distribution que vous avez créés pour ce didacticiel.
3. En haut du panneau des paramètres de distribution, choisissez Supprimer.
4. Pour confirmer la suppression, entrez `Delete` dans le champ et choisissez Supprimer.

Supprimer l'image

Suivez ces étapes pour vérifier que vous avez supprimé toutes les images créées à partir du pipeline du didacticiel. Il est peu probable que ce didacticiel crée une image à moins que suffisamment de temps ne se soit écoulé depuis que vous avez créé le pipeline qu'il exécute, conformément au calendrier de génération.

1. Pour voir la liste des images créées sous votre compte, choisissez Images dans le volet de navigation.
2. Choisissez la version de l'image que vous souhaitez supprimer. Cela ouvre la page des versions de génération d'images.
3. Cochez la case à côté de la version pour toute image que vous souhaitez supprimer. Vous pouvez sélectionner plusieurs versions d'image à la fois.
4. En haut du panneau des versions de génération d'images, choisissez Supprimer la version.
5. Pour confirmer la suppression, entrez `Delete` dans le champ et choisissez Supprimer.

AWS Task Orchestrator and Executor gestionnaire de composants

EC2 Image Builder utilise AWS Task Orchestrator and Executor l'application AWSTOE() pour orchestrer des flux de travail complexes, modifier les configurations des systèmes et tester vos systèmes sans écrire de code. Cette application gère et exécute les composants qui implémentent son schéma de document déclaratif.

Comme il s'agit d'une application autonome, elle ne nécessite aucune configuration de serveur supplémentaire. Il peut fonctionner sur n'importe quelle infrastructure cloud et sur site.

Table des matières

AWSTOE téléchargements

Pour l'installer AWSTOE, choisissez le lien de téléchargement correspondant à votre architecture et à votre plate-forme. Si vous vous connectez à un point de terminaison VPC pour votre service (Image Builder, par exemple), il doit être associé à une politique de point de terminaison personnalisée qui inclut l'accès au compartiment S3 pour AWSTOE les téléchargements. Dans le cas contraire, vos instances de build et de test ne pourront pas télécharger le script bootstrap (`bootstrap.sh`) et installer l' AWSTOE application. Pour plus d'informations, consultez Création d'une politique de point de terminaison VPC pour Image Builder.

> **Important**
>
> AWS supprime progressivement le support pour les versions 1.0 et 1.1 du protocole TLS. Pour accéder au compartiment S3 à des fins de AWSTOE téléchargement, votre logiciel client doit utiliser la version 1.2 ou ultérieure du protocole TLS. Pour plus d'informations, consultez ce billet AWS de blog sur la sécurité.

Architecture	Plateforme	Lien de téléchargement	Exemple
386	AL 2 et 2023 RHEL 7 et 8 Ubuntu 16.04, 18.04, 20.04 et 22.04 CentOS 7 et 8 SUSE 12 et 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 et 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 et 2023 RHEL 7 et 8 Ubuntu 16.04, 18.04, 20.04 et 22.04 CentOS 7 et 8 CentOS Stream 8 SUSE 12 et 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

Architecture	Plateforme	Lien de téléchargement	Exemple
ARM64	AL 2 et 2023 RHEL 7 et 8 Ubuntu 16.04, 18.04, 20.04 et 22.04 CentOS 7 et 8 CentOS Stream 8 SUSE 12 et 15	`https://awstoe- <region>.s3.amazonaws.com/latest/linux/arm64/awstoe`	https://awstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/linux/arm64/awstoe

Régions prises en charge

AWSTOE est prise en charge en tant qu'application autonome dans les régions suivantes.

Région AWS nom	Région AWS
USA Est (Ohio)	us-east-2
USA Est (Virginie du Nord)	us-east-1
AWS GovCloud (USA Est)	us-gov-east-1
AWS GovCloud (US-Ouest)	us-gov-west-1
USA Ouest (Californie du Nord)	us-west-1
US West (Oregon)	us-west-2
Afrique (Le Cap)	af-south-1
Asie-Pacifique (Hong Kong)	ap-east-1
Asie-Pacifique (Osaka)	ap-northeast-3

Région AWS nom	Région AWS
Asie-Pacifique (Séoul)	ap-northeast-2
Asie-Pacifique (Mumbai)	ap-south-1
Asie-Pacifique (Hyderabad)	ap-south-2
Asie-Pacifique (Singapour)	ap-southeast-1
Asie-Pacifique (Sydney)	ap-southeast-2
Asie-Pacifique (Jakarta)	ap-southeast-3
Asie-Pacifique (Tokyo)	ap-northeast-1
Canada (Central)	ca-central-1
Europe (Francfort)	eu-central-1
Europe (Zurich)	eu-central-2
Europe (Stockholm)	eu-north-1
Europe (Milan)	eu-south-1
Europe (Espagne)	eu-south-2
Europe (Irlande)	eu-west-1
Europe (Londres)	eu-west-2
Europe (Paris)	eu-west-3
Israël (Tel Aviv)	il-central-1
Moyen-Orient (EAU)	me-central-1
Moyen-Orient (Bahreïn)	me-south-1
Amérique du Sud (São Paulo)	sa-east-1

Région AWS nom	Région AWS
Chine (Beijing)	cn-north-1
China (Ningxia)	cn-northwest-1

Commencez avec AWSTOE

L'application AWS Task Orchestrator and Executor (AWSTOE) est une application autonome qui crée, valide et exécute des commandes dans un cadre de définition de composants. AWS les services peuvent être utilisés AWSTOE pour orchestrer les flux de travail, installer des logiciels, modifier les configurations du système et tester les versions d'images.

Procédez comme suit pour installer l' AWSTOE application et l'utiliser pour la première fois.

Vérifiez la signature du téléchargement de AWSTOE l'installation

Cette section décrit le processus recommandé pour vérifier la validité du téléchargement de l'installation pour AWSTOE les systèmes d'exploitation Linux et Windows.

Rubriques

- Vérifiez la signature du téléchargement de l' AWSTOE installation sous Linux
- Vérifiez la signature du téléchargement de l' AWSTOE installation sous Windows

Vérifiez la signature du téléchargement de l' AWSTOE installation sous Linux

Cette rubrique décrit le processus recommandé pour vérifier la validité du téléchargement d'installation pour les systèmes d'exploitation basés AWSTOE sur Linux.

Chaque fois que vous téléchargez une application sur Internet, nous vous recommandons d'authentifier l'identité de l'éditeur du logiciel. Vérifiez également que l'application n'a pas été modifiée ou corrompue depuis sa publication. Cela vous évitera d'installer une version de l'application contenant un virus ou tout autre code malveillant.

Si, après avoir exécuté les étapes décrites dans cette rubrique, vous déterminez que le logiciel correspondant AWSTOE est modifié ou endommagé, n'exécutez pas le fichier d'installation. Contactez plutôt AWS Support pour plus d'informations sur vos options de support, voir AWS Support.

AWSTOE les fichiers pour les systèmes d'exploitation basés sur Linux sont signés à l'aide GnuPG d'une implémentation open source de la norme Pretty Good Privacy (OpenPGP) pour les signatures numériques sécurisées. GnuPG(également connu sous le nom deGPG) fournit une authentification et un contrôle d'intégrité par le biais d'une signature numérique. Amazon EC2 publie une clé publique et des signatures que vous pouvez utiliser pour vérifier les outils de la CLI Amazon EC2 téléchargés. Pour plus d'informations sur PGP et GnuPG (GPG), consultez http://www.gnupg.org.

La première étape consiste à établir une approbation avec l'éditeur du logiciel. Téléchargez la clé publique de l'éditeur du logiciel, vérifiez que le propriétaire de cette clé publique est bien celui qu'il prétend être, puis ajoutez la clé publique à votre porte-clés. Votre porte-clés est un ensemble de clés publiques connues. Après avoir établi l'authenticité de la clé publique, vous pouvez l'utiliser pour vérifier la signature de l'application.

Rubriques

Installation des outils GPG

Si votre système d'exploitation est Linux ou Unix, les outils GPG sont probablement déjà installés. Pour tester si les outils sont installés sur votre système, tapez gpg à partir d'une invite de commande. Si les outils GPG sont installés, une invite de commande GPG s'affiche. Si les outils GPG ne sont pas installés, un message d'erreur indiquant que la commande est introuvable s'affiche. Vous pouvez installer le package GnuPG à partir d'un référentiel.

Pour installer les outils GPG sur un système Linux basé sur Debian

- Depuis un terminal, exécutez la commande suivante : apt-get install gnupg.

Pour installer les outils GPG sur un système Linux basé sur Red Hat

- Depuis un terminal, exécutez la commande suivante : yum install gnupg.

Authentification et importation de la clé publique

L'étape suivante du processus consiste à authentifier la clé AWSTOE publique et à l'ajouter en tant que clé fiable dans votre GPG trousseau de clés.

Pour authentifier et importer la clé AWSTOE publique

1. Obtenez une copie de notre clé publique GPG en effectuant l'une des actions suivantes :

   - Téléchargez la clé depuis https://awstoe - **<region>** .s3. **<region>**.amazonaws.com/assets/awstoe.gpg. Par exemple, https://awstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/assets/awstoe.gpg.
   - Copiez la clé à partir du texte suivant et collez-la dans un fichier nommé `awstoe.gpg`. Veillez à inclure tout ce qui suit :

```
-----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. À l'invite de commande du répertoire où vous avez enregistré awstoe.gpg, utilisez la commande suivante pour importer la clé AWSTOE publique dans votre trousseau de clés.

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

   La commande renvoie des résultats semblables à ce qui suit :

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

Notez la valeur de la clé ; vous en aurez besoin lors de l'étape suivante. Dans l'exemple précédent, la valeur de la clé est `F5AEBC52`.

3. Vérifiez l'empreinte en exécutant la commande suivante, en remplaçant key-value (valeur clé) par la valeur de l'étape précédente :

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

Cette commande renvoie un résultat semblable à ce qui suit :

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

De plus, la chaîne de l'empreinte doit être identique à `F6DD E01C 869F D639 15E5 5742 DEBD C156 F5AE BC52`, comme illustré dans l'exemple précédent. Comparez l'empreinte de la clé renvoyée à celle publiée sur cette page. Elles doivent correspondre. S'ils ne correspondent pas, n'installez pas le script AWSTOE d'installation et contactez AWS Support.

Vérification de la signature du package

Après avoir installé les outils `GPG`, authentifié et importé la clé publique de AWSTOE et vérifié que la clé publique est de confiance, vous êtes prêt à vérifier la signature du script d'installation.

Pour vérifier la signature du script d'installation

1. À l'invite de commande, exécutez la commande suivante pour télécharger le fichier binaire de l'application :

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

Par exemple :

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

Les valeurs prises en charge pour **`architecture`** peuvent être `amd64`386, et`arm64`.

2. À l'invite de commande, exécutez la commande suivante pour télécharger le fichier de signature du binaire d'application correspondant à partir du même chemin de préfixe de clé S3 :

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

   Par exemple :

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

   Les valeurs prises en charge pour **`architecture`** peuvent être `amd64`386, et`arm64`.

3. Vérifiez la signature en exécutant la commande suivante à l'invite de commande dans le répertoire dans lequel vous avez enregistré `awstoe.sig` et dans le fichier AWSTOE d'installation. Ces deux fichiers doivent être présents.

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

   Le résultat doit ressembler à ce qui suit :

```
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
```

   Si le résultat contient l'expression `Good signature from "AWSTOE <awstoe@amazon.com>"`, cela signifie que la signature a été vérifiée et vous pouvez continuer à exécuter le script d'installation d' AWSTOE .

   Si le résultat inclut l'expression `BAD signature`, vérifiez si vous avez effectué la procédure correctement. Si cette réponse persiste, n'exécutez pas le fichier d'installation que vous avez téléchargé précédemment et contactez AWS Support.

Voici les informations détaillées sur les avertissements qui peuvent s'afficher :

- AVERTISSEMENT : Cette clé n'est pas certifiée par une signature fiable ! Rien n'indique que la signature appartient au propriétaire. Idéalement, vous devriez vous rendre dans un AWS bureau

et recevoir la clé en personne. Cependant, vous le téléchargeriez probablement à partir d'un site Web. Dans ce cas, le site Web est un AWS site Web.

- gpg: no ultimately trusted keys found. Cela signifie que la clé en question n'est pas « définitivement approuvée » par vous ou par d'autres personnes en qui vous avez confiance.

Pour plus d'informations, consultez http://www.gnupg.org.

Vérifiez la signature du téléchargement de l' AWSTOE installation sous Windows

Cette rubrique décrit le processus recommandé pour vérifier la validité du fichier d'installation de l' AWS Task Orchestrator and Executor application sur les systèmes d'exploitation Windows.

Lorsque vous téléchargez une application à partir d'internet, nous vous recommandons d'authentifier l'identité de l'éditeur du logiciel et de vérifier que l'application n'a pas été modifiée ou corrompue depuis sa publication. Cela vous évitera d'installer une version de l'application contenant un virus ou tout autre code malveillant.

Si, après avoir exécuté les étapes décrites dans cette rubrique, vous déterminez que le logiciel de l' AWSTOE application est modifié ou endommagé, n'exécutez pas le fichier d'installation. Au lieu de cela, contactez AWS Support.

Pour vérifier la validité du fichier binaire awstoe téléchargé sur les systèmes d'exploitation Windows, assurez-vous que l'empreinte numérique du certificat de signature Amazon Services LLC est égale à cette valeur :

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

> **Note**
>
> Pendant la période de déploiement d'un nouveau binaire, il est possible que votre certificat de signataire ne corresponde pas à la nouvelle empreinte numérique. Si votre certificat de signataire ne correspond pas, vérifiez que la valeur de l'empreinte numérique est la suivante : 5B 77 F4 F0 C3 7A 8B 89 D9 A7 8F 54 B6 85 11 CE 9E A3 BF 17

Pour vérifier cette valeur, exécutez la procédure suivante :

1. Cliquez avec le bouton droit sur le fichier `awstoe.exe` téléchargé et ouvrez la fenêtre Properties (Propriétés).

2. Choisissez l'onglet Signatures numériques.
3. Dans Signature List, choisissez Amazon Services LLC, puis Details.
4. Choisissez l'onglet General (Général), s'il n'est pas déjà sélectionné, puis View Certificate (Afficher le certificat).
5. Sélectionnez l'onglet Détails, puis sélectionnez All (Tous) dans la liste déroulante Show (Afficher), si cette option n'est pas déjà sélectionnée.
6. Faites défiler l'écran vers le bas jusqu'au champ Thumbprint (Empreinte), puis choisissez Thumbprint (Empreinte). Cela affichera la valeur complète de l'empreinte dans la fenêtre inférieure.

   - Si la valeur de l'empreinte affichée dans la fenêtre inférieure est identique à la valeur suivante :

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

     alors votre AWSTOE binaire téléchargé est authentique et peut être installé en toute sécurité.

     > **Note**
     >
     > Pendant la période de déploiement d'un nouveau binaire, il est possible que votre certificat de signataire ne corresponde pas à la nouvelle empreinte numérique. Si votre certificat de signataire ne correspond pas, vérifiez que la valeur de l'empreinte numérique est la suivante :
     > 5B 77 F4 F0 C3 7A 8B 89 D9 A7 8F 54 B6 85 11 CE 9E A3 BF 17

   - Si la valeur de l'empreinte numérique dans la fenêtre de détails inférieure n'est pas identique à la valeur précédente, ne l'exécutez pas. `awstoe.exe`

Étapes de démarrage

- Étape 1 : Installation AWSTOE
- Étape 2 : définir les AWS informations d'identification
- Étape 3 : développer les documents des composants localement
- Étape 4 : Valider AWSTOE les composants
- Étape 5 : Exécuter AWSTOE les composants

Étape 1 : Installation AWSTOE

Pour développer des composants localement, téléchargez et installez l' AWSTOE application.

1. Téléchargez l' AWSTOE application

   Pour procéder à l'installation AWSTOE, choisissez le lien de téléchargement correspondant à votre architecture et à votre plate-forme. Pour obtenir la liste complète des liens de téléchargement des applications, voir AWSTOE téléchargements

   > **Important**
   >
   > AWS supprime progressivement le support pour les versions 1.0 et 1.1 du protocole TLS. Pour accéder au compartiment S3 à des fins de AWSTOE téléchargement, votre logiciel client doit utiliser la version 1.2 ou ultérieure du protocole TLS. Pour plus d'informations, consultez ce billet AWS de blog sur la sécurité.

2. Vérifiez la signature

   Les étapes de vérification de votre téléchargement dépendent de la plate-forme du serveur sur lequel vous exécutez l' AWSTOE application après l'avoir installée. Pour vérifier votre téléchargement sur un serveur Linux, consultezVérifiez la signature sous Linux. Pour vérifier votre téléchargement sur un serveur Windows, consultezVérifiez la signature sous Windows.

> **Important**
>
> AWSTOE est invoqué directement depuis son emplacement de téléchargement. Il n'est pas nécessaire d'effectuer une étape d'installation séparée. Cela signifie également que cela AWSTOE peut apporter des modifications à l'environnement local.
> Pour vous assurer d'isoler les modifications lors du développement des composants, nous vous recommandons d'utiliser une instance EC2 pour développer et tester AWSTOE les composants.

Étape 2 : définir les AWS informations d'identification

AWSTOE nécessite des AWS informations d'identification pour se connecter à d'autres sites Services AWS, tels qu'Amazon S3 et Amazon CloudWatch, lors de l'exécution de tâches telles que :

- Téléchargement de AWSTOE documents depuis un chemin Amazon S3 fourni par l'utilisateur.
- Modules `S3Download` d'exécution ou `S3Upload` d'action.
- Le streaming se connecte à CloudWatch, lorsqu'il est activé.

Si vous exécutez AWSTOE sur une instance EC2, l'exécution AWSTOE utilise les mêmes autorisations que le rôle IAM attaché à l'instance EC2.

Pour plus d'informations sur les rôles IAM pour EC2, consultez la section Rôles IAM pour Amazon EC2.

Les exemples suivants montrent comment définir les AWS informations d'identification à l'aide des variables d'`AWS_SECRET_ACCESS_KEY`environnement `AWS_ACCESS_KEY_ID` et.

Pour définir ces variables sous Linux, macOS ou Unix, utilisez`export`.

```
$ export AWS_ACCESS_KEY_ID=your_access_key_id
```

```
$ export AWS_SECRET_ACCESS_KEY=your_secret_access_key
```

Pour définir ces variables sous Windows à l'aide de PowerShell, utilisez`$env`.

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

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

Pour définir ces variables sous Windows à l'aide de l'invite de commande, utilisez`set`.

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

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

Étape 3 : développer les documents des composants localement

AWSTOE les composants sont créés à partir de documents YAML en texte brut. Pour plus d'informations sur la syntaxe des documents, consultezUtiliser les documents relatifs aux composants dans AWSTOE.

Vous trouverez ci-dessous des exemples de documents relatifs aux composants Hello World que vous pouvez utiliser pour développer vos documents localement.

`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.'
```

Étape 4 : Valider AWSTOE les composants

Vous pouvez valider la syntaxe des AWSTOE composants localement avec l' AWSTOE application. Les exemples suivants montrent la `validate` commande AWSTOE d'application permettant de valider la syntaxe d'un composant sans l'exécuter.

> **Note**
>
> L' AWSTOE application ne peut valider que la syntaxe des composants pour le système d'exploitation actuel. Par exemple, lors de l'exécution `awstoe.exe` sous Windows, vous ne pouvez pas valider la syntaxe d'un document Linux qui utilise le module `ExecuteBash` d'action.

Windows

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

Linux

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

Étape 5 : Exécuter AWSTOE les composants

L' AWSTOE application peut exécuter une ou plusieurs phases de documents spécifiés à l'aide de l'argument de ligne de `--phases` commande. Les valeurs prises en charge pour `--phases` sont `buildvalidate`, et`test`. Plusieurs valeurs de phase peuvent être saisies sous forme de valeurs séparées par des virgules.

Lorsque vous fournissez une liste de phases, l' AWSTOE application exécute séquentiellement les phases spécifiées pour chaque document. Par exemple, AWSTOE exécute les `validate` phases `build` et de`document1.yaml`, puis les `validate` phases `build` et de`document2.yaml`.

Pour garantir que vos journaux sont stockés en toute sécurité et conservés à des fins de dépannage, nous vous recommandons de configurer le stockage des journaux dans Amazon S3. Dans Image Builder, l'emplacement Amazon S3 pour la publication des journaux est spécifié dans la configuration de l'infrastructure. Pour plus d'informations sur la configuration de l'infrastructure, voir Gérer la configuration de l'infrastructure EC2 Image Builder

Si aucune liste de phases n'est fournie, l' AWSTOE application exécute toutes les phases dans l'ordre indiqué dans le document YAML.

Pour exécuter des phases spécifiques dans un ou plusieurs documents, utilisez les commandes suivantes.

Monophasé

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

Phases multiples

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

Document exécuté

Exécuter toutes les phases dans un seul document

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

Exécuter toutes les phases dans plusieurs documents

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

Entrez les informations Amazon S3 pour télécharger les AWSTOE journaux à partir d'un chemin local défini par l'utilisateur (recommandé)

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

Exécutez toutes les phases dans un seul document et affichez tous les journaux sur la console

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

Exemple de commande

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

Exécuter un document avec un identifiant unique

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

Obtenez de l'aide pour AWSTOE

```
awstoe --help
```

Utiliser les documents relatifs aux composants dans AWSTOE

Pour créer un composant à l'aide de AWS Task Orchestrator and Executor (AWSTOE), vous devez fournir un document basé sur YAML qui représente les phases et les étapes applicables au composant que vous créez. Services AWS utilisent votre composant lorsqu'ils créent une nouvelle image de machine Amazon (AMI) ou une nouvelle image de conteneur.

Rubriques

- Flux de travail documentaire sur les composants
- Journalisation des composants
- Chainage des entrées et des sorties
- Schéma du document et définitions
- Exemples de schémas de documents
- Définissez et référencez des variables dans AWSTOE

- Utiliser des constructions en boucle dans AWSTOE

Flux de travail documentaire sur les composants

Le document du AWSTOE composant utilise des phases et des étapes pour regrouper les tâches associées et organiser ces tâches dans un flux de travail logique pour le composant.

> **Tip**
>
> Le service qui utilise votre composant pour créer une image peut implémenter des règles concernant les phases à utiliser pour son processus de génération et le moment où ces phases sont autorisées à s'exécuter. Il est important d'en tenir compte lors de la conception de votre composant.

Phases

Les phases représentent la progression de votre flux de travail tout au long du processus de création de l'image. Par exemple, le service Image Builder utilise `build` et met en `validate` phase les images qu'il produit au cours de sa phase de création. Il utilise les `container-host-test` phases `test` et au cours de sa phase de test pour s'assurer que l'instantané d'image ou l'image du conteneur produit les résultats attendus avant de créer l'AMI finale ou de distribuer l'image du conteneur.

Lorsque le composant s'exécute, les commandes associées à chaque phase sont appliquées dans l'ordre dans lequel elles apparaissent dans le document du composant.

Règles pour les phases

- Chaque nom de phase doit être unique dans un document.
- Vous pouvez définir de nombreuses phases dans votre document.
- Vous devez inclure au moins l'une des phases suivantes dans votre document :
  - build — pour Image Builder, cette phase est généralement utilisée pendant la phase de construction.
  - valider — pour Image Builder, cette phase est généralement utilisée pendant la phase de construction.
  - test — pour Image Builder, cette phase est généralement utilisée pendant la phase de test.

- Les phases s'exécutent toujours dans l'ordre dans lequel elles sont définies dans le document. L'ordre dans lequel elles sont spécifiées pour AWSTOE les commandes du n' AWS CLI a aucun effet.

Étapes

Les étapes sont des unités de travail individuelles qui définissent le flux de travail au sein de chaque phase. Les étapes sont exécutées par ordre séquentiel. Cependant, l'entrée ou la sortie d'une étape peuvent également alimenter une étape suivante en tant qu'entrée. C'est ce qu'on appelle le « chaînage ».

Règles relatives aux étapes

- Le nom de l'étape doit être unique pour la phase.
- L'étape doit utiliser une action prise en charge (module d'action) qui renvoie un code de sortie.

  Pour obtenir la liste complète des modules d'action pris en charge, leur fonctionnement, les valeurs d'entrée/sortie et des exemples, voir. Modules d'action pris en charge par le gestionnaire de AWSTOE composants

Journalisation des composants

AWSTOE crée un nouveau dossier journal sur les instances EC2 utilisées pour créer et tester une nouvelle image, chaque fois que votre composant s'exécute. Pour les images de conteneur, le dossier journal est stocké dans le conteneur.

Pour faciliter le dépannage en cas de problème lors du processus de création de l'image, le document d'entrée et tous les fichiers de sortie AWSTOE créés lors de l'exécution du composant sont stockés dans le dossier journal.

Le nom du dossier journal comprend les éléments suivants :

1. Répertoire des journaux : lorsqu'un service exécute un AWSTOE composant, il est transmis dans le répertoire des journaux, ainsi que les autres paramètres de la commande. Dans les exemples suivants, nous montrons le format de fichier journal utilisé par Image Builder.
   - Linux : `/var/lib/amazon/toe/`
   - Windows : `$env:ProgramFiles\Amazon\TaskOrchestratorAndExecutor\`
2. Préfixe de fichier — Il s'agit d'un préfixe standard utilisé pour tous les composants : « »`TOE_`.

3. Durée d'exécution : il s'agit d'un horodatage au format YYYY-MM-DD_HH-MM-SS_UTC-0.
4. ID d'exécution : il s'agit du GUID attribué lors de l' AWSTOE exécution d'un ou de plusieurs composants.

Exemple : */var/lib/amazon/toe/*T0E*_2021-07-01_12-34-56_UTC-0_a1bcd2e3-45f6-789a-bcde-0fa1b2c3def4*

AWSTOE stocke les fichiers principaux suivants dans le dossier journal :

Fichiers d'entrée

- document.yaml — Document utilisé comme entrée pour la commande. Une fois le composant exécuté, ce fichier est stocké sous forme d'artefact.

Fichiers de sortie

- application.log — Le journal de l'application contient des informations horodatées AWSTOE sur le niveau de débogage indiquant ce qui se passe pendant l'exécution du composant.
- detailedoutput.json — Ce fichier JSON contient des informations détaillées sur l'état d'exécution, les entrées, les sorties et les échecs pour tous les documents, phases et étapes applicables au composant lors de son exécution.
- console.log — Le journal de la console contient toutes les informations de sortie standard (stdout) et d'erreur standard (stderr) écrites sur la console pendant l'exécution du composant. AWSTOE
- chaining.json — Ce fichier JSON représente les optimisations AWSTOE appliquées pour résoudre les expressions de chaînage.

> **Note**
>
> Le dossier journal peut également contenir d'autres fichiers temporaires qui ne sont pas abordés ici.

Chainage des entrées et des sorties

L'application AWSTOE de gestion de configuration fournit une fonctionnalité permettant de chaîner les entrées et les sorties en écrivant des références dans les formats suivants :

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

or

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

La fonction de chaînage vous permet de recycler le code et d'améliorer la maintenabilité du document.

Règles relatives au chaînage

- Les expressions de chaînage ne peuvent être utilisées que dans la section des entrées de chaque étape.
- Les instructions contenant des expressions de chaînage doivent être placées entre guillemets. Par exemple :
  - Expression non valide : `echo {{ phase.step.inputs.variable }}`
  - Expression valide : `"echo {{ phase.step.inputs.variable }}"`
  - Expression valide : `'echo {{ phase.step.inputs.variable }}'`
- Les expressions de chaînage peuvent faire référence à des variables issues d'autres étapes et phases du même document. Cependant, le service d'appel peut avoir des règles qui exigent que le chaînage des expressions ne fonctionne que dans le contexte d'une seule étape. Par exemple, Image Builder ne prend pas en charge le chaînage entre la phase de création et la phase de test, car il exécute chaque étape indépendamment.
- Les index des expressions de chaînage suivent une indexation basée sur zéro. L'indice commence par zéro (0) pour référencer le premier élément.

Exemples

Pour faire référence à la variable source dans la deuxième entrée de l'étape d'exemple suivante, le modèle de chaînage est`{{ 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'
```

Pour faire référence à la variable de sortie (égale à « Hello ») de l'étape d'exemple suivante, le modèle de chaînage est`{{ build.SamplePowerShellStep.outputs.stdout }}`.

```
phases:
  -
    name: 'build'
    steps:
      -
        name: SamplePowerShellStep
        action: ExecutePowerShell
        timeoutSeconds: 120
        onFailure: Abort
        maxAttempts: 3
        inputs:
          commands:
            - 'Write-Host "Hello"'
```

Schéma du document et définitions

Le schéma YAML d'un document est le suivant.

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

phases:
  - name: "string"
    steps:
        - name: "string"
          action: "string"
          timeoutSeconds: integer
```

```
        onFailure: "Abort|Continue|Ignore"
        maxAttempts: integer
        inputs:
```

Les définitions de schéma d'un document sont les suivantes.

Champ	Description	Type	Obligatoire
name	Nom du document.	Chaîne	Non
description	Description du document	Chaîne	Non
schemaVersion	Version du schéma du document, actuellement 1.0.	Chaîne	Oui
phases	Une liste des phases avec leurs étapes.	Liste	Oui

Les définitions du schéma d'une phase sont les suivantes.

Champ	Description	Type	Obligatoire
name	Nom de la phase.	Chaîne	Oui
steps	Liste des étapes de la phase.	Liste	Oui

Les définitions du schéma d'une étape sont les suivantes.

Champ	Description	Type	Obligatoire	Valeur par défaut
name	Nom défini par l'utilisateur pour l'étape.	Chaîne		

Champ	Description	Type	Obligatoire	Valeur par défaut
action	Mot-clé relatif au module qui exécute l'étape.	Chaîne		
timeoutSeconds	Nombre de secondes pendant lesquelles l'étape s'exécute avant d'échouer ou de réessayer. Supporte également la valeur -1, ce qui indique un délai d'expirat ion infini. 0 et les autres valeurs négatives ne sont pas autorisées.	Entier	Non	7 200 secondes (120 minutes)

Champ	Description	Type	Obligatoire	Valeur par défaut
onFailure	Spécifie ce que l'étape doit faire en cas d'échec. Les valeurs valides sont les suivantes : • Abandonner — Échoue l'étape après le nombre maximum de tentatives et arrête l'exécution. Définit le statut de la phase et du document à`Failed`. • Continuer — Échoue l'étape après le nombre maximum de tentatives et continue d'exécuter les étapes restantes. Définit le statut de la phase et du document à`Failed`.	Chaîne	Non	Interruption

Champ	Description	Type	Obligatoire	Valeur par défaut
	• Ignorer : définit l'étape `IgnoredFailure` après le nombre maximum de tentatives infructueuses et continue à exécuter les étapes restantes. Définit le statut de la phase et du document à`SuccessWithIgnoredFailure`.			
maxAttempts	Nombre maximum de tentatives autorisées avant d'échouer à l'étape.	Entier	Non	1
inputs	Contient les paramètres requis par le module d'action pour exécuter l'étape.	Dict	Oui	

Exemples de schémas de documents

Voici un exemple de schéma de document pour installer toutes les mises à jour Windows disponibles, exécuter un script de configuration, valider les modifications avant la création de l'AMI et tester les modifications après la création de l'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}}'
```

Voici un exemple de schéma de document pour télécharger et exécuter un fichier binaire Linux personnalisé.

```
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 }}'
```

Voici un exemple de schéma de document pour installer le AWS CLI sur une instance Windows à l'aide du fichier d'installation.

```
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 }}'
```

Voici un exemple de schéma de document pour installer le à l' AWS CLI aide du programme d'installation 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 }}'
```

Définissez et référencez des variables dans AWSTOE

Les variables permettent d'étiqueter les données avec des noms significatifs qui peuvent être utilisés dans l'ensemble d'une application. Vous pouvez définir des variables personnalisées avec des formats simples et lisibles pour les flux de travail complexes, et les référencer dans le document du composant d'application YAML correspondant à un AWSTOE composant.

Cette section fournit des informations pour vous aider à définir les variables de votre AWSTOE composant dans le document du composant d'application YAML, notamment la syntaxe, les contraintes de nom et des exemples.

Paramètres

Les paramètres sont des variables mutables, avec des paramètres que l'application appelante peut fournir au moment de l'exécution. Vous pouvez définir des paramètres dans la `Parameters` section du document YAML.

Règles relatives aux noms de paramètres

- Le nom doit comporter entre 3 et 128 caractères.
- Le nom ne peut contenir que des caractères alphanumériques (a-z, A-Z, 0-9), des tirets (-) ou des traits de soulignement (_).
- Le nom doit être unique dans le document.
- Le nom doit être spécifié sous forme de chaîne YAML.

Syntaxe

```
parameters:
  - <name>:
```

```
      type: <parameter type>
      default: <parameter value>
      description: <parameter description>
```

Nom de la touche	Obligatoire	Description
`name`	Oui	Nom du paramètre. Doit être unique pour le document (il ne doit pas être identique aux autres noms de paramètres ou constantes).
`type`	Oui	Type de données du paramètre. Les types pris en charge incluent :`string`.
`default`	Non	La valeur par défaut du paramètre.
`description`	Non	Décrit le paramètre.

Valeurs des paramètres de référence dans un document

Vous pouvez référencer des paramètres dans des entrées par étapes ou en boucle dans votre document YAML, comme suit :

- Les références aux paramètres distinguent les majuscules et minuscules, et le nom doit correspondre exactement.
- Le nom doit être placé entre accolades `{{ MyParameter}}`doubles.
- Les espaces sont autorisés à l'intérieur des bretelles bouclées et sont automatiquement rognés. Par exemple, toutes les références suivantes sont valides :

  `{{ MyParameter }}`, `{{ MyParameter}}`, `{{MyParameter }}`, `{{MyParameter}}`
- La référence dans le document YAML doit être spécifiée sous forme de chaîne (entre guillemets simples ou doubles).

  Par exemple : n'`- {{ MyParameter }}`est pas valide, car il n'est pas identifié comme une chaîne.

Toutefois, les références suivantes sont toutes deux valides : - '{{ *MyParameter* }}' et- "{{ *MyParameter* }}".

Exemples

Les exemples suivants montrent comment utiliser des paramètres dans votre document YAML :

- Référez-vous à un paramètre dans les entrées d'étape :

```
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'
```

- Référez-vous à un paramètre dans les entrées de boucle :

```
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 }}
```

Remplacer les paramètres lors de l'exécution

Vous pouvez utiliser l'`--parameters`option AWS CLI avec une paire clé-valeur pour définir une valeur de paramètre lors de l'exécution.

- <value>Spécifiez la paire clé-valeur du paramètre sous forme de nom et de valeur, séparés par un signe égal (<name>=).
- Les paramètres multiples doivent être séparés par une virgule.
- Les noms de paramètres introuvables dans le document du composant YAML sont ignorés.
- Le nom et la valeur du paramètre sont tous deux obligatoires.

> **Important**
>
> Les paramètres des composants sont des valeurs en texte brut et sont connectés AWS CloudTrail. Nous vous recommandons d'utiliser AWS Secrets Manager le AWS Systems Manager Parameter Store pour stocker vos secrets. Pour plus d'informations sur Secrets Manager, voir Qu'est-ce que Secrets Manager ? dans le guide de AWS Secrets Manager l'utilisateur. Pour plus d'informations sur AWS Systems Manager Parameter Store, voir AWS Systems Manager Parameter Store dans le guide de AWS Systems Manager l'utilisateur.

Syntaxe

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

Option CLI	Obligatoire	Description
--nom des paramètres = *valeur*,...	Non	Cette option prend une liste de paires clé-valeur, avec le nom du paramètre comme clé.

Exemples

Les exemples suivants montrent comment utiliser des paramètres dans votre document YAML :

- La paire clé-valeur du paramètre spécifiée dans cette `--parameter` option n'est pas valide :

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

- Définissez une paire clé-valeur de paramètre avec l'`--parameter`option dans le champ : AWS CLI

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

- Définissez plusieurs paires clé-valeur de paramètres à l'aide de l'`--parameter`option suivante : AWS CLI

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

Constantes

Les constantes sont des variables immuables qui ne peuvent pas être modifiées ou remplacées une fois définies. Les constantes peuvent être définies à l'aide des valeurs figurant dans la `constants` section d'un AWSTOE document.

Règles pour les noms de constantes

- Le nom doit comporter entre 3 et 128 caractères.
- Le nom ne peut contenir que des caractères alphanumériques (a-z, A-Z, 0-9), des tirets (-) ou des traits de soulignement (_).
- Le nom doit être unique dans le document.
- Le nom doit être spécifié sous forme de chaîne YAML.

Syntaxe

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

```
      value: <constant value>
```

Nom de la touche	Obligatoire	Description
`name`	Oui	Nom de la constante. Doit être unique pour le document (il ne doit pas être identique aux autres noms de paramètres ou constantes).
`value`	Oui	Valeur de la constante.
`type`	Oui	Type de constante. Le type pris en charge est`string`.

Valeurs constantes de référence dans un document

Vous pouvez référencer des constantes dans les entrées d'étape ou de boucle à l'intérieur de votre document YAML, comme suit :

- Les références constantes distinguent les majuscules et minuscules, et le nom doit correspondre exactement.
- Le nom doit être placé entre accolades `{{ MyConstant}}`doubles.
- Les espaces sont autorisés à l'intérieur des bretelles bouclées et sont automatiquement rognés. Par exemple, toutes les références suivantes sont valides :

  `{{ MyConstant }}`, `{{ MyConstant}}`, `{{MyConstant }}`, `{{MyConstant}}`
- La référence dans le document YAML doit être spécifiée sous forme de chaîne (entre guillemets simples ou doubles).

  Par exemple : n'`- {{ MyConstant }}`est pas valide, car il n'est pas identifié comme une chaîne.

  Toutefois, les références suivantes sont toutes deux valides : `- '{{ MyConstant }}'` et`- "{{ MyConstant }}"`.

Exemples

Constante référencée dans les entrées d'étape

```
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 référencée dans les entrées de boucle

```
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 }}
```

Utiliser des constructions en boucle dans AWSTOE

Cette section fournit des informations pour vous aider à créer des constructions en boucle dans le. AWSTOE Les constructions en boucle définissent une séquence répétée d'instructions. Vous pouvez utiliser les types de constructions en boucle suivants dans : AWSTOE

- `for`constructs — Itérer sur une séquence bornée d'entiers.
- `forEach`construit
  - `forEach`boucle avec liste d'entrées : itère sur une collection limitée de chaînes.
  - `forEach`boucle avec liste délimitée : itère sur une collection limitée de chaînes jointes par un délimiteur.

> **Note**
>
> Les constructions en boucle ne prennent en charge que les types de données sous forme de chaîne.

Sujets de construction en boucle

- Variables d'itération de référence
- Types de constructions en boucle
- Champs d'étape
- Sorties d'étape et d'itération

Variables d'itération de référence

Pour faire référence à l'index et à la valeur de la variable d'itération en cours, l'expression de référence `{{ loop.* }}` doit être utilisée dans le corps d'entrée d'une étape contenant une construction en boucle. Cette expression ne peut pas être utilisée pour faire référence aux variables d'itération de la construction en boucle d'une autre étape.

L'expression de référence comprend les membres suivants :

- `{{ loop.index }}`— Position ordinale de l'itération en cours, qui est indexée à. `0`
- `{{ loop.value }}`— La valeur associée à la variable d'itération en cours.

Noms des boucles

Toutes les constructions en boucle ont un champ de nom facultatif pour l'identification. Si un nom de boucle est fourni, il peut être utilisé pour faire référence à des variables d'itération dans le corps d'entrée de l'étape. Pour faire référence aux indices d'itération et aux valeurs d'une boucle nommée,

utilisez `{{ <loop_name>.* }}` with `{{ loop.* }}` dans le corps de saisie de l'étape. Cette expression ne peut pas être utilisée pour faire référence à la construction en boucle nommée d'une autre étape.

L'expression de référence comprend les membres suivants :

- `{{ <loop_name>.index }}`— Position ordinale de l'itération en cours de la boucle nommée, qui est indexée à. `0`
- `{{ <loop_name>.value }}`— La valeur associée à la variable d'itération actuelle de la boucle nommée.

Résoudre les expressions de référence

AWSTOE Résout les expressions de référence comme suit :

- `{{ <loop_name>.* }}`— AWSTOE résout cette expression selon la logique suivante :
  - Si la boucle de l'étape en cours d'exécution correspond à la `<loop_name>` valeur, l'expression de référence est résolue en fonction de la structure en boucle de l'étape en cours d'exécution.
  - `<loop_name>`aboutit à la construction en boucle nommée si elle apparaît dans l'étape en cours d'exécution.
- `{{ loop.* }}`— AWSTOE résout l'expression à l'aide de la construction en boucle définie dans l'étape en cours d'exécution.

Si des expressions de référence sont utilisées dans une étape ne contenant pas de boucle, les expressions AWSTOE ne sont pas résolues et elles apparaissent dans l'étape sans être remplacées.

> **Note**
>
> Les expressions de référence doivent être placées entre guillemets pour être correctement interprétées par le compilateur YAML.

Types de constructions en boucle

Cette section fournit des informations et des exemples sur les types de construction en boucle qui peuvent être utilisés dans le AWSTOE.

Types de constructions en boucle

- forboucle
- forEachboucle avec liste d'entrées
- forEachboucle avec liste délimitée

forboucle

La `for` boucle itère sur une plage d'entiers spécifiée dans une limite définie par le début et la fin des variables. Les valeurs itératives font partie de l'ensemble `[start, end]` et incluent les valeurs limites.

AWSTOE vérifie les `updateBy` valeurs `start``end`, et pour s'assurer que la combinaison ne donne pas lieu à une boucle infinie.

`for`schéma de boucle

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

`for`entrée en boucle

Champ	Description	Type	Obligatoire	Par défaut
`name`	Nom unique de la boucle. Il doit être unique par rapport aux autres noms de boucle de la même phase.	Chaîne	Non	""
`start`	Valeur de départ de l'itération.	Entier	Oui	N/A

Champ	Description	Type	Obligatoire	Par défaut
	N'accepte pas le chaînage d'expressions.			
`end`	Valeur finale de l'itération. N'accepte pas le chaînage d'expressions.	Entier	Oui	N/A
`updateBy`	Différence selon laquelle une valeur itérative est mise à jour par addition. Il doit s'agir d'une valeur négative ou positive différente de zéro. N'accepte pas le chaînage d'expressions.	Entier	Oui	N/A

`for`exemple d'entrée en boucle

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

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

`forEach`boucle avec liste d'entrées

La `forEach` boucle itère sur une liste explicite de valeurs, qui peuvent être des chaînes ou des expressions enchaînées.

`forEach`boucle avec schéma de liste d'entrées

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

`forEach`boucle avec entrée de liste d'entrées

Champ	Description	Type	Obligatoire	Par défaut
`name`	Nom unique de la boucle. Il doit être unique par rapport aux autres noms de boucle de la même phase.	Chaîne	Non	""
Liste des chaînes de	Liste des chaînes pour l'itération.	Liste de chaînes	Oui	N/A

Champ	Description	Type	Obligatoire	Par défaut
`forEach` boucles	Accepte les expressions chaînées sous forme de chaînes dans la liste. Les expressions chaînées doivent être placées entre guillemets pour que le compilateur YAML puisse les interpréter correctement.			

`forEach`boucle avec liste d'entrées, exemple 1

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

`forEach`boucle avec liste d'entrées, exemple 2

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

`forEach`exemple de boucle avec liste d'entrées 3

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

`forEach`boucle avec liste délimitée

La boucle itère sur une chaîne contenant des valeurs séparées par un délimiteur. Pour itérer sur les constituants de la chaîne, utilisez AWSTOE le délimiteur pour diviser la chaîne en un tableau adapté à l'itération.

`forEach`boucle avec schéma de liste délimité

```
name: "StepName"
action: "ActionModule"
loop:
  name: "string"
```

```
  forEach:
    list: "string"
    delimiter: ".,;:\n\t -_"
inputs:
  ...
```

`forEach`boucle avec entrée de liste délimitée

Champ	Description	Type	Obligatoire	Par défaut
`name`	Nom unique attribué à la boucle. Il doit être unique par rapport aux autres noms de boucles de la même phase.	Chaîne	Non	""
`list`	Chaîne composée de chaînes constitutives reliées par un caractère délimiteur commun. Accepte également les expressions chaînées. Dans le cas d'expressions chaînées, assurez-vous qu'elles sont placées entre guillemets pour une interprétation correcte	Chaîne	Oui	N/A

Champ	Description	Type	Obligatoire	Par défaut
	par le compilateur YAML.			
`delimiter`	Caractère utilisé pour séparer les chaînes d'un bloc. La virgule est la virgule par défaut. Un seul caractère délimiteur est autorisé dans la liste donnée : • Point : `"."` • Virgule : `","` • Point-virgule : `";"` • Colon : `":"` • Nouvelle gamme : `"\n"` • Onglet : `"\t"` • Espace : `" "` • Trait d'union : `"-"` • Souligner : `"_"` Les expressions de chaînage ne peuvent pas être utilisées.	Chaîne	Non	Virgule : `","`

> **Note**
>
> La valeur de `list` est traitée comme une chaîne immuable. Si la source de `list` est modifiée pendant l'exécution, elle ne sera pas reflétée pendant l'exécution.

`forEach`boucle avec liste délimitée, exemple 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 }}"
```

`forEach`boucle avec liste délimitée, exemple 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 }}"
```

Champs d'étape

Les boucles font partie d'une étape. Aucun champ lié à l'exécution d'une étape n'est appliqué aux itérations individuelles. Les champs d'étape s'appliquent uniquement au niveau de l'étape, comme suit :

- TimeoutSeconds — Toutes les itérations de la boucle doivent être exécutées dans le délai spécifié par ce champ. Si le délai d'exécution de la boucle expire, AWSTOE exécute la politique de

nouvelle tentative de l'étape et réinitialise le paramètre de délai d'expiration pour chaque nouvelle tentative. Si le délai d'exécution de la boucle dépasse le délai d'expiration après avoir atteint le nombre maximal de tentatives, le message d'échec de l'étape indique que le délai d'exécution de la boucle a expiré.

- OnFailure — La gestion des défaillances est appliquée à l'étape comme suit :
  - Si OnFailure est défini sur`Abort`, AWSTOE quitte la boucle et réessaie l'étape conformément à la politique de nouvelles tentatives. Une fois le nombre maximal de tentatives atteint, AWSTOE marque l'étape en cours comme ayant échoué et arrête l'exécution du processus.

    AWSTOE définit le code d'état de la phase parent et du document sur`Failed`.

    > **Note**
    >
    > Aucune autre étape n'est exécutée après l'échec de l'étape.

  - Si OnFailure est défini sur`Continue`, AWSTOE quitte la boucle et réessaie l'étape conformément à la politique de nouvelles tentatives. Une fois le nombre maximal de tentatives atteint, AWSTOE marque l'étape en cours comme ayant échoué et passe à l'étape suivante.

    AWSTOE définit le code d'état de la phase parent et du document sur`Failed`.

  - Si OnFailure est défini sur`Ignore`, AWSTOE quitte la boucle et réessaie l'étape conformément à la politique de nouvelles tentatives. Une fois le nombre maximal de tentatives atteint, AWSTOE marque l'étape en cours comme telle `IgnoredFailure` et passe à l'étape suivante.

    AWSTOE définit le code d'état de la phase parent et du document sur`SuccessWithIgnoredFailure`.

    > **Note**
    >
    > Ceci est toujours considéré comme une exécution réussie, mais inclut des informations vous indiquant qu'une ou plusieurs étapes ont échoué et ont été ignorées.

- MaxAttempts — À chaque nouvelle tentative, l'étape complète et toutes les itérations sont exécutées depuis le début.
- status — État général de l'exécution d'une étape. `status`ne représente pas le statut des itérations individuelles. Le statut d'une étape comportant des boucles est déterminé comme suit :
  - Si une seule itération échoue, le statut d'une étape indique un échec.

  - Si toutes les itérations sont couronnées de succès, le statut d'une étape indique un succès.
- StartTime : heure de début globale de l'exécution d'une étape. Ne représente pas l'heure de début des itérations individuelles.
- EndTime — Heure de fin globale de l'exécution d'une étape. Ne représente pas l'heure de fin des itérations individuelles.
- FailureMessage — Inclut les indices d'itération qui ont échoué en cas d'erreurs autres que le délai d'expiration. En cas d'erreur de temporisation, le message indique que l'exécution de la boucle a échoué. Les messages d'erreur individuels pour chaque itération ne sont pas fournis afin de minimiser la taille des messages d'échec.

Sorties d'étape et d'itération

Chaque itération contient une sortie. À la fin d'une exécution en boucle, AWSTOE consolide toutes les sorties d'itération réussies dans. `detailedOutput.json` Les sorties consolidées sont un assemblage de valeurs appartenant aux clés de sortie correspondantes telles que définies dans le schéma de sortie du module d'action. L'exemple suivant montre comment les sorties sont consolidées :

Sortie de **`ExecuteBash`** pour l'itération 1

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

Sortie de **`ExecuteBash`** pour l'itération 2

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

Sortie de **`ExecuteBash`** for Step

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

Par exemple, `ExecuteBash`ExecutePowerShell, et `ExecuteBinary` sont des modules d'action qui renvoient `STDOUT` en tant que sortie du module d'action. `STDOUT`les messages sont joints au nouveau caractère de ligne pour produire le résultat global de l'étape d'introduction`detailedOutput.json`.

AWSTOE ne consolidera pas les résultats des itérations infructueuses.

Modules d'action pris en charge par le gestionnaire de AWSTOE composants

Les services de création d'images, tels que EC2 Image Builder, AWSTOE utilisent des modules d'action pour configurer les instances EC2 utilisées pour créer et tester des images de machine personnalisées. Cette section décrit les fonctionnalités des modules d' AWSTOE action couramment utilisés et explique comment les configurer, y compris des exemples.

AWSTOE les composants sont créés à partir de documents YAML en texte brut. Pour plus d'informations sur la syntaxe des documents, consultezUtiliser les documents relatifs aux composants dans AWSTOE.

> **Note**
>
> Tous les modules d'action utilisent le même compte que l'agent Systems Manager lorsqu'ils s'exécutent, sous Linux et `NT Authority\SYSTEM` sous Windows. `root`

Types de modules d'action

- Modules d'exécution généraux
- Modules de téléchargement et de téléversement de fichiers
- Modules de fonctionnement du système de fichiers
- Actions d'installation du logiciel
- Modules d'action du système

Modules d'exécution généraux

La section suivante contient des détails sur les modules d'action qui exécutent des commandes et des instructions d'exécution générales.

Modules d'action d'exécution générale

- ExecuteBash
- ExecuteBinary
- ExecuteDocument

- ExecutePowerShell

ExecuteBash

Le module ExecuteBashd'action vous permet d'exécuter des scripts bash avec du code/des commandes shell intégrés. Ce module est compatible avec Linux.

Toutes les commandes et instructions que vous spécifiez dans le bloc de commandes sont converties dans un fichier (par exemple`input.sh`) et exécutées avec le shell bash. Le résultat de l'exécution du fichier shell est le code de sortie de l'étape.

Le ExecuteBashmodule gère les redémarrages du système si le script se termine avec un code de sortie de. `194` Lorsqu'elle est lancée, l'application exécute l'une des actions suivantes :

- L'application transmet le code de sortie à l'appelant s'il est exécuté par l'agent Systems Manager. L'agent Systems Manager gère le redémarrage du système et exécute la même étape que celle à l'origine du redémarrage, comme décrit dans la section Redémarrage d'une instance gérée à partir de scripts.
- L'application enregistre le courant`executionstate`, configure un déclencheur de redémarrage pour réexécuter l'application et redémarre le système.

Après le redémarrage du système, l'application exécute la même étape que celle à l'origine du redémarrage. Si vous avez besoin de cette fonctionnalité, vous devez écrire des scripts idempotents capables de gérer plusieurs invocations de la même commande shell.

Entrée

Primitif	Description	Type	Obligatoire
`commands`	Contient une liste d'instructions ou de commandes à exécuter conformément à la syntaxe bash. Le YAML multiligne est autorisé.	Liste	Oui

Exemple de saisie : avant et après un redémarrage

```
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
```

Sortie

Champ	Description	Type
`stdout`	Sortie standard de l'exécution des commandes.	chaîne

Si vous lancez un redémarrage et que vous renvoyez le code de sortie dans le 194 cadre du module d'action, la compilation reprendra à l'étape du module d'action qui a initié le redémarrage. Si vous démarrez un redémarrage sans le code de sortie, le processus de génération risque d'échouer.

Exemple de sortie : avant le redémarrage (première fois via le document)

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

Exemple de sortie : après le redémarrage, (deuxième fois dans le document)

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

ExecuteBinary

Le module ExecuteBinaryd'action vous permet d'exécuter des fichiers binaires avec une liste d'arguments de ligne de commande.

Le ExecuteBinarymodule gère les redémarrages du système si le fichier binaire se termine avec un code de sortie `194` (Linux) ou `3010` (Windows). Dans ce cas, l'application exécute l'une des actions suivantes :

- L'application transmet le code de sortie à l'appelant s'il est exécuté par l'agent Systems Manager. L'agent Systems Manager gère le redémarrage du système et exécute la même étape que celle qui a initié le redémarrage, comme décrit dans Redémarrage d'une instance gérée à partir de scripts.
- L'application enregistre le courant`executionstate`, configure un déclencheur de redémarrage pour réexécuter l'application et redémarre le système.

Après le redémarrage du système, l'application exécute la même étape que celle à l'origine du redémarrage. Si vous avez besoin de cette fonctionnalité, vous devez écrire des scripts idempotents capables de gérer plusieurs invocations de la même commande shell.

Entrée

Primitif	Description	Type	Obligatoire
`path`	Le chemin d'accès au fichier binaire à exécuter.	Chaîne	Oui
`arguments`	Contient une liste d'arguments de ligne de commande à utiliser lors de l'exécution du binaire.	Liste de chaînes	Non

Exemple de saisie : installer .NET

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

Sortie

Champ	Description	Type
`stdout`	Sortie standard de l'exécution des commandes.	chaîne

Exemple de sortie

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

ExecuteDocument

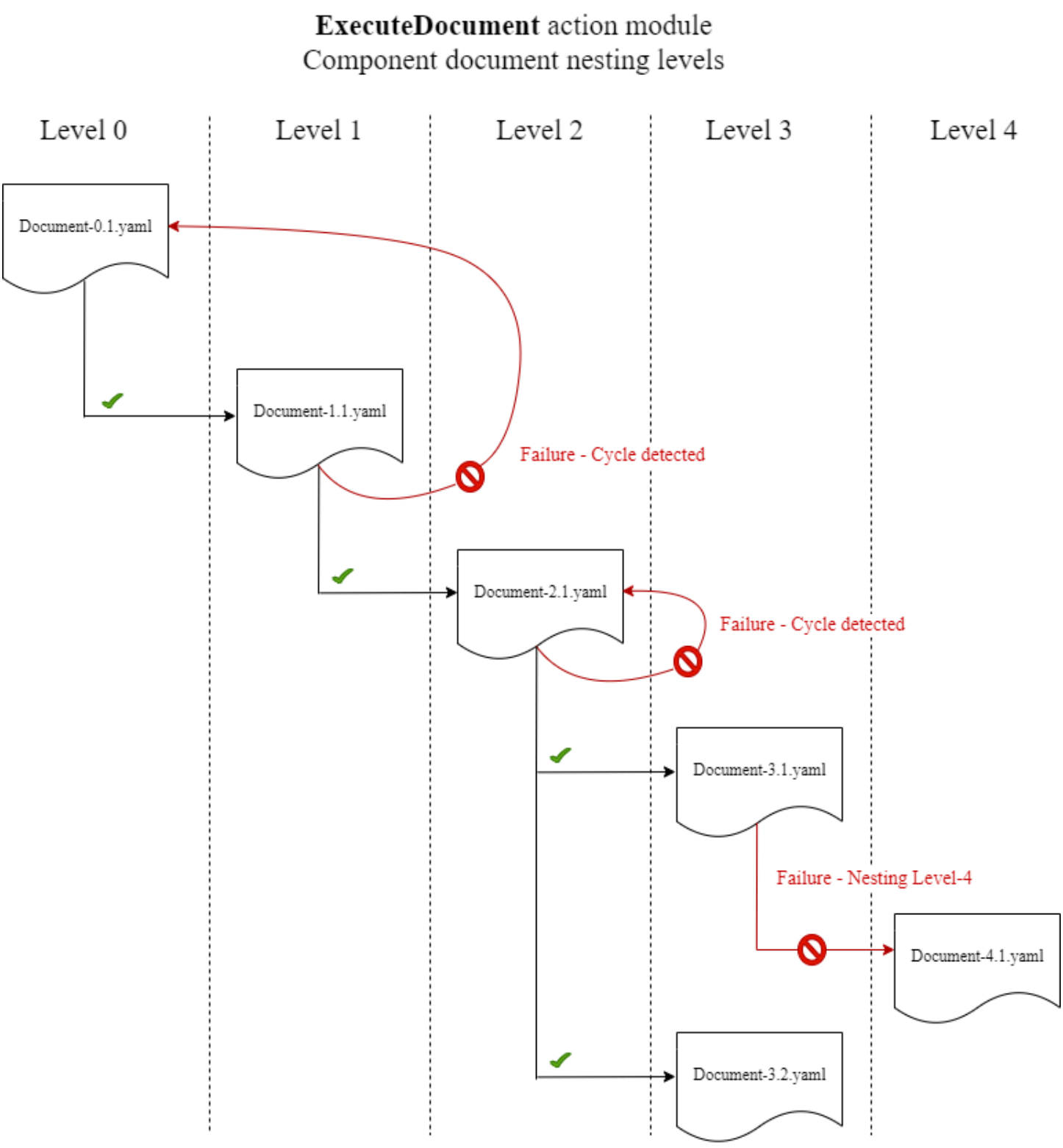
Le module ExecuteDocumentd'action ajoute la prise en charge des documents de composants imbriqués, en exécutant plusieurs documents de composants à partir d'un seul document. AWSTOE valide le document transmis dans le paramètre d'entrée lors de l'exécution.

Restrictions

- Ce module d'action ne s'exécute qu'une seule fois, sans qu'aucune nouvelle tentative ne soit autorisée et qu'aucune option ne permet de définir des limites de délai d'expiration. ExecuteDocumentdéfinit les valeurs par défaut suivantes et renvoie une erreur si vous essayez de les modifier.
  - `timeoutSeconds`: -1
  - `maxAttempts` : 1

> **Note**
>
> Vous pouvez laisser ces valeurs vides et AWSTOE utiliser les valeurs par défaut.

- L'imbrication de documents est autorisée, jusqu'à trois niveaux de profondeur, mais pas plus. Trois niveaux d'imbrication se traduisent par quatre niveaux de document, le niveau supérieur n'étant pas imbriqué. Dans ce scénario, le document de niveau le plus bas ne doit appeler aucun autre document.
- L'exécution cyclique des documents des composants n'est pas autorisée. Tout document qui s'appelle lui-même en dehors d'une construction en boucle, ou qui appelle un autre document situé plus haut dans la chaîne d'exécution actuelle, lance un cycle qui peut entraîner une boucle sans fin. Lorsqu'il AWSTOE détecte une exécution cyclique, il arrête l'exécution et enregistre l'échec.



Si un document de composant essaie de s'exécuter lui-même ou d'exécuter l'un des documents de composant situés plus haut dans la chaîne d'exécution en cours, l'exécution échoue.

Entrée

Primitif	Description	Type	Obligatoire
`document`	Chemin du document du composant. Les options valides sont les suivantes : • Chemins de fichiers locaux • URI S3 • ARN de la version de build du composant EC2 Image Builder	Chaîne	Oui
`document-s3-bucket-owner`	ID de compte du propriétaire du compartiment S3 pour le compartiment S3 dans lequel les documents des composants sont stockés. (Recommandé si vous utilisez des URI S3 dans le document de votre composant.)	Chaîne	Non
`phases`	Phases à exécuter dans le document du composant, exprimées sous forme de liste séparée par des virgules. Si aucune phase n'est	Chaîne	Non

Primitif	Description	Type	Obligatoire
	spécifiée, toutes les phases sont exécutées.		
`parameters`	Paramètres d'entrée transmis au document du composant lors de l'exécution sous forme de paires clé-valeur.	Liste des cartes de paramètres	Non

Entrée de mappage de paramètres

Primitif	Description	Type	Obligatoire
`name`	Nom du paramètre d'entrée à transmett re au document du composant que le module ExecuteDo cumentd'action exécute.	Chaîne	Oui
`value`	La valeur du paramètre d'entrée.	Chaîne	Oui

Exemples de saisie

Les exemples suivants montrent les variations des entrées pour le document de votre composant, en fonction de votre chemin d'installation.

Exemple de saisie : chemin du document 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
```

Exemple de saisie : URI S3 en tant que chemin de document

```
# 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
```

Exemple de saisie : ARN du composant EC2 Image Builder en tant que chemin de document

```
# 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
```

Utiliser une ForEach boucle pour exécuter des documents

```
# 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
```

Utilisation d'une boucle For pour exécuter des documents

```
# 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
```

Sortie

AWSTOE crée un fichier de sortie appelé à `detailedoutput.json` chaque fois qu'il s'exécute. Le fichier contient des détails sur chaque phase et étape de chaque document composant invoqué pendant son exécution. Pour le module ExecuteDocumentd'action, vous trouverez un bref résumé de l'exécution dans le `outputs` champ, ainsi que des détails sur les phases, les étapes et les documents qu'il exécute dans le`detailedOutput`.

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

L'objet récapitulatif de sortie de chaque document de composant contient les détails suivants, comme indiqué ici, avec des exemples de valeurs :

- executedStepCount« :1
- « ID d'exécution » « 12345a67-89bc-01de-2f34-abcd56789012 »
- failedStepCount« 0 : 0
- « Message d'échec » : « »
- « ignoredFailedStep Compter » : 0
- « URL du journal » : « »
- « statut » « succès »

Exemple de sortie

L'exemple suivant montre la sortie du module ExecuteDocumentd'action lorsqu'une exécution imbriquée se produit. Dans cet exemple, le document du `main.yaml` composant exécute correctement le document du `Sample-1.yaml` composant.


```
{
    "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

Le module ExecutePowerShelld'action vous permet d'exécuter des PowerShell scripts avec du code/des commandes shell intégrés. Ce module prend en charge la plate-forme Windows et Windows PowerShell.

Toutes les commandes/instructions spécifiées dans le bloc de commandes sont converties en un fichier de script (par exemple,`input.ps1`) et exécutées sous Windows. PowerShell Le résultat de l'exécution du fichier shell est le code de sortie.

Le ExecutePowerShellmodule gère les redémarrages du système si la commande shell se termine avec un code de sortie de. `3010` Lorsqu'elle est lancée, l'application exécute l'une des actions suivantes :

- Transmet le code de sortie à l'appelant s'il est exécuté par l'agent Systems Manager. L'agent Systems Manager gère le redémarrage du système et exécute la même étape que celle à l'origine du redémarrage, comme décrit dans la section Redémarrage d'une instance gérée à partir de scripts.
- Enregistre le courant`executionstate`, configure un déclencheur de redémarrage pour réexécuter l'application et redémarre le système.

Après le redémarrage du système, l'application exécute la même étape que celle à l'origine du redémarrage. Si vous avez besoin de cette fonctionnalité, vous devez écrire des scripts idempotents capables de gérer plusieurs invocations de la même commande shell.

Entrée

Primitif	Description	Type	Obligatoire
`commands`	Contient une liste d'instructions ou de commandes à exécuter conformément à PowerShell la syntaxe. Le YAML multiligne est autorisé.	Liste de chaînes	Oui. Doit spécifier `commands` ou `file` pas les deux.
`file`	Contient le chemin d'accès à un fichier de PowerShell script. PowerShell sera exécuté sur ce fichier en utilisant l'argument de ligne de `-file` commande. Le chemin doit pointer vers un `.ps1` fichier.	Chaîne	Oui. Doit spécifier `commands` ou `file` pas les deux.

Exemple de saisie : avant et après un redémarrage

```
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)
```

Sortie

Champ	Description	Type
`stdout`	Sortie standard de l'exécution des commandes.	chaîne

Si vous exécutez un redémarrage et que vous renvoyez le code de sortie dans le `3010` cadre du module d'action, la compilation reprendra à l'étape du module d'action qui a initié le redémarrage. Si vous redémarrez sans le code de sortie, le processus de génération risque d'échouer.

Exemple de sortie : avant le redémarrage (première fois via le document)

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

Exemple de sortie : après le redémarrage, (deuxième fois dans le document)

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

Modules de téléchargement et de téléversement de fichiers

La section suivante contient des informations détaillées sur les modules d'action qui exécutent les commandes et instructions de téléchargement et de chargement.

Téléchargez et chargez des modules d'action

- Téléchargement du S3
- Téléchargement S3
- WebDownload

Téléchargement du S3

Avec le module `S3Download` d'action, vous pouvez télécharger un objet Amazon S3, ou un ensemble d'objets, dans un fichier ou un dossier local que vous spécifiez avec le `destination` chemin. Si un fichier existe déjà à l'emplacement spécifié et que l'`overwrite`indicateur est défini sur true, le fichier `S3Download` est remplacé.

Votre `source` position peut pointer vers un objet spécifique dans Amazon S3, ou vous pouvez utiliser un préfixe de clé avec un astérisque (*) pour télécharger un ensemble d'objets correspondant au chemin du préfixe clé. Lorsque vous spécifiez un préfixe clé dans votre `source` position, le module `S3Download` d'action télécharge tout ce qui correspond au préfixe (fichiers et dossiers inclus). Assurez-vous que le préfixe clé se termine par une barre oblique, suivie d'un astérisque (`/*`), afin de télécharger tout ce qui correspond au préfixe. Par exemple : *`s3://my-bucket/my-folder/*`*.

> **Note**
>
> Tous les dossiers du chemin de destination doivent exister avant le téléchargement, sinon le téléchargement échoue.

Si l'`S3Download`action pour un préfixe de clé spécifié échoue lors d'un téléchargement, le contenu du dossier n'est pas rétabli dans son état antérieur à l'échec. Le dossier de destination reste tel qu'il était au moment de l'échec.

Cas d'utilisation pris en charge

Le module `S3Download` d'action prend en charge les cas d'utilisation suivants :

- L'objet Amazon S3 est téléchargé dans un dossier local, comme indiqué dans le chemin de téléchargement.
- Les objets Amazon S3 (avec un préfixe clé dans le chemin du fichier Amazon S3) sont téléchargés dans le dossier local spécifié, qui copie de manière récursive tous les objets Amazon S3 correspondant au préfixe clé dans le dossier local.

Exigences relatives à l'IAM

Le rôle IAM que vous associez à votre profil d'instance doit être autorisé à exécuter le module `S3Download` d'action. Les politiques IAM suivantes doivent être associées au rôle IAM associé au profil d'instance :

- Fichier unique : `s3:GetObject` contre le bucket ou l'objet (par exemple,`arn:aws:s3:::BucketName/*`).
- Plusieurs fichiers : `s3:ListBucket` contre le bucket ou l'objet (par exemple,`arn:aws:s3:::BucketName`) et `s3:GetObject` contre le bucket ou l'objet (par exemple,). `arn:aws:s3:::BucketName/*`

Entrée

Primitif	Description	Type	Obligatoire	Par défaut
`source`	Le compartiment Amazon S3 qui est la source de votre téléchargement. Vous pouvez spécifier le chemin d'accès à un objet spécifique ou utiliser un préfixe de clé se terminant par une barre oblique suivie d'un astérisque (`/*`) pour télécharger un ensemble d'objets correspondant au préfixe de clé.	Chaîne	Oui	N/A

Primitif	Description	Type	Obligatoire	Par défaut
`destination`	Le chemin local où les objets Amazon S3 sont téléchargés. Pour télécharger un seul fichier, vous devez spécifier le nom du fichier dans le chemin. Par exemple, */myfolder/package.zip* .	Chaîne	Oui	N/A
`expectedBucketOwner`	ID de compte propriétaire attendu du bucket indiqué dans le `source` chemin. Nous vous recommandons de vérifier la propriété du compartiment Amazon S3 spécifié dans la source.	Chaîne	Non	N/A

Primitif	Description	Type	Obligatoire	Par défaut
`overwrite`	Lorsqu'il est défini sur true, si un fichier du même nom existe déjà dans le dossier de destination pour le chemin local spécifié, le fichier de télécharg ement remplace le fichier local. Lorsqu'il est défini sur false, le fichier existant sur le système local est protégé contre le remplacem ent et le module d'action échoue en raison d'une erreur de téléchargement. Par exemple, `Error: S3Download: File already exists and "overwrit e" prope rty for "destinat ion" file`	Booléen	Non	true

Primitif	Description	Type	Obligatoire	Par défaut
	is set to false. Cannot download.			

> **Note**
>
> Dans les exemples suivants, le chemin du dossier Windows peut être remplacé par un chemin Linux. Par exemple, *C:\myfolder\package.zip* peut être remplacé par*/myfolder/package.zip*.

Exemple de saisie : copier un objet Amazon S3 dans un fichier local

L'exemple suivant montre comment copier un objet Amazon S3 dans un fichier 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
```

Exemple de saisie : copier tous les objets Amazon S3 dans un compartiment Amazon S3 avec le préfixe key dans un dossier local

L'exemple suivant montre comment copier tous les objets Amazon S3 d'un compartiment Amazon S3 avec le préfixe key dans un dossier local. Amazon S3 n'a aucune notion de dossier. Par conséquent, tous les objets correspondant au préfixe de clé sont copiés. Le nombre maximum d'objets pouvant être téléchargés est de 1 000.

```
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
```

Sortie

Aucune.

Téléchargement S3

Avec le module d'action S3Upload, vous pouvez télécharger un fichier depuis un fichier ou un dossier source vers un emplacement Amazon S3. Vous pouvez utiliser un caractère générique (*) dans le chemin indiqué pour votre emplacement source afin de télécharger tous les fichiers dont le chemin correspond au modèle générique.

Si l'action récursive S3Upload échoue, tous les fichiers déjà chargés resteront dans le compartiment Amazon S3 de destination.

Cas d'utilisation pris en charge

- Fichier local vers un objet Amazon S3.
- Fichiers locaux dans un dossier (avec caractère générique) avec le préfixe de clé Amazon S3.
- Copiez le dossier local (doit avoir été `recurse` défini sur`true`) dans le préfixe de clé Amazon S3.

Exigences relatives à l'IAM

Le rôle IAM que vous associez à votre profil d'instance doit être autorisé à exécuter le module `S3Upload` d'action. La politique IAM suivante doit être attachée au rôle IAM associé au profil

d'instance. La politique doit accorder `s3:PutObject` des autorisations au compartiment Amazon S3 cible. Par exemple, `arn:aws:s3:::BucketName/*`).

Entrée

Primitif	Description	Type	Obligatoire	Par défaut
`source`	Le chemin local d'où proviennent les fichiers/dossiers source. `source`Supporte un caractère générique astérisque (`*`).	Chaîne	Oui	N/A
`destination`	Le chemin du compartiment Amazon S3 de destination où les fichiers/dossiers source sont chargés.	Chaîne	Oui	N/A
`recurse`	Lorsqu'il est défini sur`true`, exécute S3Upload de manière récursive.	Chaîne	Non	`false`
`expectedBucketOwner`	L'ID de compte propriétaire attendu pour le compartiment Amazon S3 spécifié dans le chemin de destinati	Chaîne	Non	N/A

Primitif	Description	Type	Obligatoire	Par défaut
	on. Nous vous recommandons de vérifier la propriété du compartiment Amazon S3 spécifié dans la destination.			

Exemple de saisie : copie d'un fichier local dans un objet Amazon S3

L'exemple suivant montre comment copier un fichier local dans un objet Amazon S3.

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

Exemple de saisie : copier tous les fichiers d'un dossier local dans un compartiment Amazon S3 avec le préfixe key

L'exemple suivant montre comment copier tous les fichiers du dossier local dans un compartiment Amazon S3 avec le préfixe key. Cet exemple ne copie pas les sous-dossiers ou leur contenu car cela n'`recurse`est pas spécifié, et sa valeur par défaut est. `false`

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

Exemple de saisie : copie récursive de tous les fichiers et dossiers d'un dossier local vers un compartiment Amazon S3

L'exemple suivant montre comment copier tous les fichiers et dossiers de manière récursive d'un dossier local vers un compartiment Amazon S3 avec le préfixe key.

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

Sortie

Aucune.

WebDownload

Le module WebDownloadd'action vous permet de télécharger des fichiers et des ressources depuis un emplacement distant via le protocole HTTP/HTTPS (HTTPS est recommandé). Il n'y a aucune limite quant au nombre ou à la taille des téléchargements. Ce module gère la logique des nouvelles tentatives et des retards exponentiels.

Chaque opération de téléchargement dispose d'un maximum de 5 tentatives pour réussir en fonction des entrées de l'utilisateur. Ces tentatives sont différentes de celles spécifiées dans le `maxAttempts` champ du document`steps`, qui sont liées à des défaillances du module d'action.

Ce module d'action gère implicitement les redirections. Tous les codes d'état HTTP, à l'exception de`200`, génèrent une erreur.

Entrée

Primitif	Description	Type	Obligatoire	Par défaut
`source`	URL HTTP/ HTTPS valide (HTTPS est recommand	Chaîne	Oui	N/A

Primitif	Description	Type	Obligatoire	Par défaut
	é), conforme à la norme RFC 3986. Les expressions de chaînage sont autorisées.			
`destination`	Un chemin de fichier ou de dossier absolu ou relatif sur le système local. Les chemins de dossier doivent se terminer par`/`. S'ils ne se terminent pas par`/`, ils seront traités comme des chemins de fichiers. Le module crée tout fichier ou dossier requis pour des téléchargements réussis. Les expressions de chaînage sont autorisées.	Chaîne	Oui	N/A

Primitif	Description	Type	Obligatoire	Par défaut
`overwrite`	Lorsque cette option est activée, elle remplace tous les fichiers existants sur le système local par le fichier ou la ressource téléchargé. Lorsque cette option n'est pas activée, aucun fichier existant sur le système local n'est remplacé et le module d'action échoue avec une erreur. Lorsque le remplacem ent est activé et que la somme de contrôle et l'algorithme sont spécifiés, le module d'action télécharg e le fichier uniquement si la somme de contrôle et le hachage des fichiers préexista	Booléen	Non	`true`

Primitif	Description	Type	Obligatoire	Par défaut
	nts ne correspon dent pas.			
`checksum`	Lorsque vous spécifiez la somme de contrôle, elle est comparée au hachage du fichier télécharg é généré avec l'algorithme fourni. Pour que la vérificat ion des fichiers soit activée, la somme de contrôle et l'algorithme doivent être fournis. Les expressions de chaînage sont autorisées.	Chaîne	Non	N/A

Primitif	Description	Type	Obligatoire	Par défaut
`algorithm`	Algorithme utilisé pour calculer le checksum. Les options sont MD5, SHA1, SHA256 et SHA512. Pour que la vérification des fichiers soit activée, la somme de contrôle et l'algorithme doivent être fournis. Les expressions de chaînage sont autorisées.	Chaîne	Non	N/A
`ignoreCertificateErrors`	La validation du certificat SSL est ignorée lorsqu'elle est activée.	Booléen	Non	`false`

Sortie

Primitif	Description	Type
`destination`	Chaîne de nouvelle ligne délimitée par des caractères qui indique	Chaîne

Primitif	Description	Type				
	le chemin de destination où sont stockés les fichiers ou les ressources téléchargés.					

Exemple de saisie : téléchargement d'un fichier distant vers une destination locale

```
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"
}
```

Exemple de saisie : téléchargement de plusieurs fichiers distants vers plusieurs destinations locales

```
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"
}
```

Exemple de saisie : téléchargement d'un fichier distant sans remplacer la destination locale et téléchargement d'un autre fichier distant avec vérification du fichier

```
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"
}
```

Exemple de saisie : télécharger un fichier distant et ignorer la validation de la certification 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"
```

```
}
```

Modules de fonctionnement du système de fichiers

La section suivante contient des détails sur les modules d'action qui exécutent les commandes et les instructions de fonctionnement du système de fichiers.

Modules d'action relatifs au fonctionnement du système de fichiers

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

AppendFile

Le module AppendFiled'action ajoute le contenu spécifié au contenu préexistant d'un fichier.

Si la valeur de codage de fichier est différente de la valeur d'encodage (`utf-8`) par défaut, vous pouvez spécifier la valeur de codage de fichier à l'aide de l'`encoding`option. Par défaut, `utf-16` et `utf-32` sont supposés utiliser le codage little-endian.

Le module d'action renvoie une erreur lorsque les événements suivants se produisent :

- Le fichier spécifié n'existe pas au moment de l'exécution.
- Vous n'êtes pas autorisé à écrire pour modifier le contenu du fichier.
- Le module rencontre une erreur lors de l'opération sur le fichier.

Entrée

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
`path`	Le chemin du fichier.	Chaîne	Oui	N/A	N/A	Oui
`content`	Le contenu à ajouter au fichier.	Chaîne	Non	Chaîne vide	N/A	Oui
`encoding`	La norme d'encodage.	Chaîne	Non	`utf8`	`utf8`,`utf-8`, `utf-8-LE`, `utf-16-LE` `utf16-BE`utf-16-BE ,`utf32`,`utf-LE`,`utf-32-LE` ,`utf32-BE`, et `utf-32-BE` . La valeur de l'option de codage ne distingue	Oui

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
					pas les majuscules et minuscules.	

Exemple de saisie : ajout d'un fichier sans encodage (Linux)

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

Exemple de saisie : ajout d'un fichier sans encodage (Windows)

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

Exemple de saisie : ajout d'un fichier avec encodage (Linux)

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

Exemple de saisie : ajout d'un fichier avec encodage (Windows)

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

Exemple de saisie : ajouter un fichier avec une chaîne vide (Linux)

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

Exemple de saisie : ajouter un fichier avec une chaîne vide (Windows)

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

Sortie

Aucune.

CopyFile

Le module CopyFiled'action copie les fichiers de la source spécifiée vers la destination spécifiée. Par défaut, le module crée de manière récursive le dossier de destination s'il n'existe pas au moment de l'exécution.

Si un fichier portant le nom spécifié existe déjà dans le dossier spécifié, le module d'action remplace par défaut le fichier existant. Vous pouvez annuler ce comportement par défaut en définissant l'option de remplacement sur. `false` Lorsque l'option de remplacement est définie sur et qu'il existe déjà un fichier portant le nom spécifié à l'emplacement spécifié, le module d'action renvoie une erreur. `false` Cette option fonctionne de la même manière que la `cp` commande sous Linux, qui remplace par défaut.

Le nom du fichier source peut inclure un caractère générique (*). Les caractères génériques ne sont acceptés qu'après le dernier séparateur de chemin de fichier (/ou\). Si des caractères génériques sont inclus dans le nom du fichier source, tous les fichiers correspondant au caractère générique

sont copiés dans le dossier de destination. Si vous souhaitez déplacer plusieurs fichiers à l'aide d'un caractère générique, la saisie de l'`destination`option doit se terminer par un séparateur de chemin de fichier (`/`ou`\`), ce qui indique que l'entrée de destination est un dossier.

Si le nom du fichier de destination est différent du nom du fichier source, vous pouvez spécifier le nom du fichier de destination à l'aide de l'`destination`option. Si vous ne spécifiez pas de nom de fichier de destination, le nom du fichier source est utilisé pour créer le fichier de destination. Tout texte qui suit le dernier séparateur de chemin de fichier (`/`ou`\`) est traité comme le nom du fichier. Si vous souhaitez utiliser le même nom de fichier que le fichier source, l'entrée de l'`destination`option doit se terminer par un séparateur de chemin de fichier (`/`ou`\`).

Le module d'action renvoie une erreur lorsque les événements suivants se produisent :

- Vous n'êtes pas autorisé à créer un fichier dans le dossier indiqué.
- Les fichiers source n'existent pas au moment de l'exécution.
- Il existe déjà un dossier portant le nom de fichier spécifié et l'`overwrite`option est définie sur`false`.
- Le module d'action rencontre une erreur lors de l'exécution de l'opération.

Entrée

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
`source`	Le chemin du fichier source.	Chaîne	Oui	N/A	N/A	Oui
`destination`	Le chemin du fichier de destination.	Chaîne	Oui	N/A	N/A	Oui
`overwrite`	Lorsqu'il est défini	Booléen	Non	`true`	N/A	Oui

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
	sur false, les fichiers de destination ne seront pas remplacés s'il existe déjà un fichier portant le nom spécifié à l'emplacement spécifié.					

Exemple de saisie : copier un fichier (Linux)

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

Exemple de saisie : copier un fichier (Windows)

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

Exemple de saisie : copier un fichier en utilisant le nom du fichier source (Linux)

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

Exemple de saisie : copier un fichier en utilisant le nom du fichier source (Windows)

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

Exemple de saisie : copier un fichier à l'aide du caractère générique (Linux)

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

Exemple de saisie : copie d'un fichier à l'aide du caractère générique (Windows)

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

Exemple de saisie : copier un fichier sans le remplacer (Linux)

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

Exemple de saisie : copier un fichier sans le remplacer (Windows)

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

Sortie

Aucune.

CopyFolder

Le module CopyFolderd'action copie un dossier de la source spécifiée vers la destination spécifiée. L'entrée de l'`source`option est le dossier à copier, et l'entrée de l'`destination`option est le dossier dans lequel le contenu du dossier source est copié. Par défaut, le module crée de manière récursive le dossier de destination s'il n'existe pas au moment de l'exécution.

Si un dossier portant le nom spécifié existe déjà dans le dossier spécifié, le module d'action remplace par défaut le dossier existant. Vous pouvez annuler ce comportement par défaut en définissant l'option de remplacement sur. `false` Lorsque l'option de remplacement est définie sur et qu'il existe déjà un dossier portant le nom spécifié à l'emplacement spécifié, le module d'action renvoie une erreur. `false`

Le nom du dossier source peut inclure un caractère générique (`*`). Les caractères génériques ne sont acceptés qu'après le dernier séparateur de chemin de fichier (`/`ou`\`). Si des caractères génériques sont inclus dans le nom du dossier source, tous les dossiers correspondant au caractère générique sont copiés dans le dossier de destination. Si vous souhaitez copier plusieurs dossiers à l'aide d'un caractère générique, la saisie de l'`destination`option doit se terminer par un séparateur de chemin de fichier (`/`ou`\`), ce qui indique que l'entrée de destination est un dossier.

Si le nom du dossier de destination est différent du nom du dossier source, vous pouvez spécifier le nom du dossier de destination à l'aide de l'`destination`option. Si vous ne spécifiez pas de nom de dossier de destination, le nom du dossier source est utilisé pour créer le dossier de destination. Tout texte qui suit le dernier séparateur de chemin de fichier (`/`ou`\`) est traité comme le nom du dossier. Si vous souhaitez utiliser le même nom de dossier que le dossier source, l'entrée de l'`destination`option doit se terminer par un séparateur de chemin de fichier (`/`ou`\`).

Le module d'action renvoie une erreur lorsque les événements suivants se produisent :

- Vous n'êtes pas autorisé à créer un dossier dans le dossier spécifié.
- Les dossiers source n'existent pas au moment de l'exécution.
- Il existe déjà un dossier portant le nom de dossier spécifié et l'`overwrite`option est définie sur`false`.
- Le module d'action rencontre une erreur lors de l'exécution de l'opération.

Entrée

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
`source`	Le chemin du dossier source.	Chaîne	Oui	N/A	N/A	Oui
`destination`	Le chemin du dossier de destination.	Chaîne	Oui	N/A	N/A	Oui
`overwrite`	Lorsqu'il est défini sur false, les dossiers de destination ne seront pas remplacés s'il existe déjà un dossier portant le nom	Booléen	Non	`true`	N/A	Oui

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
	spécifié à l'emplacement spécifié.					

Exemple de saisie : copier un dossier (Linux)

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

Exemple de saisie : copier un dossier (Windows)

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

Exemple de saisie : copier un dossier en utilisant le nom du dossier source (Linux)

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

Exemple de saisie : copier un dossier en utilisant le nom du dossier source (Windows)

```
name: CopyingFolderSourceFolderNameWindows
action: CopyFolder
```

```
inputs:
  - source: C:\Sample\MyFolder\SampleFolder
    destination: C:\MyFolder\
```

Exemple de saisie : copier un dossier à l'aide du caractère générique (Linux)

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

Exemple de saisie : copier un dossier à l'aide du caractère générique (Windows)

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

Exemple de saisie : copier un dossier sans le remplacer (Linux)

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

Exemple de saisie : copier un dossier sans le remplacer (Windows)

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

Sortie

Aucune.

CreateFile

Le module CreateFiled'action crée un fichier dans un emplacement spécifié. Par défaut, si nécessaire, le module crée également les dossiers parents de manière récursive.

Si le fichier existe déjà dans le dossier spécifié, le module d'action tronque ou remplace par défaut le fichier existant. Vous pouvez annuler ce comportement par défaut en définissant l'option de remplacement sur. `false` Lorsque l'option de remplacement est définie sur et qu'il existe déjà un fichier portant le nom spécifié à l'emplacement spécifié, le module d'action renvoie une erreur. `false`

Si la valeur de codage de fichier est différente de la valeur d'encodage (`utf-8`) par défaut, vous pouvez spécifier la valeur de codage de fichier à l'aide de l'`encoding`option. Par défaut, `utf-16` et `utf-32` sont supposés utiliser le codage little-endian.

`owner`, `group`, et `permissions` sont des entrées facultatives. L'entrée pour `permissions` doit être une valeur de chaîne. Les fichiers sont créés avec des valeurs par défaut lorsqu'elles ne sont pas fournies. Ces options ne sont pas prises en charge sur les plateformes Windows. Ce module d'action valide et renvoie une erreur si les `permissions` options `owner`, `group`, et sont utilisées sur les plateformes Windows.

Ce module d'action peut créer un fichier dont les autorisations sont définies par la `umask` valeur par défaut du système d'exploitation. Vous devez définir la `umask` valeur si vous souhaitez remplacer la valeur par défaut.

Le module d'action renvoie une erreur lorsque les événements suivants se produisent :

- Vous n'êtes pas autorisé à créer un fichier ou un dossier dans le dossier parent spécifié.
- Le module d'action rencontre une erreur lors de l'exécution de l'opération.

Entrée

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
`path`	Le chemin du fichier.	Chaîne	Oui	N/A	N/A	Oui

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
`content`	Le contenu textuel du fichier.	Chaîne	Non	N/A	N/A	Oui
`encoding`	La norme d'encodage.	Chaîne	Non	`utf8`	`utf8`,`utf-8`, `utf-16-LE`, `utf16-BE`, `utf-16-BE`, `utf32`, `utf-32-LE`, `utf32-BE`, et `utf-32-BE`. La valeur de l'option de codage ne distingue pas les majuscules et minuscules.	Oui

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
`owner`	Le nom d'utilisateur ou l'ID.	Chaîne	Non	N/A	N/A	Non pris en charge sous Windows.
`group`	Le nom ou l'identifiant du groupe.	Chaîne	Non	L'utilisateur actuel.	N/A	Non pris en charge sous Windows.
`permissions`	Les autorisations du fichier.	Chaîne	Non	`0666`	N/A	Non pris en charge sous Windows.
`overwrite`	Si le nom du fichier spécifié existe déjà, définissez cette valeur pour `false` empêcher le fichier d'être tronqué ou remplacé par défaut.	Booléen	Non	`true`	N/A	Oui

Exemple de saisie : créer un fichier sans le remplacer (Linux)

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

Exemple de saisie : créer un fichier sans le remplacer (Windows)

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

Exemple de saisie : création d'un fichier avec des propriétés de fichier

```
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
```

Exemple de saisie : création d'un fichier sans propriétés de fichier

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

Exemple de saisie : créer un fichier vide pour ignorer une section du script de nettoyage Linux

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

Pour plus d'informations, consultez Remplacer le script de nettoyage Linux.

Sortie

Aucune.

CreateFolder

Le module CreateFolderd'action crée un dossier à un emplacement spécifié. Par défaut, si nécessaire, le module crée également les dossiers parents de manière récursive.

Si le dossier existe déjà dans le dossier spécifié, le module d'action tronque ou remplace par défaut le dossier existant. Vous pouvez annuler ce comportement par défaut en définissant l'option de remplacement sur. `false` Lorsque l'option de remplacement est définie sur et qu'il existe déjà un dossier portant le nom spécifié à l'emplacement spécifié, le module d'action renvoie une erreur. `false`

`ownergroup`, et `permissions` sont des entrées facultatives. L'entrée pour `permissions` doit être une valeur de chaîne. Ces options ne sont pas prises en charge sur les plateformes Windows. Ce module d'action valide et renvoie une erreur si les `permissions` options `ownergroup`, et sont utilisées sur les plateformes Windows.

Ce module d'action peut créer un dossier dont les autorisations sont définies par la `umask` valeur par défaut du système d'exploitation. Vous devez définir la `umask` valeur si vous souhaitez remplacer la valeur par défaut.

Le module d'action renvoie une erreur lorsque les événements suivants se produisent :

- Vous n'êtes pas autorisé à créer un dossier à l'emplacement indiqué.
- Le module d'action rencontre une erreur lors de l'exécution de l'opération.

Entrée

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
`path`	Le chemin du dossier.	Chaîne	Oui	N/A	N/A	Oui
`owner`	Le nom d'utilisateur ou l'ID.	Chaîne	Non	L'utilisateur actuel.	N/A	Non pris en charge sous Windows.
`group`	Le nom ou l'identifiant du groupe.	Chaîne	Non	Le groupe de l'utilisateur actuel.	N/A	Non pris en charge sous Windows.
`permissions`	Les autorisations relatives aux dossiers.	Chaîne	Non	`0777`	N/A	Non pris en charge sous Windows.
`overwrite`	Si le nom du fichier spécifié existe déjà, définissez cette valeur pour `false` empêcher le fichier	Booléen	Non	`true`	N/A	Oui

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
	d'être tronqué ou remplacé par défaut.					

Exemple de saisie : créer un dossier (Linux)

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

Exemple de saisie : créer un dossier (Windows)

```
name: CreatingFolderWindows
action: CreateFolder
inputs:
  - path: C:\MyFolder
```

Exemple de saisie : création d'un dossier en spécifiant les propriétés du dossier

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

Exemple de saisie : créer un dossier qui remplace le dossier existant, s'il y en a un.

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

Sortie

Aucune.

CreateSymlink

Le module CreateSymlinkd'action crée des liens symboliques ou des fichiers contenant une référence à un autre fichier. Ce module n'est pas pris en charge sur les plateformes Windows.

L'entrée pour les `target` options `path` et peut être un chemin absolu ou relatif. Si l'entrée de l'`path`option est un chemin relatif, il est remplacé par le chemin absolu lors de la création du lien.

Par défaut, lorsqu'un lien portant le nom spécifié existe déjà dans le dossier spécifié, le module d'action renvoie une erreur. Vous pouvez annuler ce comportement par défaut en définissant l'`force`option sur. `true` Lorsque l'`force`option est définie sur`true`, le module remplace le lien existant.

Si aucun dossier parent n'existe, le module d'action crée le dossier de manière récursive, par défaut.

Le module d'action renvoie une erreur lorsque les événements suivants se produisent :

- Le fichier cible n'existe pas au moment de l'exécution.
- Un fichier de lien non symbolique portant le nom spécifié existe déjà.
- Le module d'action rencontre une erreur lors de l'exécution de l'opération.

Entrée

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
`path`	Le chemin du fichier.	Chaîne	Oui	N/A	N/A	Non pris en charge sous Windows.
`target`	Le chemin du fichier cible vers lequel pointe le lien symbolique.	Chaîne	Oui	N/A	N/A	Non pris en charge sous Windows.
`force`	Force la création d'un lien lorsqu'un lien portant le même nom existe déjà.	Booléen	Non	`false`	N/A	Non pris en charge sous Windows.

Exemple de saisie : créer un lien symbolique qui force la création d'un lien

```
name: CreatingSymbolicLinkWithForce
action: CreateSymlink
inputs:
  - path: /Folder2/Symboliclink.txt
    target: /Folder/Sample.txt
```

```
    force: true
```

Exemple de saisie : créer un lien symbolique qui ne force pas la création d'un lien

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

Sortie

Aucune.

DeleteFile

Le module DeleteFiled'action supprime un ou plusieurs fichiers dans un emplacement spécifié.

L'entrée de `path` doit être un chemin de fichier valide ou un chemin de fichier avec un caractère générique (*) dans le nom du fichier. Lorsque des caractères génériques sont spécifiés dans le nom du fichier, tous les fichiers du même dossier qui correspondent au caractère générique sont supprimés.

Le module d'action renvoie une erreur lorsque les événements suivants se produisent :

- Vous n'êtes pas autorisé à effectuer des opérations de suppression.
- Le module d'action rencontre une erreur lors de l'exécution de l'opération.

Entrée

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
`path`	Le chemin du fichier.	Chaîne	Oui	N/A	N/A	Oui

Exemple de saisie : suppression d'un seul fichier (Linux)

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

Exemple de saisie : supprimer un seul fichier (Windows)

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

Exemple de saisie : supprimer un fichier qui se termine par « log » (Linux)

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

Exemple de saisie : supprimer un fichier qui se termine par « log » (Windows)

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

Exemple de saisie : supprimer tous les fichiers d'un dossier spécifié (Linux)

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

Exemple de saisie : supprimer tous les fichiers d'un dossier spécifié (Windows)

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

Sortie

Aucune.

DeleteFolder

Le module DeleteFolderd'action supprime les dossiers.

Si le dossier n'est pas vide, vous devez définir l'`force`option `true` de suppression du dossier et de son contenu. Si vous ne définissez pas l'`force`option sur `true` et que le dossier que vous essayez de supprimer n'est pas vide, le module d'action renvoie une erreur. La valeur par défaut de l'`force`option est`false`.

Le module d'action renvoie une erreur lorsque les événements suivants se produisent :

- Vous n'êtes pas autorisé à effectuer des opérations de suppression.
- Le module d'action rencontre une erreur lors de l'exécution de l'opération.

Entrée

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
`path`	Le chemin du dossier.	Chaîne	Oui	N/A	N/A	Oui
`force`	Supprime le dossier, qu'il soit vide ou non.	Booléen	Non	`false`	N/A	Oui

Exemple de saisie : supprimer un dossier qui n'est pas vide à l'aide de l'**`force`**option (Linux)

```
name: DeletingFolderWithForceOptionLinux
action: DeleteFolder
```

```
inputs:
  - path: /Sample/MyFolder/Sample/
    force: true
```

Exemple de saisie : supprimer un dossier qui n'est pas vide à l'aide de l'**`force`**option (Windows)

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

Exemple de saisie : suppression d'un dossier (Linux)

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

Exemple de saisie : suppression d'un dossier (Windows)

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

Sortie

Aucune.

ListFiles

Le module ListFilesd'action répertorie les fichiers d'un dossier spécifié. Lorsque l'option récursive est définie sur`true`, elle répertorie les fichiers dans des sous-dossiers. Ce module ne répertorie pas les fichiers dans les sous-dossiers par défaut.

Pour répertorier tous les fichiers dont les noms correspondent à un modèle spécifié, utilisez l'`fileNamePattern`option permettant de fournir le modèle. L'`fileNamePattern`option accepte la valeur wildcard (`*`). Lorsque le `fileNamePattern` est fourni, tous les fichiers correspondant au format de nom de fichier spécifié sont renvoyés.

Le module d'action renvoie une erreur lorsque les événements suivants se produisent :

- Le dossier spécifié n'existe pas au moment de l'exécution.
- Vous n'êtes pas autorisé à créer un fichier ou un dossier dans le dossier parent spécifié.
- Le module d'action rencontre une erreur lors de l'exécution de l'opération.

Entrée

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
`path`	Le chemin du dossier.	Chaîne	Oui	N/A	N/A	Oui
`fileNamePattern`	Le modèle à associer pour répertorier tous les fichiers dont les noms correspondent au modèle.	Chaîne	Non	N/A	N/A	Oui
`recursive`	Répertorie les fichiers du dossier de manière récursive.	Booléen	Non	`false`	N/A	Oui

Exemple de saisie : liste des fichiers dans le dossier spécifié (Linux)

```
name: ListingFilesInSampleFolderLinux
action: ListFiles
```

```
inputs:
  - path: /Sample/MyFolder/Sample
```

Exemple de saisie : liste des fichiers dans le dossier spécifié (Windows)

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

Exemple de saisie : liste les fichiers qui se terminent par « log » (Linux)

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

Exemple de saisie : liste les fichiers qui se terminent par « log » (Windows)

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

Exemple de saisie : liste des fichiers de manière récursive

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

Sortie

Primitif	Description	Type
`files`	La liste des fichiers.	Chaîne

Exemple de sortie

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

MoveFile

Le module MoveFiled'action déplace les fichiers de la source spécifiée vers la destination spécifiée.

Si le fichier existe déjà dans le dossier spécifié, le module d'action remplace par défaut le fichier existant. Vous pouvez annuler ce comportement par défaut en définissant l'option de remplacement sur. `false` Lorsque l'option de remplacement est définie sur et qu'il existe déjà un fichier portant le nom spécifié à l'emplacement spécifié, le module d'action renvoie une erreur. `false` Cette option fonctionne de la même manière que la `mv` commande sous Linux, qui remplace par défaut.

Le nom du fichier source peut inclure un caractère générique (*). Les caractères génériques ne sont acceptés qu'après le dernier séparateur de chemin de fichier (/ou\). Si des caractères génériques sont inclus dans le nom du fichier source, tous les fichiers correspondant au caractère générique sont copiés dans le dossier de destination. Si vous souhaitez déplacer plusieurs fichiers à l'aide d'un caractère générique, la saisie de l'`destination`option doit se terminer par un séparateur de chemin de fichier (/ou\), ce qui indique que l'entrée de destination est un dossier.

Si le nom du fichier de destination est différent du nom du fichier source, vous pouvez spécifier le nom du fichier de destination à l'aide de l'`destination`option. Si vous ne spécifiez pas de nom de fichier de destination, le nom du fichier source est utilisé pour créer le fichier de destination. Tout texte qui suit le dernier séparateur de chemin de fichier (/ou\) est traité comme le nom du fichier. Si vous souhaitez utiliser le même nom de fichier que le fichier source, l'entrée de l'`destination`option doit se terminer par un séparateur de chemin de fichier (/ou\).

Le module d'action renvoie une erreur lorsque les événements suivants se produisent :

- Vous n'êtes pas autorisé à créer un fichier dans le dossier indiqué.
- Les fichiers source n'existent pas au moment de l'exécution.
- Il existe déjà un dossier portant le nom de fichier spécifié et l'`overwrite`option est définie sur`false`.
- Le module d'action rencontre une erreur lors de l'exécution de l'opération.

Entrée

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
`source`	Le chemin du fichier source.	Chaîne	Oui	N/A	N/A	Oui
`destination`	Le chemin du fichier de destination.	Chaîne	Oui	N/A	N/A	Oui
`overwrite`	Lorsqu'il est défini sur false, les fichiers de destination ne seront pas remplacés s'il existe déjà un fichier portant le nom spécifié à l'emplacement spécifié.	Booléen	Non	`true`	N/A	Oui

Exemple de saisie : déplacer un fichier (Linux)

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

Exemple de saisie : déplacer un fichier (Windows)

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

Exemple de saisie : déplacer un fichier en utilisant le nom du fichier source (Linux)

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

Exemple de saisie : déplacer un fichier en utilisant le nom du fichier source (Windows)

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

Exemple de saisie : déplacer un fichier à l'aide d'un caractère générique (Linux)

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

Exemple de saisie : déplacer un fichier à l'aide d'un caractère générique (Windows)

```
name: MovingFilesWithWildCardWindows
```

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

Exemple de saisie : déplacer un fichier sans le remplacer (Linux)

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

Exemple de saisie : déplacer un fichier sans le remplacer (Windows)

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

Sortie

Aucune.

MoveFolder

Le module MoveFolderd'action déplace les dossiers de la source spécifiée vers la destination spécifiée. L'entrée de l'`source`option est le dossier à déplacer, et l'entrée de l'`destination`option est le dossier dans lequel le contenu des dossiers source est déplacé.

Si le dossier parent de destination ou l'entrée de l'`destination`option n'existe pas au moment de l'exécution, le comportement par défaut du module consiste à créer le dossier de manière récursive à la destination spécifiée.

Si un dossier identique au dossier source existe déjà dans le dossier de destination, le module d'action remplace par défaut le dossier existant. Vous pouvez annuler ce comportement par défaut en définissant l'option de remplacement sur. `false` Lorsque l'option de remplacement est définie

sur et qu'il existe déjà un dossier portant le nom spécifié à l'emplacement spécifié, le module d'action renvoie une erreur. `false`

Le nom du dossier source peut inclure un caractère générique (`*`). Les caractères génériques ne sont acceptés qu'après le dernier séparateur de chemin de fichier (`/`ou`\`). Si des caractères génériques sont inclus dans le nom du dossier source, tous les dossiers correspondant au caractère générique sont copiés dans le dossier de destination. Si vous souhaitez déplacer plusieurs dossiers à l'aide d'un caractère générique, la saisie de l'`destination`option doit se terminer par un séparateur de chemin de fichier (`/`ou`\`), ce qui indique que l'entrée de destination est un dossier.

Si le nom du dossier de destination est différent du nom du dossier source, vous pouvez spécifier le nom du dossier de destination à l'aide de l'`destination`option. Si vous ne spécifiez pas de nom de dossier de destination, le nom du dossier source est utilisé pour créer le dossier de destination. Tout texte qui suit le dernier séparateur de chemin de fichier (`/`ou`\`) est traité comme le nom du dossier. Si vous souhaitez utiliser le même nom de dossier que le dossier source, l'entrée de l'`destination`option doit se terminer par un séparateur de chemin de fichier (`/`ou`\`).

Le module d'action renvoie une erreur lorsque les événements suivants se produisent :

- Vous n'êtes pas autorisé à créer un dossier dans le dossier de destination.
- Les dossiers source n'existent pas au moment de l'exécution.
- Il existe déjà un dossier portant le nom spécifié et l'`overwrite`option est définie sur`false`.
- Le module d'action rencontre une erreur lors de l'exécution de l'opération.

Entrée

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
`source`	Le chemin du dossier source.	Chaîne	Oui	N/A	N/A	Oui
`destination`	Le chemin du dossier	Chaîne	Oui	N/A	N/A	Oui

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
	de destination.					
`overwrite`	Lorsqu'il est défini sur false, les dossiers de destination ne seront pas remplacés s'il existe déjà un dossier portant le nom spécifié à l'emplacement spécifié.	Booléen	Non	`true`	N/A	Oui

Exemple de saisie : déplacer un dossier (Linux)

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

Exemple de saisie : déplacer un dossier (Windows)

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

Exemple de saisie : déplacer un dossier en utilisant le nom du dossier source (Linux)

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

Exemple de saisie : déplacer un dossier en utilisant le nom du dossier source (Windows)

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

Exemple de saisie : déplacer un dossier à l'aide d'un caractère générique (Linux)

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

Exemple de saisie : déplacer un dossier à l'aide d'un caractère générique (Windows)

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

Exemple de saisie : déplacer un dossier sans le remplacer (Linux)

```
name: MovingFoldersWithoutOverwriteLinux
```

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

Exemple de saisie : déplacer un dossier sans le remplacer (Windows)

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

Sortie

Aucune.

ReadFile

Le module ReadFiled'action lit le contenu d'un fichier texte de type chaîne. Ce module peut être utilisé pour lire le contenu d'un fichier afin de l'utiliser dans les étapes suivantes via le chaînage ou pour lire des données dans le `console.log` fichier. Si le chemin spécifié est un lien symbolique, ce module renvoie le contenu du fichier cible. Ce module ne prend en charge que les fichiers texte.

Si la valeur de codage de fichier est différente de la valeur d'encodage (`utf-8`) par défaut, vous pouvez spécifier la valeur de codage de fichier à l'aide de l'`encoding`option. Par défaut, `utf-16` et `utf-32` sont supposés utiliser le codage little-endian.

Par défaut, ce module ne peut pas imprimer le contenu du fichier dans le `console.log` fichier. Vous pouvez annuler ce paramètre en attribuant à `true` la `printFileContent` propriété la valeur.

Ce module ne peut renvoyer que le contenu d'un fichier. Il ne peut pas analyser les fichiers, tels que les fichiers Excel ou JSON.

Le module d'action renvoie une erreur lorsque les événements suivants se produisent :

- Le fichier n'existe pas au moment de l'exécution.
- Le module d'action rencontre une erreur lors de l'exécution de l'opération.

Entrée

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
`path`	Le chemin du fichier.	Chaîne	Oui	N/A	N/A	Oui
`encoding`	La norme d'encodage.	Chaîne	Non	`utf8`	`utf8`,`utf-8-LE`, `utf-16-LE` `utf16-BE`utf-16-BE ,`utf32`,`utf-LE`,`utf-32-LE` ,`utf32-BE`, et `utf-32-BE` . La valeur de l'option de codage ne distingue pas les majuscules et minuscules.	Oui	6,utf1
`printFileContent`	Imprime le contenu du fichier	Booléen	Non	false	N/A	Oui.

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
	dans le `console.log` fichier.					

Exemple de saisie : lecture d'un fichier (Linux)

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

Exemple de saisie : lecture d'un fichier (Windows)

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

Exemple de saisie : lecture d'un fichier et spécification de la norme de codage

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

Exemple de saisie : lire un fichier et imprimer dans le **`console.log`** fichier

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

Sortie

Champ	Description	Type
`content`	Le contenu du fichier.	chaîne

Exemple de sortie

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

SetFileEncoding

Le module SetFileEncodingd'action modifie la propriété de codage d'un fichier existant. Ce module peut convertir le codage de fichiers `utf-8` à partir d'une norme de codage spécifiée. Par défaut, `utf-16` et `utf-32` sont supposés être des encodages little-endian.

Le module d'action renvoie une erreur lorsque les événements suivants se produisent :

- Vous n'êtes pas autorisé à effectuer la modification spécifiée.
- Le fichier n'existe pas au moment de l'exécution.
- Le module d'action rencontre une erreur lors de l'exécution de l'opération.

Entrée

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
`path`	Le chemin du fichier.	Chaîne	Oui	N/A	N/A	Oui
`encoding`	La norme d'encodage.	Chaîne	Non	`utf8`	`utf8`,`utf-8`, `LE`,	Oui

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
					`utf-16-LE` `utf16-BEutf-16-BE` ,`utf32`,`utfLE`,`utf-32-LE` ,`utf32-BE`, et `utf-32-BE` . La valeur de l'option de codage ne distingue pas les majuscules et minuscules.	

Exemple de saisie : définir la propriété de codage du fichier

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

Sortie

Aucune.

SetFileOwner

Le module SetFileOwnerd'action modifie les propriétés `owner` et le `group` propriétaire d'un fichier existant. Si le fichier spécifié est un lien symbolique, le module modifie la `owner` propriété du fichier source. Ce module n'est pas pris en charge sur les plateformes Windows.

Ce module accepte les noms d'utilisateur et de groupe en entrée. Si le nom du groupe n'est pas fourni, le module affecte le propriétaire du fichier au groupe auquel appartient l'utilisateur.

Le module d'action renvoie une erreur lorsque les événements suivants se produisent :

- Vous n'êtes pas autorisé à effectuer la modification spécifiée.
- Le nom d'utilisateur ou de groupe spécifié n'existe pas au moment de l'exécution.
- Le fichier n'existe pas au moment de l'exécution.
- Le module d'action rencontre une erreur lors de l'exécution de l'opération.

Entrée

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
`path`	Le chemin du fichier.	Chaîne	Oui	N/A	N/A	Non pris en charge sous Windows.
`owner`	Nom de l'utilisateur.	chaîne	Oui	N/A	N/A	Non pris en charge sous Windows.
`group`	Nom du groupe	Chaîne	Non	Nom du groupe	N/A	Non pris en charge

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
	d'utilisateurs.			auquel appartient l'utilisateur.		sous Windows.

Exemple de saisie : définir la propriété du propriétaire du fichier sans spécifier le nom du groupe d'utilisateurs

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

Exemple de saisie : définir la propriété du propriétaire du fichier en spécifiant le propriétaire et le groupe d'utilisateurs

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

Sortie

Aucune.

SetFolderOwner

Le module SetFolderOwnerd'action modifie de manière récursive les propriétés `owner` et le `group` propriétaire d'un dossier existant. Par défaut, le module peut modifier la propriété de l'ensemble du contenu d'un dossier. Vous pouvez définir l'`recursive`option `false` pour annuler ce comportement. Ce module n'est pas pris en charge sur les plateformes Windows.

Ce module accepte les noms d'utilisateur et de groupe en entrée. Si le nom du groupe n'est pas fourni, le module affecte le propriétaire du fichier au groupe auquel appartient l'utilisateur.

Le module d'action renvoie une erreur lorsque les événements suivants se produisent :

- Vous n'êtes pas autorisé à effectuer la modification spécifiée.
- Le nom d'utilisateur ou de groupe spécifié n'existe pas au moment de l'exécution.
- Le dossier n'existe pas au moment de l'exécution.
- Le module d'action rencontre une erreur lors de l'exécution de l'opération.

Entrée

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
`path`	Le chemin du dossier.	Chaîne	Oui	N/A	N/A	Non pris en charge sous Windows.
`owner`	Nom de l'utilisateur.	chaîne	Oui	N/A	N/A	Non pris en charge sous Windows.
`group`	Nom du groupe d'utilisateurs.	Chaîne	Non	Nom du groupe auquel appartient l'utilisateur.	N/A	Non pris en charge sous Windows.
`recursive`	Remplace le comportement par	Booléen	Non	`true`	N/A	Non pris en charge sous Windows.

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
	défaut qui consiste à modifier la propriété de l'ensemble du contenu d'un dossier lorsqu'il est défini sur. `false`					

Exemple de saisie : définir la propriété du propriétaire du dossier sans spécifier le nom du groupe d'utilisateurs

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

Exemple de saisie : définir la propriété du propriétaire du dossier sans annuler la propriété de l'ensemble du contenu d'un dossier

```
name: SettingFolderPropertyWithOutRecursively
action: SetFolderOwner
inputs:
  - path: /SampleFolder/
    owner: LinuxUser
    recursive: false
```

Exemple de saisie : définir la propriété de propriété du fichier en spécifiant le nom du groupe d'utilisateurs

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

Sortie

Aucune.

SetFilePermissions

Le module SetFilePermissionsd'action modifie `permissions` un fichier existant. Ce module n'est pas pris en charge sur les plateformes Windows.

L'entrée pour `permissions` doit être une valeur de chaîne.

Ce module d'action peut créer un fichier dont les autorisations sont définies par la valeur umask par défaut du système d'exploitation. Vous devez définir la `umask` valeur si vous souhaitez remplacer la valeur par défaut.

Le module d'action renvoie une erreur lorsque les événements suivants se produisent :

- Vous n'êtes pas autorisé à effectuer la modification spécifiée.
- Le fichier n'existe pas au moment de l'exécution.
- Le module d'action rencontre une erreur lors de l'exécution de l'opération.

Entrée

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
`path`	Le chemin du fichier.	Chaîne	Oui	N/A	N/A	Non pris en charge sous Windows.
`permissions`	Les autorisations du fichier.	Chaîne	Oui	N/A	N/A	Non pris en charge sous Windows.

Exemple de saisie : modification des autorisations de fichier

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

Sortie

Aucune.

SetFolderPermissions

Le module SetFolderPermissionsd'action modifie `permissions` de manière récursive le dossier existant et tous ses sous-fichiers et sous-dossiers. Par défaut, ce module peut modifier les autorisations pour tout le contenu du dossier spécifié. Vous pouvez définir l'`recursive`option `false` pour annuler ce comportement. Ce module n'est pas pris en charge sur les plateformes Windows.

L'entrée pour `permissions` doit être une valeur de chaîne.

Ce module d'action peut modifier les autorisations en fonction de la valeur umask par défaut du système d'exploitation. Vous devez définir la `umask` valeur si vous souhaitez remplacer la valeur par défaut.

Le module d'action renvoie une erreur lorsque les événements suivants se produisent :

- Vous n'êtes pas autorisé à effectuer la modification spécifiée.
- Le dossier n'existe pas au moment de l'exécution.
- Le module d'action rencontre une erreur lors de l'exécution de l'opération.

Entrée

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
`path`	Le chemin du dossier.	Chaîne	Oui	N/A	N/A	Non pris en charge sous Windows.
`permissions`	Les autorisations relatives aux dossiers.	Chaîne	Oui	N/A	N/A	Non pris en charge sous Windows.
`recursive`	Remplace le comportement par défaut qui consiste à modifier les autorisat	Booléen	Non	`true`	N/A	Non pris en charge sous Windows.

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables	Pris en charge sur toutes les plateformes
	ions pour l'ensemble du contenu d'un dossier lorsqu'il est défini sur. `false`					

Exemple de saisie : définir les autorisations des dossiers

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

Exemple de saisie : définir les autorisations d'un dossier sans modifier les autorisations pour l'ensemble du contenu d'un dossier

```
name: SettingFolderPermissionsNoRecursive
action: SetFolderPermissions
inputs:
  - path: /home/UserName/SampleFolder/
    permissions: 777
    recursive: false
```

Sortie

Aucune.

Actions d'installation du logiciel

Cette section décrit les modules d'action qui exécutent les commandes d'action et les instructions d'installation du logiciel.

Exigences relatives à l'IAM

Si le chemin de téléchargement de votre installation est un URI S3, le rôle IAM que vous associez à votre profil d'instance doit être autorisé à exécuter le module `S3Download` d'action. Pour accorder l'autorisation requise, attachez la politique `S3:GetObject` IAM au rôle IAM associé à votre profil d'instance et spécifiez le chemin de votre bucket. Par exemple, *`arn:aws:s3:::BucketName/*`*).

Entrées MSI complexes

Si vos chaînes d'entrée contiennent des guillemets (''), vous devez utiliser l'une des méthodes suivantes pour vous assurer qu'elles sont interprétées correctement :

- Vous pouvez utiliser des guillemets simples (') à l'extérieur de votre chaîne pour la contenir, et des guillemets doubles («) à l'intérieur de votre chaîne, comme indiqué dans l'exemple suivant.

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

  Dans ce cas, si vous devez utiliser une apostrophe à l'intérieur de votre chaîne, vous devez y échapper. Cela signifie qu'il faut utiliser un autre guillemet simple (') avant l'apostrophe.
- Vous pouvez utiliser des guillemets («) à l'extérieur de votre chaîne pour la contenir. Et vous pouvez éviter les guillemets doubles à l'intérieur de votre chaîne en utilisant la barre oblique inverse (\), comme indiqué dans l'exemple suivant.

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

Ces deux méthodes transmettent la valeur `COMPANYNAME="Acme ""Widgets"" and ""Gizmos."""` à la msiexec commande.

Modules d'action pour l'installation du logiciel

- Installez MSI
- Désinstaller MSI

Installez MSI

Le module `InstallMSI` d'action installe une application Windows à l'aide d'un fichier MSI. Vous pouvez spécifier le fichier MSI à l'aide d'un chemin local, d'un URI d'objet S3 ou d'une URL Web. L'option de redémarrage configure le comportement de redémarrage du système.

AWSTOE génère la msiexec commande en fonction des paramètres d'entrée du module d'action. Les valeurs des paramètres d'entrée `path` (emplacement du fichier MSI) et `logFile` (emplacement du fichier journal) doivent être placées entre guillemets («).

Les codes de sortie MSI suivants sont considérés comme réussis :

- 0 (Succès)
- 1614 (ERROR_PRODUCT_UNINSTALL)
- 1641 (redémarrage initié)
- 3010 (redémarrage requis)

Entrée

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
`path`	Spécifiez l'emplacement du fichier MSI à l'aide de l'une des méthodes suivantes : • Le chemin du fichier local. Le chemin peut être	Chaîne	Oui	N/A	N/A

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
	absolu ou relatif • Un URI d'objet S3 valide. • Une URL HTTP/HTTPS Web valide (HTTPS est recommandé) conforme à la norme RFC 3986. Les expressions de chaînage sont autorisées.				

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
`reboot`	Configurez le comportement de redémarrage du système après une exécution réussie du module d'action. Paramètres : • `Force`— Lance un redémarrage du système une fois la msiexec commande exécutée avec succès. • `Allow`— Lance un redémarrage du système si la msiexec commande renvoie	Chaîne	Non	`Allow`	`Allow`, `Force`, `Skip`

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
	un code de sortie indiquant qu'un redémarrage est nécessaire. • `Skip`— Enregistre un message d'information dans le `console.log` fichier indiquant qu'un redémarrage a été ignoré. Cette option empêche le redémarrage, même si la msiexec commande renvoie un code de sortie indiquant qu'un				

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
	redémarrage est nécessaire.				

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
`logOptions`	Spécifiez les options à utiliser pour la journalisation de l'installation MSI. Les indicateurs spécifiés sont transmis au programme d'installation MSI, avec le paramètre de ligne de `/L` commande pour activer la journalisation. Si aucun indicateur n'est spécifié, AWSTOE utilise la valeur par défaut. Pour plus d'informations sur les options de journalisation pour MSI,	Chaîne	Non	`*VX`	`i,w,e,a,r,u,c,m,o,p,v,x,+,!,*`

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
	consultez la section Options de ligne de commande dans la documentation du produit Microsoft Windows Installer.				
`logFile`	Un chemin absolu ou relatif vers l'emplacement du fichier journal. Si le chemin du fichier journal n'existe pas, il est créé. Si le chemin du fichier journal n'est pas fourni, AWSTOE ne stocke pas le journal d'installation MSI.	Chaîne	Non	N/A	N/A

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
`properties`	Paires clé-valeur des propriétés de journalisation MSI, par exemple : `TARGETDIR: "C:\target\location"` Remarque : La modification des propriétés suivantes n'est pas autorisée : • `REBOOT="ReallySuppress"` • `REINSTALLMODE="ecmus"` • `REINSTALL="ALL"`	Carte [String] Chaîne	Non	N/A	N/A

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
`ignoreAuthenticodeSignatureErrors`	Indicateur permettant d'ignorer les erreurs de validation de signature authenticode pour le programme d'installation spécifié dans le chemin. La Get-AuthenticodeSignature commande est utilisée pour valider les programmes d'installation. Paramètres : • `true`— Les erreurs de validation sont ignorées et le programme d'install	Booléen	Non	`false`	`true`, `false`

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
	ation s'exécute. • `false`— Les erreurs de validation ne sont pas ignorées. Le programme d'installation ne s'exécute que lorsque la validation est réussie. Il s'agit du comportement de par défaut.				

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
`allowUnsignedInstaller`	Indicateur permettant d'exécuter le programme d'installation non signé spécifié dans le chemin. La Get-AuthenticodeSignature commande est utilisée pour valider les programmes d'installation. Paramètres : • `true`— Ignore le `NotSigned` statut renvoyé par la Get-AuthenticodeSignature commande et exécute le programme	Booléen	Non	`false`	`true`, `false`

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
	d'install ation. • `false`— Nécessite la signature du programme d'install ation. Les programme s d'install ation non signés ne s'exécute ront pas. Il s'agit du comportem ent de par défaut.				

Exemples

Les exemples suivants montrent des variantes de la section de saisie de votre document de composant, en fonction de votre chemin d'installation.

Exemple de saisie : installation du chemin de document 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
```

Exemple de saisie : installation du chemin 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
```

Exemple de saisie : installation du chemin 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
```

Sortie

Voici un exemple de sortie du module `InstallMSI` d'action.

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

Désinstaller MSI

Le module `UninstallMSI` d'action permet de supprimer une application Windows à l'aide d'un fichier MSI. Vous pouvez spécifier l'emplacement du fichier MSI à l'aide d'un chemin de fichier local, d'un URI d'objet S3 ou d'une URL Web. L'option de redémarrage configure le comportement de redémarrage du système.

AWSTOE génère la msiexec commande en fonction des paramètres d'entrée du module d'action. L'emplacement du fichier MSI (`path`) et l'emplacement du fichier journal (`logFile`) sont explicitement placés entre guillemets («) lors de la génération de la msiexec commande.

Les codes de sortie MSI suivants sont considérés comme réussis :

- 0 (Succès)
- 1605 (ERROR_UNKNOWN_PRODUCT)
- 1614 (ERROR_PRODUCT_UNINSTALL)
- 1641 (redémarrage initié)
- 3010 (redémarrage requis)

Entrée

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
`path`	Spécifiez l'emplacement du fichier MSI à l'aide de l'une des méthodes suivantes : • Le chemin du fichier local. Le	Chaîne	Oui	N/A	N/A

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
	chemin peut être absolu ou relatif. • Un URI d'objet S3 valide. • Une URL HTTP/HTTPS Web valide (HTTPS est recommandé) conforme à la norme RFC 3986.				
	Les expressions de chaînage sont autorisées.				

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
`reboot`	Configure le comportement de redémarrage du système après une exécution réussie du module d'action. Paramètres : • `Force`— Lance un redémarrage du système une fois la msiexec commande exécutée avec succès. • `Allow`— Lance un redémarrage du système si la msiexec commande renvoie	Chaîne	Non	`Allow`	`Allow`, `Force`, `Skip`

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
	un code de sortie indiquant qu'un redémarrage est nécessaire. • `Skip`— Enregistre un message d'information dans le `console.log` fichier indiquant qu'un redémarrage a été ignoré. Cette option empêche le redémarrage, même si la msiexec commande renvoie un code de sortie indiquant qu'un				

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
	redémarrage est nécessaire.				

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
`logOptions`	Spécifiez les options à utiliser pour la journalisation de l'installation MSI. Les indicateurs spécifiés sont transmis au programme d'installation MSI, avec le paramètre de ligne de `/L` commande pour activer la journalisation. Si aucun indicateur n'est spécifié, AWSTOE utilise la valeur par défaut. Pour plus d'informations sur les options de journalisation pour MSI,	Chaîne	Non	`*VX`	`i,w,e,a,r,u,c,m,o,p,v,x,+,!,*`

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
	consultez la section Options de ligne de commande dans la documentation du produit Microsoft Windows Installer.				
`logFile`	Un chemin absolu ou relatif vers l'emplacement du fichier journal. Si le chemin du fichier journal n'existe pas, il est créé. Si le chemin du fichier journal n'est pas fourni, AWSTOE ne stocke pas le journal d'installation MSI.	Chaîne	Non	N/A	N/A

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
`properties`	Paires clé-valeur des propriétés de journalisation MSI, par exemple : `TARGETDIR: "C:\target\location"` Remarque : La modification des propriétés suivantes n'est pas autorisée : • `REBOOT="ReallySuppress"` • `REINSTALLMODE="ecmus"` • `REINSTALL="ALL"`	Carte [String] Chaîne	Non	N/A	N/A

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
`ignoreAuthenticodeSignatureErrors`	Indicateur permettant d'ignorer les erreurs de validation de signature authenticode pour le programme d'installation spécifié dans le chemin. La Get-AuthenticodeSignature commande est utilisée pour valider les programmes d'installation. Paramètres : • `true`— Les erreurs de validation sont ignorées et le programme d'install	Booléen	Non	`false`	`true`, `false`

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
	ation s'exécute. • `false`— Les erreurs de validation ne sont pas ignorées. Le programme d'installation ne s'exécute que lorsque la validation est réussie. Il s'agit du comportement de par défaut.				

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
`allowUnsignedInstaller`	Indicateur permettant d'exécuter le programme d'installation non signé spécifié dans le chemin. La Get-AuthenticodeSignature commande est utilisée pour valider les programmes d'installation. Paramètres : • `true`— Ignore le `NotSigned` statut renvoyé par la Get-AuthenticodeSignature commande et exécute le programme	Booléen	Non	`false`	`true`, `false`

Primitif	Description	Type	Obligatoire	Valeur par défaut	Valeurs acceptables
	d'install ation. • `false`— Nécessite la signature du programme d'install ation. Les programme s d'install ation non signés ne s'exécute ront pas. Il s'agit du comportem ent de par défaut.				

Exemples

Les exemples suivants montrent des variantes de la section de saisie de votre document de composant, en fonction de votre chemin d'installation.

Exemple de saisie : installation de suppression du chemin de document 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
```

Exemple de saisie : supprimer l'installation du chemin 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
```

Exemple de saisie : installation de suppression du chemin 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
```

Sortie

Voici un exemple de sortie du module `UninstallMSI` d'action.

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

Modules d'action du système

La section suivante décrit les modules d'action qui exécutent les commandes et instructions d'action du système de fichiers.

Modules d'action du système

- Redémarrer
- SetRegistry
- Mettre à jour le système d'exploitation

Redémarrer

Le module d'action Reboot redémarre l'instance. Il dispose d'une option configurable pour retarder le démarrage du redémarrage. Par défaut, `delaySeconds` est défini sur`0`, ce qui signifie qu'il n'y a aucun délai. Le délai d'expiration des étapes n'est pas pris en charge pour le module d'action Reboot, car il ne s'applique pas au redémarrage de l'instance.

Si l'application est appelée par l'agent Systems Manager, elle transmet le code de sortie (`3010`pour Windows, `194` pour Linux) à l'agent Systems Manager. L'agent Systems Manager gère le redémarrage du système comme décrit dans Reboot Managed Instance from Scripts.

Si l'application est invoquée sur l'hôte en tant que processus autonome, elle enregistre l'état d'exécution actuel, configure un déclencheur d'exécution automatique après le redémarrage pour réexécuter l'application après le redémarrage, puis redémarre le système.

Déclencheur d'exécution automatique après le redémarrage :

- Fenêtres. AWSTOE crée une entrée du planificateur de tâches Windows avec un déclencheur qui s'exécute automatiquement sur `SystemStartup`
- Linux. AWSTOE ajoute une tâche dans crontab qui s'exécute automatiquement après le redémarrage du système.

```
@reboot /download/path/awstoe run --document s3://bucket/key/doc.yaml
```

Ce déclencheur est nettoyé au démarrage de l'application.

Nouvelle tentative

Par défaut, le nombre maximum de tentatives est défini sur le Systems Manager`CommandRetryLimit`. Si le nombre de redémarrages dépasse la limite de nouvelles tentatives, l'automatisation échoue. Vous pouvez modifier la limite en modifiant le fichier de configuration de l'agent Systems Manager (`Mds.CommandRetryLimit`). Consultez la section Configuration du runtime dans l'agent open source de Systems Manager.

Pour utiliser le module d'action Redémarrer, pour les étapes contenant un redémarrage `exitcode` (par exemple,`3010`), vous devez exécuter le binaire de l'application sous le nom de`sudo user`.

Entrée

Primitif	Description	Type	Obligatoire	Par défaut
`delaySeconds`	Retarde un certain temps avant de lancer un redémarrage.	Entier	Non	`0`

Exemple de saisie : étape de redémarrage

```
name: RebootStep
action: Reboot
onFailure: Abort
maxAttempts: 2
inputs:
    delaySeconds: 60
```

Sortie

Aucune.

Lorsque le module Reboot est terminé, Image Builder passe à l'étape suivante de la génération.

SetRegistry

Le module SetRegistryd'action accepte une liste d'entrées et vous permet de définir la valeur de la clé de registre spécifiée. Si aucune clé de registre n'existe, elle est créée dans le chemin défini. Cette fonctionnalité s'applique uniquement à Windows.

Entrée

Primitif	Description	Type	Obligatoire
`path`	Chemin de la clé de registre.	Chaîne	Oui
`name`	Nom de la clé de registre.	Chaîne	Oui
`value`	Valeur de la clé de registre.	Chaîne, numéro, tableau	Oui
`type`	Type de valeur de la clé de registre.	Chaîne	Oui

Préfixes de chemin pris en charge

- `HKEY_CLASSES_ROOT / HKCR:`
- `HKEY_USERS / HKU:`
- `HKEY_LOCAL_MACHINE / HKLM:`
- `HKEY_CURRENT_CONFIG / HKCC:`
- `HKEY_CURRENT_USER / HKCU:`

Types pris en charge

- `BINARY`
- `DWORD`
- `QWORD`
- `SZ`
- `EXPAND_SZ`

- MULTI_SZ

Exemple de saisie : définir les valeurs des clés de registre

```
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
```

Sortie

Aucune.

Mettre à jour le système d'exploitation

Le module d'action UpdateOS prend en charge l'installation des mises à jour Windows et Linux. Il installe toutes les mises à jour disponibles par défaut. Vous pouvez également configurer une liste d'une ou de plusieurs mises à jour spécifiques à installer par le module d'action. Vous pouvez également spécifier les mises à jour à exclure de l'installation.

Si des listes « à inclure » et « à exclure » sont fournies, la liste des mises à jour qui en résulte ne peut inclure que celles répertoriées dans la liste « à inclure » qui ne figurent pas dans la liste « à exclure ».

> **Note**
>
> UpdateOS ne prend pas en charge Amazon Linux 2023 (AL2023). Nous vous recommandons de mettre à jour votre AMI de base vers la nouvelle version fournie avec chaque nouvelle version. Pour d'autres alternatives, consultez la section Contrôler les mises à jour reçues à partir des versions majeures et mineures dans le Guide de l'utilisateur Amazon Linux 2023.

- Fenêtres. Les mises à jour sont installées à partir de la source de mise à jour configurée sur la machine cible.

- Linux. L'application recherche le gestionnaire de packages pris en charge sur la plate-forme Linux et utilise l'un ou l'autre `yum` ou le gestionnaire de `apt-get` packages. Si aucun des deux n'est pris en charge, une erreur est renvoyée. Vous devez être `sudo` autorisé à exécuter le module d'action UpdateOS. Si vous n'avez pas d'`sudo`autorisations, un `error.Input` est renvoyé.

Entrée

Primitif	Description	Type	Obligatoire
`include`	Pour Windows, vous pouvez spécifier les éléments suivants : • Un ou plusieurs numéros d'article de la Base de connaissances Microsoft (KB) à inclure dans la liste des mises à jour qui peuvent être installées. Les formats valides sont `KB1234567` ou `1234567`. • Un nom de mise à jour utilisant une valeur générique (`*`). Les formats valides sont `Security*` ou `*Security*` . Pour Linux, vous pouvez spécifier un ou plusieurs packages à inclure dans la liste	Liste de chaînes	Non

Primitif	Description	Type	Obligatoire
	des mises à jour à installer.		
`exclude`	Pour Windows, vous pouvez spécifier les éléments suivants : • Un ou plusieurs numéros d'article de la Base de connaissances Microsoft (KB) à inclure dans la liste des mises à jour à exclure de l'installation. Les formats valides sont `KB1234567` ou `1234567`. • Un nom de mise à jour utilisant une valeur générique (*). Les formats valides sont : `Security*` ou`*Security*` . Pour Linux, vous pouvez spécifier un ou plusieurs packages à exclure de la liste des mises à jour à installer.	Liste de chaînes	Non

Exemple de saisie : ajout d'un support pour l'installation de mises à jour Linux

```
name: UpdateMyLinux
action: UpdateOS
onFailure: Abort
maxAttempts: 3
inputs:
    exclude:
        - ec2-hibinit-agent
```

Exemple de saisie : ajout d'un support pour l'installation des mises à jour Windows

```
name: UpdateWindowsOperatingSystem
action: UpdateOS
onFailure: Abort
maxAttempts: 3
inputs:
  include:
    - KB1234567
    - '*Security*'
```

Sortie

Aucune.

Configurer l'entrée pour la commande d' AWSTOE exécution

Pour rationaliser la saisie de votre commande sur la ligne de AWSTOE run commande, vous pouvez inclure les paramètres et options de commande dans un fichier de configuration d'entrée au format JSON avec une extension de `.json` fichier. AWSTOE peut lire votre fichier depuis l'un des emplacements suivants :

- Un chemin de fichier local (*. /config.json*).
- Un compartiment S3 (*s3 ://///config.json* <bucket-path><bucket-name>).

Lorsque vous entrez la run commande, vous pouvez spécifier le fichier de configuration d'entrée à l'aide du --config paramètre. Par exemple :

```
awstoe run --config <file-path>/config.json
```

Fichier de configuration d'entrée

Le fichier JSON de configuration d'entrée inclut des paires clé-valeur pour tous les paramètres que vous pouvez fournir directement via les paramètres et options de run commande. Si vous spécifiez un paramètre à la fois dans le fichier de configuration d'entrée et dans la run commande, en tant que paramètre ou option, les règles de priorité suivantes s'appliquent :

Règles de priorité

1. Un paramètre fourni directement à la run commande dans le AWS CLI, via un paramètre ou une option, remplace toute valeur définie dans le fichier de configuration d'entrée pour le même paramètre.
2. Un paramètre du fichier de configuration d'entrée remplace la valeur par défaut d'un composant.
3. Si aucun autre paramètre n'est transmis au document du composant, celui-ci peut appliquer une valeur par défaut, s'il en existe une.

Il existe deux exceptions à cette règle : les documents et les paramètres. Ces paramètres fonctionnent différemment dans la configuration d'entrée et en tant que paramètres de commande. Si vous utilisez le fichier de configuration d'entrée, vous ne devez pas spécifier ces paramètres directement dans la run commande. Cela générera une erreur.

Réglages des composants

Le fichier de configuration d'entrée contient les paramètres suivants. Pour rationaliser votre fichier, vous pouvez omettre les paramètres facultatifs qui ne sont pas nécessaires. Tous les paramètres sont facultatifs, sauf indication contraire.

- cwIgnoreFailures(Boolean) — Ignorez les échecs de journalisation dans les CloudWatch journaux.
- cwLogGroup(String) — Le `LogGroup` nom des CloudWatch journaux.
- cwLogRegion(String) — AWS Région qui s'applique aux CloudWatch journaux.
- cwLogStream(String) — Le `LogStream` nom des CloudWatch journaux, qui indique AWSTOE où diffuser le `console.log` fichier.
- DocumentS3 BucketOwner (String) : ID de compte du propriétaire du compartiment pour les documents basés sur l'URI S3.
- documents (tableau d'objets, obligatoire) — Un tableau d'objets JSON représentant les documents du composant YAML exécutés par la AWSTOE run commande. Au moins un document composant doit être spécifié.

Chaque objet comprend les champs suivants :

- path (String, obligatoire) — Emplacement du fichier du document du composant YAML. Il doit s'agir de l'une des options suivantes :
  - Un chemin de fichier local (*. /component-doc-example.yaml).*
  - Un URI S3 (`s3://`*bucket/key*).
  - *Un composant Image Builder construit la version ARN (arn:aws:imagebuilder:us-west-2:123456789012:component/ /2021.12.02/1). my-example-component*
- paramètres (tableau d'objets) — Tableau d'objets d'une paire clé-valeur, chacun représentant un paramètre spécifique au composant que la run commande transmet lorsqu'elle exécute le document du composant. Les paramètres sont facultatifs pour les composants. Le document du composant peut avoir des paramètres définis ou non.

  Chaque objet comprend les champs suivants :

  - name (String, obligatoire) — Le nom du paramètre du composant.
  - value (String, obligatoire) — La valeur à transmettre au document du composant pour le paramètre nommé.

  Pour en savoir plus sur les paramètres des composants, consultez la section Paramètres de la Définissez et référencez des variables dans AWSTOE page.
- ExecutonId (String) — Il s'agit de l'identifiant unique qui s'applique à l'exécution de la commande en cours. run Cet identifiant est inclus dans les noms des fichiers de sortie et des fichiers journaux, afin d'identifier ces fichiers de manière unique et de les lier à l'exécution de la commande en cours. Si ce paramètre est omis, AWSTOE génère un GUID.
- LogDirectory (String) — Le répertoire de destination où sont AWSTOE stockés tous les fichiers journaux de l'exécution de cette commande. Par défaut, ce répertoire se trouve dans le répertoire parent suivant : `T0E_<DATETIME>_<EXECUTIONID>` Si vous ne spécifiez pas le répertoire du journal, AWSTOE utilise le répertoire de travail actuel (`.`).
- LogS3 BucketName (chaîne) : si les journaux des composants sont stockés dans Amazon S3 (recommandé), AWSTOE télécharge les journaux des applications des composants dans le compartiment S3 nommé dans ce paramètre.
- LogS3 BucketOwner (chaîne) — Si les journaux des composants sont stockés dans Amazon S3 (recommandé), il s'agit de l'ID de compte propriétaire du compartiment dans lequel AWSTOE les fichiers journaux sont écrits.

- LogS3 KeyPrefix (chaîne) — Si les journaux des composants sont stockés dans Amazon S3 (recommandé), il s'agit du préfixe de clé d'objet S3 pour l'emplacement du journal dans le compartiment.
- paramètres (tableau d'objets) — Tableau d'objets d'une paire clé-valeur qui représente des paramètres qui s'appliquent globalement à tous les composants inclus dans l'exécution de la run commande en cours.
  - name (String, obligatoire) — Le nom du paramètre global.
  - value (String, obligatoire) — La valeur à transmettre à tous les documents du composant pour le paramètre nommé.
- phases (String) — Liste séparée par des virgules qui spécifie les phases à exécuter à partir des documents du composant YAML. Si le document d'un composant inclut des phases supplémentaires, celles-ci ne seront pas exécutées.
- StateDirectory (String) — Le chemin du fichier dans lequel les fichiers de suivi d'état sont stockés.
- trace (booléen) — Active la journalisation détaillée sur la console.

Exemples

L'exemple suivant montre un fichier de configuration d'entrée qui exécute les `test` phases `build` et pour deux documents de composants : `sampledoc.yaml` et`conversation-intro.yaml`. Chaque document composant possède un paramètre qui ne s'applique qu'à lui-même, et les deux utilisent un paramètre partagé. Le `project` paramètre s'applique aux deux documents du composant.

```
{
   "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>"
}
```

Composants gérés par le distributeur pour Windows

AWS Systems Manager Distributor vous aide à empaqueter et à publier des logiciels sur des nœuds AWS Systems Manager gérés. Vous pouvez empaqueter et publier vos propres logiciels ou utiliser le distributeur pour rechercher et publier les packages logiciels AWS fournis par les agents. Pour plus d'informations sur Systems Manager Distributor, consultez la section AWS Systems Manager Distributor dans le guide de AWS Systems Manager l'utilisateur.

Composants gérés pour le distributeur

Les composants gérés par Image Builder suivants utilisent AWS Systems Manager Distributor pour installer des packages d'applications sur des instances Windows.

- Le composant `distributor-package-windows` géré utilise AWS Systems Manager Distributor pour installer les packages d'applications que vous spécifiez sur votre instance de génération d'image Windows. Pour configurer les paramètres lorsque vous incluez ce composant dans votre recette, consultezConfiguration `distributor-package-windows` en tant que composant autonome.

- Le `aws-vss-components-windows` composant utilise AWS Systems Manager Distributor pour installer le `AwsVssComponents` package sur votre instance de génération d'image Windows. Pour configurer les paramètres lorsque vous incluez ce composant dans votre recette, consultezConfiguration `aws-vss-components-windows` en tant que composant autonome.

Pour plus d'informations sur l'utilisation des composants gérés dans votre recette Image Builder, consultez Création d'une nouvelle version d'une recette d'image les recettes d'images ou Création d'une nouvelle version d'une recette de conteneur les recettes de conteneurs. Pour plus d'informations sur le `AwsVssComponents` package, consultez la section Créer un instantané cohérent avec les applications VSS dans le guide de l'utilisateur Amazon EC2 pour les instances Windows.

Prérequis

Avant d'utiliser les composants Image Builder qui s'appuient sur Systems Manager Distributor pour installer des packages d'applications, vous devez vous assurer que les conditions préalables suivantes sont remplies.

- Les composants Image Builder qui utilisent Systems Manager Distributor pour installer des packages d'applications sur votre instance doivent être autorisés à appeler l'API Systems Manager. Avant d'utiliser les composants d'une recette Image Builder, vous devez créer la politique et le rôle IAM qui accordent l'autorisation. Pour configurer les autorisations, consultezConfigurer les autorisations du distributeur Systems Manager.

> **Note**
>
> Image Builder ne prend actuellement pas en charge les packages Systems Manager Distributor qui redémarrent l'instance. Par exemple, les packages`AWSNVMe`,`AWSPVDrivers`, et `AwsEnaNetworkDriver` Distributor redémarrent l'instance et ne sont donc pas autorisés.

Configurer les autorisations du distributeur Systems Manager

Le `distributor-package-windows` composant et les autres composants qui l'utilisent, tels que`aws-vss-components-windows`, nécessitent une autorisation supplémentaire sur l'instance de build pour s'exécuter. L'instance de build doit être en mesure d'appeler l'API Systems Manager pour démarrer une installation de Distributor et demander le résultat.

Suivez ces procédures AWS Management Console pour créer une politique et un rôle IAM personnalisés autorisant les composants Image Builder à installer les packages Systems Manager Distributor à partir de l'instance de build.

Étape 1 : créer une politique

Créez une politique IAM pour les autorisations des distributeurs.

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le volet de navigation, sélectionnez Politiques, puis Créer une politique.
3. Sur la page Créer une politique, choisissez l'onglet JSON, puis remplacez le contenu par défaut par la politique JSON suivante, en remplaçant la partition, la région et l'ID de compte si nécessaire, ou en utilisant des caractères génériques.

```
{
 "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. Choisissez Examiner une stratégie.

5. Pour Nom, saisissez un nom pour identifier la stratégie, par exemple *InvokeDistributor* ou tout autre nom de votre choix.
6. (Facultatif) Pour Description, saisissez une description de l'objectif du rôle.
7. Choisissez Créer une stratégie.

Étape 2 : créer un rôle

Créez un rôle IAM pour les autorisations de distributeur.

1. Dans le volet de navigation de la console IAM, choisissez Roles, puis Create role.
2. Sous Select type of trusted entity (Sélectionner le type d'entité approuvée), choisissez Service AWS.
3. Immédiatement sous Choisir le service qui utilisera ce rôle, choisissez EC2, puis Suivant : Autorisations.
4. Sous Sélectionner votre cas d'utilisation, choisissez EC2, puis Suivant : Autorisations.
5. Dans la liste des politiques, cochez la case à côté d'AmazonSSM ManagedInstanceCore. (Entrez `SSM` dans la zone de recherche si vous avez besoin d'une liste plus petite.)
6. Dans cette liste de politiques, cochez la case à côté de EC2. InstanceProfileForImageBuilder (Entrez `ImageBuilder` dans la zone de recherche si vous avez besoin d'une liste plus petite.)
7. Choisissez Next: Tags (Suivant : Balises).
8. (Facultatif) Ajoutez une ou plusieurs paires clé-valeur de balise pour organiser, suivre ou contrôler l'accès pour ce rôle, puis choisissez Suivant : Révision.
9. Pour Nom du rôle, entrez un nom pour le rôle, par exemple *InvokeDistributor* ou autre, en fonction de vos préférences.
10. (Facultatif) Pour Description du rôle, remplacez le texte par défaut par une description de l'objectif de ce rôle.
11. Sélectionnez Créer un rôle. Le système vous renvoie à la page Rôles.

Étape 3 : associer la politique au rôle

La dernière étape pour configurer vos autorisations de distributeur consiste à associer la politique IAM au rôle IAM.

1. Sur la page Rôles de la console IAM, choisissez le rôle que vous venez de créer. La page Résumé du rôle s'ouvre.

2. Choisissez Attach Policies (Attacher des stratégies).
3. Recherchez la politique que vous avez créée dans la procédure précédente et cochez la case à côté du nom.
4. Choisissez Attach policy (Attacher une politique).

Utilisez ce rôle dans la ressource de configuration de l'infrastructure Image Builder pour toute image incluant des composants utilisant Systems Manager Distributor. Pour plus d'informations, consultez Créer une configuration d'infrastructure.

Configuration **`distributor-package-windows`** en tant que composant autonome

Pour utiliser le `distributor-package-windows` composant dans une recette, définissez les paramètres suivants qui configurent le package à installer.

> **Note**
>
> Avant d'utiliser le `distributor-package-windows` composant dans une recette, vous devez vous assurer que toutes les conditions Prérequis sont respectées.

- Action (obligatoire) — Spécifiez si vous souhaitez installer ou désinstaller le package. Les valeurs valides sont `Install` et `Uninstall`. La valeur par défaut est. `Install`
- PackageName(Obligatoire) : nom du package du distributeur à installer ou à désinstaller. Pour obtenir la liste des noms de packages valides, consultezTrouver des packages pour distributeurs.
- PackageVersion(Facultatif) — Version du package du distributeur à installer. PackageVersion par défaut, c'est la version recommandée.
- AdditionalArguments(Facultatif) — Chaîne JSON contenant les paramètres supplémentaires à fournir à votre script pour installer, désinstaller ou mettre à jour un package. Pour plus d'informations, consultez AdditionalArguments dans la section AWS:ConfigurePackage Inputs de la page de référence du plug-in du document Systems Manager Command.

Configuration **aws-vss-components-windows** en tant que composant autonome

Lorsque vous utilisez le `aws-vss-components-windows` composant dans une recette, vous pouvez éventuellement définir le `PackageVersion` paramètre pour utiliser une version spécifique du `AwsVssComponents` package. Lorsque vous omettez ce paramètre, le composant utilise par défaut la version recommandée du `AwsVssComponents` package.

> **Note**
>
> Avant d'utiliser le `aws-vss-components-windows` composant dans une recette, vous devez vous assurer que toutes les conditions Prérequis sont respectées.

Trouver des packages pour distributeurs

Amazon et des tiers fournissent des packages publics que vous pouvez installer avec Systems Manager Distributor.

Pour consulter les packages disponibles dans le AWS Management Console, connectez-vous à la AWS Systems Manager console et choisissez Distributor dans le volet de navigation. La page Distributeur affiche tous les packages mis à votre disposition. Pour plus d'informations sur la liste des packages disponibles avec le AWS CLI, voir Afficher les packages (ligne de commande) dans le guide de AWS Systems Manager l'utilisateur.

Vous pouvez également créer vos propres packages privés de distribution Systems Manager. Pour plus d'informations, voir Création d'un package dans le guide de AWS Systems Manager l'utilisateur.

composants de durcissement CIS

Le Center for Internet Security (CIS) est une organisation communautaire à but non lucratif. Leurs experts en cybersécurité travaillent ensemble pour élaborer des directives de sécurité informatique qui protègent les organisations publiques et privées contre les cybermenaces. Leur ensemble de bonnes pratiques mondialement reconnu, connu sous le nom de CIS Benchmarks, aide les organisations informatiques du monde entier à configurer leurs systèmes en toute sécurité. Pour les articles tendance, les articles de blog, les podcasts, les webinaires et les livres blancs, consultez CIS Insights sur le site Web du Center for Internet Security.

Evaluations CIS

CIS crée et gère un ensemble de directives de configuration, connues sous le nom de benchmarks CIS, qui fournissent les meilleures pratiques de configuration pour des technologies spécifiques, notamment les systèmes d'exploitation, les plateformes cloud, les applications, les bases de données, etc. Les benchmarks CIS sont reconnus comme une norme industrielle par des organisations et des normes telles que PCI DSS, HIPAA, DoD Cloud Computing SRG, FISMA, DFARS et FEDRAMP. Pour en savoir plus, consultez la section CIS Benchmarks sur le site Web du Center for Internet Security.

composants de durcissement CIS

Lorsque vous vous abonnez à une image renforcée CIS dans AWS Marketplace, vous avez également accès au composant de renforcement associé qui exécute un script pour appliquer les directives CIS Benchmarks de niveau 1 à votre configuration. L'organisation CIS possède et entretient les composants de durcissement CIS afin de s'assurer qu'ils reflètent les dernières directives.

> **Note**
>
> Les composants de durcissement CIS ne respectent pas les règles de classement des composants standard des recettes Image Builder. Les composants de renforcement CIS sont toujours exécutés en dernier pour garantir que les tests de référence s'exécutent par rapport à votre image de sortie.

Amazon a géré les composants de renforcement STIG pour EC2 Image Builder

Les guides de mise en œuvre technique de sécurité (STIG) sont des normes de renforcement de la configuration créées par la Defense Information Systems Agency (DISA) pour sécuriser les systèmes d'information et les logiciels. Pour rendre vos systèmes conformes aux normes STIG, vous devez installer, configurer et tester différents paramètres de sécurité.

Image Builder fournit des composants de renforcement STIG pour vous aider à créer plus efficacement des images conformes aux normes STIG de base. Ces composants STIG détectent les erreurs de configuration et exécutent un script de correction. L'utilisation de composants conformes aux normes STIG est gratuite.

> **Important**
>
> À quelques exceptions près, les composants de renforcement STIG n'installent pas de packages tiers. Si des packages tiers sont déjà installés sur l'instance, et s'il existe des STIG connexes pris en charge par Image Builder pour ce package, le composant de renforcement les applique.

Cette page répertorie tous les STIG pris en charge par Image Builder et appliqués aux instances EC2 lancées par Image Builder lorsque vous créez et testez une nouvelle image. Si vous souhaitez appliquer des paramètres STIG supplémentaires à votre image, vous pouvez créer un composant personnalisé pour la configurer. Pour plus d'informations sur les composants personnalisés et sur la façon de les créer, consultezGérez les composants avec Image Builder.

Lorsque vous créez une image, les composants de durcissement STIG enregistrent si les STIG pris en charge sont appliqués ou ignorés. Nous vous recommandons de consulter les journaux Image Builder de vos images qui utilisent des composants de renforcement STIG. Pour plus d'informations sur la façon d'accéder aux journaux d'Image Builder et de les consulter, consultezRésoudre les problèmes liés aux builds de pipelines.

Niveaux de conformité

- Élevé (Catégorie I)

  Risque le plus grave. Inclut toute vulnérabilité pouvant entraîner une perte de confidentialité, de disponibilité ou d'intégrité.
- Moyen (Catégorie II)

  Inclut toute vulnérabilité susceptible d'entraîner une perte de confidentialité, de disponibilité ou d'intégrité, mais les risques peuvent être atténués.
- Faible (catégorie III)

  Toute vulnérabilité qui dégrade les mesures de protection contre la perte de confidentialité, de disponibilité ou d'intégrité.

Rubriques

- Composants de renforcement Windows STIG
- Journal d'historique des versions de STIG pour Windows

- Composants de renforcement Linux STIG
- Journal d'historique des versions de STIG pour Linux
- Composant de validation de conformité SCAP

Composants de renforcement Windows STIG

AWSTOE Les composants de renforcement de Windows STIG sont conçus pour les serveurs autonomes et appliquent une stratégie de groupe locale. Les composants de renforcement conformes aux STIG sont installés InstallRoot par le ministère de la Défense (DoD) sur l'infrastructure Windows pour télécharger, installer et mettre à jour les certificats du DoD. Ils suppriment également les certificats inutiles pour maintenir la conformité aux STIG. Actuellement, les lignes de base STIG sont prises en charge pour les versions suivantes de Windows Server : 2012 R2, 2016, 2019 et 2022.

Cette section répertorie les paramètres actuels de chacun des composants de renforcement de Windows STIG, suivis d'un historique des versions.

STIG-Build-Windows-Low version 2022.4.x

La liste suivante contient les paramètres STIG que le composant de renforcement applique à votre infrastructure. Si aucun paramètre pris en charge n'est applicable à votre infrastructure, le composant de renforcement ignore ce paramètre et passe à autre chose. Par exemple, certains paramètres STIG peuvent ne pas s'appliquer aux serveurs autonomes. Les politiques spécifiques à l'organisation peuvent également affecter les paramètres appliqués par le composant de renforcement, par exemple l'obligation pour les administrateurs de revoir les paramètres du document.

Pour obtenir la liste complète des paramètres STIG Windows, consultez la Bibliothèque de documents STIG. Pour plus d'informations sur l'affichage de la liste complète, veuillez consulter la rubrique Outils d'affichage STIG.

- Windows Server 2022 STIG version 1, version 1

  V-254335, V-254336, V-254337, V-254338, V-254351, V-254357, V-254363 et V-254481
- Windows Server 2019 STIG version 2 version 5

  V-205691, V-205819, V-205858, V-205859, V-205860, V-205870, V-205871 et V-205923
- Windows Server 2016 STIG version 2 version 5

  V-224916, V-224917, V-224918, V-224919, V-224931, V-224942 et V-225060

- Windows Server 2012 R2 MS STIG version 3 version 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 et V-225250

- Microsoft .NET Framework 4.0 STIG version 2, version 2

  Aucun paramètre STIG n'est appliqué au Microsoft .NET Framework pour les vulnérabilités de catégorie III.

- Pare-feu Windows STIG version 2, version 1

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

- Internet Explorer 11 STIG version 2 version 3

  V-46477, V-46629 et V-97527

- Microsoft Edge STIG version 1 version 6 (Windows Server 2022 uniquement)

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

STIG-Build-Windows-Medium version 2022.4.x

La liste suivante contient les paramètres STIG que le composant de renforcement applique à votre infrastructure. Si aucun paramètre pris en charge n'est applicable à votre infrastructure, le composant de renforcement ignore ce paramètre et passe à autre chose. Par exemple, certains paramètres STIG peuvent ne pas s'appliquer aux serveurs autonomes. Les politiques spécifiques à l'organisation peuvent également affecter les paramètres appliqués par le composant de renforcement, par exemple l'obligation pour les administrateurs de revoir les paramètres du document.

Pour obtenir la liste complète des paramètres STIG Windows, consultez la Bibliothèque de documents STIG. Pour plus d'informations sur l'affichage de la liste complète, veuillez consulter la rubrique Outils d'affichage STIG.

> **Note**
>
> Les composants de renforcement STIG-Build-Windows-Medium incluent tous les paramètres STIG répertoriés qui AWSTOE s'appliquent aux composants de renforcement faible STIG-Build-Windows-Low, en plus des paramètres STIG répertoriés spécifiquement pour les vulnérabilités de catégorie II.

- Windows Server 2022 STIG version 1, version 1

  Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique aux vulnérabilités de catégorie III (faible), ainsi que :

  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-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-254315, V-254316, V-254317, V-254318, V-254319, V-254320, V-254321, V-254322, V-254323, V-254324, V-254325, 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-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-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-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-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 et V-254512
- Windows Server 2019 STIG version 2 version 5

  Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique aux vulnérabilités de catégorie III (faible), ainsi que :

  V-205625, V-205626, V-205627, V-205629, V-205630, V-205633, V-205634, 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-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-205712, V-205716 5720, 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-205795, V-205796, V-205757 97, 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-205826, V-205821 5827, 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-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 et V-236001

- Windows Server 2016 STIG version 2 version 5

  Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique aux vulnérabilités de catégorie III (faible), ainsi que :

  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-224882, V-224872 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-224902, V-224902 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-224951, V-224947 52, 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-225022 23, 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 et V-236000

- Windows Server 2012 R2 MS STIG version 3 version 5

Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique aux vulnérabilités de catégorie III (faible), ainsi que :

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-225455 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-225344, V-225344 341, 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-225299, V-225298, V-225297, V-225296, V-225295, V-225294, V-225294 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 et V-225259 239

- Microsoft .NET Framework 4.0 STIG version 2, version 2

  Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique aux vulnérabilités de catégorie III (faibles), ainsi que le V-225238

- Pare-feu Windows STIG version 2, version 1

  Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique aux vulnérabilités de catégorie III (faible), ainsi que :

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

- Internet Explorer 11 STIG version 2 version 3

Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique aux vulnérabilités de catégorie III (faible), ainsi que :

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 et V-75171

- Microsoft Edge STIG version 1 version 6 (Windows Server 2022 uniquement)

  Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique aux vulnérabilités de catégorie III (faible), ainsi que :

  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 et V-246736

- Defender STIG version 2 version 4 (Windows Server 2022 uniquement)

  Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique aux vulnérabilités de catégorie III (faible), ainsi que :

  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-213446, V-213447, V-213448, V-213448, V-213448 V-213450, V-213451, V-213455, V-213464, V-213465 et V-213466

STIG-Build-Windows-High version 2022.4.x

La liste suivante contient les paramètres STIG que le composant de renforcement applique à votre infrastructure. Si aucun paramètre pris en charge n'est applicable à votre infrastructure, le composant de renforcement ignore ce paramètre et passe à autre chose. Par exemple, certains paramètres STIG peuvent ne pas s'appliquer aux serveurs autonomes. Les politiques spécifiques à l'organisation peuvent également affecter les paramètres appliqués par le composant de renforcement, par exemple l'obligation pour les administrateurs de revoir les paramètres du document.

Pour obtenir la liste complète des paramètres STIG Windows, consultez la Bibliothèque de documents STIG. Pour plus d'informations sur l'affichage de la liste complète, veuillez consulter la rubrique Outils d'affichage STIG.

> **Note**
>
> Les composants de renforcement STIG-Build-Windows-High incluent tous les paramètres STIG répertoriés qui AWSTOE s'appliquent aux composants de renforcement STIG-Build-Windows-Low et STIG-Build-Windows-Medium, en plus des paramètres STIG répertoriés spécifiquement pour les vulnérabilités de catégorie I.

- Windows Server 2022 STIG version 1, version 1

  Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique aux vulnérabilités de catégories II et III (moyennes et faibles), ainsi que :

  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 et V-254500
- Windows Server 2019 STIG version 2 version 5

  Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique aux vulnérabilités de catégories II et III (moyennes et faibles), ainsi que :

  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 et V-205919
- Windows Server 2016 STIG version 2 version 5

  Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique aux vulnérabilités de catégories II et III (moyennes et faibles), ainsi que :

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 et V-225079

- Windows Server 2012 R2 MS STIG version 3 version 5

  Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique aux vulnérabilités de catégories II et III (moyennes et faibles), ainsi que :

  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 et V-225274
- Microsoft .NET Framework 4.0 STIG version 2, version 2

  Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique pour les vulnérabilités de catégories II et III (moyennes et faibles) dans le Microsoft .NET Framework. Aucun paramètre STIG supplémentaire ne s'applique aux vulnérabilités de catégorie I.
- Pare-feu Windows STIG version 2, version 1

  Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique aux vulnérabilités de catégories II et III (moyennes et faibles), ainsi que :

  V-241992, V-241997 et V-242002
- Internet Explorer 11 STIG version 2 version 3

  Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique pour les vulnérabilités de catégories II et III (moyennes et faibles) dans Internet Explorer 11. Aucun paramètre STIG supplémentaire ne s'applique aux vulnérabilités de catégorie I.
- Microsoft Edge STIG version 1 version 6 (Windows Server 2022 uniquement)

  Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique aux vulnérabilités de catégories II et III (moyennes et faibles), ainsi que :

  V-235758 et V-235759
- Defender STIG version 2 version 4 (Windows Server 2022 uniquement)

  Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique aux vulnérabilités de catégories II et III (moyennes et faibles), ainsi que :

  V-213426, V-213452 et V-213453

Journal d'historique des versions de STIG pour Windows

Cette section enregistre l'historique des versions des composants de renforcement de Windows pour les mises à jour trimestrielles de STIG. Pour voir les modifications et les versions publiées pendant un trimestre, choisissez le titre pour développer les informations.

Changements du premier trimestre 2024 - 02/06/2024 (aucun changement) :

Aucune modification n'a été apportée au composant Windows STIGS pour la version du premier trimestre 2024.

Changements du quatrième trimestre 2023 - 12/04/2023 (aucun changement) :

Aucune modification n'a été apportée au composant Windows STIGS pour la version du quatrième trimestre 2023.

Changements du troisième trimestre 2023 - 10/04/2023 (aucun changement) :

Aucune modification n'a été apportée au composant Windows STIGS pour la version du troisième trimestre 2023.

Changements du deuxième trimestre 2023 - 05/03/2023 (aucun changement) :

Aucune modification n'a été apportée au composant Windows STIGS pour la version du deuxième trimestre 2023.

Changements du premier trimestre 2023 - 27/03/2023 (aucun changement) :

Aucune modification n'a été apportée au composant Windows STIGS pour la version du premier trimestre 2023.

Changements du quatrième trimestre 2022 - 01/02/2023 :

Versions STIG mises à jour et STIGS appliquées pour la version du quatrième trimestre 2022 comme suit :

STIG-Build-Windows-Low version 2022.4.x

- Windows Server 2022 utilisant STIG version 1 sortie 1
- Windows Server 2019 utilisant STIG version 2 sortie 5

- Windows Server 2016 utilisant STIG version 2 sortie 5
- Windows Server 2012 R2 MS utilisant STIG version 3 sortie 5
- Microsoft .NET Framework 4.0 utilisant STIG version 2 sortie 2
- Pare-feu Windows utilisant STIG version 2 sortie 1
- Internet Explorer 11 utilisant STIG version 2 sortie 3
- Microsoft Edge STIG version 1 version 6 (Windows Server 2022 uniquement)

STIG-Build-Windows-Medium version 2022.4.x

- Windows Server 2022 utilisant STIG version 1 sortie 1
- Windows Server 2019 utilisant STIG version 2 sortie 5
- Windows Server 2016 utilisant STIG version 2 sortie 5
- Windows Server 2012 R2 MS utilisant STIG version 3 sortie 5
- Microsoft .NET Framework 4.0 utilisant STIG version 2 sortie 2
- Pare-feu Windows utilisant STIG version 2 sortie 1
- Internet Explorer 11 utilisant STIG version 2 sortie 3
- Microsoft Edge STIG version 1 version 6 (Windows Server 2022 uniquement)
- Defender STIG version 2 version 4 (Windows Server 2022 uniquement)

STIG-Build-Windows-High version 2022.4.x

- Windows Server 2022 utilisant STIG version 1 sortie 1
- Windows Server 2019 utilisant STIG version 2 sortie 5
- Windows Server 2016 utilisant STIG version 2 sortie 5
- Windows Server 2012 R2 MS utilisant STIG version 3 sortie 5
- Microsoft .NET Framework 4.0 utilisant STIG version 2 sortie 2
- Pare-feu Windows utilisant STIG version 2 sortie 1
- Internet Explorer 11 utilisant STIG version 2 sortie 3
- Microsoft Edge STIG version 1 version 6 (Windows Server 2022 uniquement)
- Defender STIG version 2 version 4 (Windows Server 2022 uniquement)

Changements du troisième trimestre 2022 - 30/09/2022 (aucun changement) :

Aucune modification n'a été apportée au composant Windows STIGS pour la version du troisième trimestre 2022.

Changements du deuxième trimestre 2022 - 08/02/2022 :

Versions STIG mises à jour et STIGS appliqués pour la version du deuxième trimestre 2022.

STIG-Build-Windows-Low version 1.5.x

- Windows Server 2019 STIG version 2 version 4
- Windows Server 2016 STIG version 2 version 4
- Windows Server 2012 R2 MS STIG version 3 version 3
- Microsoft .NET Framework 4.0 STIG version 2 version 1
- Pare-feu Windows utilisant STIG version 2 sortie 1
- Internet Explorer 11 STIG version 1, version 19

STIG-Build-Windows-Medium version 1.5.x

- Windows Server 2019 STIG version 2 version 4
- Windows Server 2016 STIG version 2 version 4
- Windows Server 2012 R2 MS STIG version 3 version 3
- Microsoft .NET Framework 4.0 STIG version 2 version 1
- Pare-feu Windows utilisant STIG version 2 sortie 1
- Internet Explorer 11 STIG version 1, version 19

STIG-Build-Windows-High version 1.5.x

- Windows Server 2019 STIG version 2 version 4
- Windows Server 2016 STIG version 2 version 4
- Windows Server 2012 R2 MS STIG version 3 version 3
- Microsoft .NET Framework 4.0 STIG version 2 version 1
- Pare-feu Windows utilisant STIG version 2 sortie 1

- Internet Explorer 11 STIG version 1, version 19

Changements du premier trimestre 2022 - 08/02/2022 (aucun changement) :

Aucune modification n'a été apportée au composant Windows STIGS pour la version du premier trimestre 2022.

Changements du quatrième trimestre 2021 - 20/12/2021 :

Versions STIG mises à jour et STIGS appliquées pour la version du quatrième trimestre 2021.

STIG-Build-Windows-Low version 1.5.x

- Windows Server 2019 STIG version 2 version 3
- Windows Server 2016 STIG version 2 version 3
- Windows Server 2012 R2 MS STIG version 3 version 3
- Microsoft .NET Framework 4.0 STIG version 2 version 1
- Pare-feu Windows utilisant STIG version 2 sortie 1
- Internet Explorer 11 STIG version 1, version 19

STIG-Build-Windows-Medium version 1.5.x

- Windows Server 2019 STIG version 2 version 3
- Windows Server 2016 STIG version 2 version 3
- Windows Server 2012 R2 MS STIG version 3 version 3
- Microsoft .NET Framework 4.0 STIG version 2 version 1
- Pare-feu Windows utilisant STIG version 2 sortie 1
- Internet Explorer 11 STIG version 1, version 19

STIG-Build-Windows-High version 1.5.x

- Windows Server 2019 STIG version 2 version 3
- Windows Server 2016 STIG version 2 version 3
- Windows Server 2012 R2 MS STIG version 3 version 3

- Microsoft .NET Framework 4.0 STIG version 2 version 1
- Pare-feu Windows utilisant STIG version 2 sortie 1
- Internet Explorer 11 STIG version 1, version 19

Changements du troisième trimestre 2021 au 30/09/2021 :

Versions STIG mises à jour et STIGS appliquées pour la version du troisième trimestre 2021.

Version 1.4.x de STIG-Build-Windows-Low

- Windows Server 2019 STIG version 2 version 2
- Windows Server 2016 STIG version 2 version 2
- Windows Server 2012 R2 MS STIG version 3 version 2
- Microsoft .NET Framework 4.0 STIG version 2 version 1
- Pare-feu Windows STIG version 1 version 7
- Internet Explorer 11 STIG version 1, version 19

STIG-Build-Windows-Medium version 1.4.x

- Windows Server 2019 STIG version 2 version 2
- Windows Server 2016 STIG version 2 version 2
- Windows Server 2012 R2 MS STIG version 3 version 2
- Microsoft .NET Framework 4.0 STIG version 2 version 1
- Pare-feu Windows STIG version 1 version 7
- Internet Explorer 11 STIG version 1, version 19

STIG-Build-Windows-High version 1.4.x

- Windows Server 2019 STIG version 2 version 2
- Windows Server 2016 STIG version 2 version 2
- Windows Server 2012 R2 MS STIG version 3 version 2
- Microsoft .NET Framework 4.0 STIG version 2 version 1
- Pare-feu Windows STIG version 1 version 7

- Internet Explorer 11 STIG version 1, version 19

Composants de renforcement Linux STIG

Cette section contient des informations sur les composants de renforcement de Linux STIG, suivies d'un journal de l'historique des versions. Si la distribution Linux ne possède pas ses propres paramètres STIG, le composant de renforcement applique les paramètres RHEL. Le composant de renforcement applique les paramètres STIG pris en charge à l'infrastructure basée sur la distribution Linux, comme suit :

Paramètres STIG de Red Hat Enterprise Linux (RHEL) 7

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

Paramètres RHEL 8 STIG

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

Version 2024.1.x de STIG-Build-Linux-Low

La liste suivante contient les paramètres STIG que le composant de renforcement applique à votre infrastructure. Si aucun paramètre pris en charge n'est applicable à votre infrastructure, le composant de renforcement ignore ce paramètre et passe à autre chose. Par exemple, certains paramètres STIG peuvent ne pas s'appliquer aux serveurs autonomes. Les politiques spécifiques à l'organisation peuvent également affecter les paramètres appliqués par le composant de renforcement, par exemple l'obligation pour les administrateurs de revoir les paramètres du document.

Pour obtenir une liste complète, veuillez consulter STIGs Document Library. Pour plus d'informations sur l'affichage de la liste complète, veuillez consulter la rubrique Outils d'affichage STIG.

RHEL 7 STIG Version 3, version 14

- RHEL 7/CentOS 7

V-204452, V-204576 et V-204605

- AL2

  V-204452, V-204576 et V-204605

RHEL 8 STIG Version 1 Version 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-230498, V-230499 et V-230281

Ubuntu 18.04 STIG version 2, version 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 et V-219333

Ubuntu 20.04 STIG version 1 version 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 et V-238308

STIG-Build-Linux-Medium version 2024.1.x

La liste suivante contient les paramètres STIG que le composant de renforcement applique à votre infrastructure. Si aucun paramètre pris en charge n'est applicable à votre infrastructure, le composant de renforcement ignore ce paramètre et passe à autre chose. Par exemple, certains paramètres STIG peuvent ne pas s'appliquer aux serveurs autonomes. Les politiques spécifiques à l'organisation peuvent également affecter les paramètres appliqués par le composant de renforcement, par exemple l'obligation pour les administrateurs de revoir les paramètres du document.

Pour obtenir une liste complète, veuillez consulter STIGs Document Library. Pour plus d'informations sur l'affichage de la liste complète, veuillez consulter la rubrique Outils d'affichage STIG.

> **Note**
>
> Les composants de renforcement STIG-Build-Linux-Medium incluent tous les paramètres STIG répertoriés qui AWSTOE s'appliquent aux composants à faible durcissement STIG-

Build-Linux-Low, en plus des paramètres STIG répertoriés spécifiquement pour les vulnérabilités de catégorie II.

RHEL 7 STIG Version 3, version 14

Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique aux vulnérabilités de catégorie III (faible) pour cette distribution Linux, ainsi que :

- 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-204598, 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-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-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-204566, V-204567, 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-2045, V-204631, V-204633 et V-256970

- TOUS LES 2 :

  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-204598, 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-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-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-204566, V-204567, 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-2045, V-204631, V-204633 et V-256970

RHEL 8 STIG Version 1 Version 13

Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique aux vulnérabilités de catégorie III (faible) pour cette distribution Linux, ainsi que :

- 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-230233, V-230324, 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-230538, V-230539, V-230540, V-230541, V-230542, V-230542, V-230537, V-230538, V-230539, V-230540, V-230541, V-230542, V-230537, V-230538, V-230539, V-230540, V-230541, V-230542, V-230542, V-230537, V-230538, V-230539, V-230540, V-230541, V-230542, V-230542, V-543, 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, V-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-230345, V-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-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-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-230559, V-230560, V-230560, V-230478 561, V-237640 et V-256974

Ubuntu 18.04 STIG version 2, version 13

Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique aux vulnérabilités de catégorie III (faible) pour cette distribution Linux, ainsi que :

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-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-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-219235, V-219236, V-219238, V-219239, V-219240, V-219241, V-219242, V-219243, V-2194, 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-219276, V-219277, V-219279, V-219281, V-219287, V-219291, V-219297, V-219298, 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-219337 et V-219335

Ubuntu 20.04 STIG version 1 version 11

Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique aux vulnérabilités de catégorie III (faible) pour cette distribution Linux, ainsi que :

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-238240, V-238241, V-23824, V-238244, V-238245, V-238246, V-238247, V-238248, V-238249, V-238250, V-238251, V-238252, V-238253, V-238254, V-238255, V-238256, V-238257, V-238258, V-238264, V-238268, V-238271, V-238277772, 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 et V-238334

STIG-Build-Linux-High version 2024.1.x

La liste suivante contient les paramètres STIG que le composant de renforcement applique à votre infrastructure. Si aucun paramètre pris en charge n'est applicable à votre infrastructure, le composant de renforcement ignore ce paramètre et passe à autre chose. Par exemple, certains paramètres

STIG peuvent ne pas s'appliquer aux serveurs autonomes. Les politiques spécifiques à l'organisation peuvent également affecter les paramètres appliqués par le composant de renforcement, par exemple l'obligation pour les administrateurs de revoir les paramètres du document.

Pour obtenir une liste complète, veuillez consulter STIGs Document Library. Pour plus d'informations sur l'affichage de la liste complète, veuillez consulter la rubrique Outils d'affichage STIG.

> **Note**
>
> Les composants de renforcement STIG-Build-Linux-High incluent tous les paramètres STIG répertoriés qui AWSTOE s'appliquent aux composants de renforcement STIG-Build-Linux-Low et STIG-Build-Linux-Medium, en plus des paramètres STIG répertoriés qui s'appliquent spécifiquement aux vulnérabilités de catégorie I.

RHEL 7 STIG Version 3, version 14

Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique pour les vulnérabilités de catégories II et III (moyennes et faibles) de cette distribution Linux, ainsi que :

- RHEL 7/CentOS 7

  V-204425, V-204594, V-204455, V-204424, V-204442, V-204443, V-204447, V-204448, V-204502, V-204620 et V-204621
- TOUS LES 2 :

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

RHEL 8 STIG Version 1 Version 13

Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique pour les vulnérabilités de catégories II et III (moyennes et faibles) de cette distribution Linux, ainsi que :

- RHEL 8/CentOS 8/AL 2023

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

Ubuntu 18.04 STIG version 2, version 13

Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique pour les vulnérabilités de catégories II et III (moyennes et faibles) de cette distribution Linux, ainsi que :

V-219157, V-219158, V-219177, V-219212 V-219308, V-219314, V-219316 et V-251507

Ubuntu 20.04 STIG version 1 version 11

Inclut tous les paramètres STIG pris en charge que le composant de renforcement applique pour les vulnérabilités de catégories II et III (moyennes et faibles) de cette distribution Linux, ainsi que :

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

Journal d'historique des versions de STIG pour Linux

Cette section enregistre l'historique des versions des composants Linux. Pour voir les modifications et les versions publiées pendant un trimestre, choisissez le titre pour développer les informations.

Changements du 1er trimestre 2024 - 02/06/2024 :

Versions STIG mises à jour et STIGS appliquées pour la version du premier trimestre 2024 comme suit :

Version 2024.1.x de STIG-Build-Linux-Low

- RHEL 7 STIG Version 3, version 14
- RHEL 8 STIG Version 1 Version 13
- Ubuntu 18.04 STIG version 2, version 13
- Ubuntu 20.04 STIG version 1 version 11

STIG-Build-Linux-Medium version 2024.1.x

- RHEL 7 STIG Version 3, version 14
- RHEL 8 STIG Version 1 Version 13
- Ubuntu 18.04 STIG version 2, version 13
- Ubuntu 20.04 STIG version 1 version 11

STIG-Build-Linux-High version 2024.1.x

- RHEL 7 STIG Version 3, version 14

- RHEL 8 STIG Version 1 Version 13
- Ubuntu 18.04 STIG version 2, version 13
- Ubuntu 20.04 STIG version 1 version 11

Modifications apportées au quatrième trimestre 2023 - 07/12/2023 :

Versions STIG mises à jour et STIGS appliquées pour la version du quatrième trimestre 2023 comme suit :

Version 2023.4.x de STIG-Build-Linux-Low

- RHEL 7 STIG Version 3 Version 13
- RHEL 8 STIG Version 1 Version 12
- Ubuntu 18.04 STIG version 2 version 12
- Ubuntu 20.04 STIG version 1 version 10

STIG-Build-Linux-Medium version 2023.4.x

- RHEL 7 STIG Version 3 Version 13
- RHEL 8 STIG Version 1 Version 12
- Ubuntu 18.04 STIG version 2 version 12
- Ubuntu 20.04 STIG version 1 version 10

STIG-Build-Linux-High version 2023.4.x

- RHEL 7 STIG Version 3 Version 13
- RHEL 8 STIG Version 1 Version 12
- Ubuntu 18.04 STIG version 2 version 12
- Ubuntu 20.04 STIG version 1 version 10

Modifications apportées au troisième trimestre 2023 - 10/04/2023 :

Versions STIG mises à jour et STIGS appliquées pour la version du troisième trimestre 2023 comme suit :

Version 2023.3.x de STIG-Build-Linux-Low

- RHEL 7 STIG version 3 version 12
- RHEL 8 STIG Version 1 Version 11
- Ubuntu 18.04 STIG version 2 version 11
- Ubuntu 20.04 STIG version 1 version 9

STIG-Build-Linux-Medium version 2023.3.x

- RHEL 7 STIG version 3 version 12
- RHEL 8 STIG Version 1 Version 11
- Ubuntu 18.04 STIG version 2 version 11
- Ubuntu 20.04 STIG version 1 version 9

STIG-Build-Linux-High version 2023.3.x

- RHEL 7 STIG version 3 version 12
- RHEL 8 STIG Version 1 Version 11
- Ubuntu 18.04 STIG version 2 version 11
- Ubuntu 20.04 STIG version 1 version 9

Changements du deuxième trimestre 2023 - 05/03/2023 :

Versions STIG mises à jour et STIGS appliquées pour la version du deuxième trimestre 2023 comme suit :

Version 2023.2.x de STIG-Build-Linux-Low

- RHEL 7 STIG version 3 version 11
- RHEL 8 STIG Version 1 Version 10
- Ubuntu 18.04 STIG version 2 version 11
- Ubuntu 20.04 STIG version 1 version 8

STIG-Build-Linux-Medium version 2023.2.x

- RHEL 7 STIG version 3 version 11
- RHEL 8 STIG Version 1 Version 10
- Ubuntu 18.04 STIG version 2 version 11
- Ubuntu 20.04 STIG version 1 version 8

STIG-Build-Linux-High version 2023.2.x

- RHEL 7 STIG version 3 version 11
- RHEL 8 STIG Version 1 Version 10
- Ubuntu 18.04 STIG version 2 version 11
- Ubuntu 20.04 STIG version 1 version 8

Changements du premier trimestre 2023 - 27/03/2023 :

Versions STIG mises à jour et STIGS appliquées pour la version du premier trimestre 2023 comme suit :

Version 2023.1.x de STIG-Build-Linux-Low

- RHEL 7 STIG version 3 version 10
- RHEL 8 STIG Version 1 Version 9
- Ubuntu 18.04 STIG version 2 version 10
- Ubuntu 20.04 STIG version 1 version 7

STIG-Build-Linux-Medium version 2023.1.x

- RHEL 7 STIG version 3 version 10
- RHEL 8 STIG Version 1 Version 9
- Ubuntu 18.04 STIG version 2 version 10
- Ubuntu 20.04 STIG version 1 version 7

STIG-Build-Linux-High version 2023.1.x

- RHEL 7 STIG version 3 version 10
- RHEL 8 STIG Version 1 Version 9
- Ubuntu 18.04 STIG version 2 version 10
- Ubuntu 20.04 STIG version 1 version 7

Changements du quatrième trimestre 2022 - 01/02/2023 :

Versions STIG mises à jour et STIGS appliquées pour la version du quatrième trimestre 2022 comme suit :

STIG-Build-Linux-Low version 2022.4.x

- RHEL 7 STIG version 3, version 9
- RHEL 8 STIG Version 1 Version 8
- Ubuntu 18.04 STIG version 2, version 9
- Ubuntu 20.04 STIG version 1 version 6

STIG-Build-Linux-Medium version 2022.4.x

- RHEL 7 STIG version 3, version 9
- RHEL 8 STIG Version 1 Version 8
- Ubuntu 18.04 STIG version 2, version 9
- Ubuntu 20.04 STIG version 1 version 6

STIG-Build-Linux-High version 2022.4.x

- RHEL 7 STIG version 3, version 9
- RHEL 8 STIG Version 1 Version 8
- Ubuntu 18.04 STIG version 2, version 9
- Ubuntu 20.04 STIG version 1 version 6

Changements du troisième trimestre 2022 - 30/09/2022 (aucun changement) :

Aucune modification n'a été apportée au composant Linux STIGS pour la version du troisième trimestre 2022.

Changements du deuxième trimestre 2022 - 08/02/2022 :

Nous avons introduit le support d'Ubuntu, mis à jour les versions de STIG et appliqué les STIGS pour la version du deuxième trimestre 2022 comme suit :

STIG-Build-Linux-Low version 2022.2.x

- RHEL 7 STIG version 3, version 7
- RHEL 8 STIG Version 1 Version 6
- Ubuntu 18.04 STIG version 2, version 6 (nouveau)
- Ubuntu 20.04 STIG version 1, version 4 (nouveau)

STIG-Build-Linux-Medium version 2022.2.x

- RHEL 7 STIG Version 3 Version 7
- RHEL 8 STIG Version 1 Version 6
- Ubuntu 18.04 STIG version 2, version 6 (nouveau)
- Ubuntu 20.04 STIG version 1, version 4 (nouveau)

STIG-Build-Linux-High version 2022.2.x

- RHEL 7 STIG Version 3 Version 7
- RHEL 8 STIG Version 1 Version 6
- Ubuntu 18.04 STIG version 2, version 6 (nouveau)
- Ubuntu 20.04 STIG version 1, version 4 (nouveau)

Changements du premier trimestre 2022 - 26/04/2022 :

Refactorisé pour inclure un meilleur support pour les conteneurs. Combinaison du script AL2 précédent avec RHEL 7. Versions STIG mises à jour et STIGS appliquées pour la version du premier trimestre 2022 comme suit :

Version 3.6.x de STIG-Build-Linux-Low

- RHEL 7 STIG Version 3 Version 6
- RHEL 8 STIG Version 1 Version 5

STIG-Build-Linux-Medium version 3.6.x

- RHEL 7 STIG Version 3 Version 6
- RHEL 8 STIG Version 1 Version 5

STIG-Build-Linux-High version 3.6.x

- RHEL 7 STIG Version 3 Version 6
- RHEL 8 STIG Version 1 Version 5

Changements du quatrième trimestre 2021 - 20/12/2021 :

Versions STIG mises à jour et STIGS appliquées pour la version du quatrième trimestre 2021 comme suit :

STIG-Build-Linux-Low version 3.5.x

- RHEL 7 STIG version 3 version 5
- RHEL 8 STIG Version 1 Version 4

STIG-Build-Linux-Medium version 3.5.x

- RHEL 7 STIG version 3 version 5
- RHEL 8 STIG Version 1 Version 4

STIG-Build-Linux-High version 3.5.x

- RHEL 7 STIG version 3 version 5
- RHEL 8 STIG Version 1 Version 4

Changements du troisième trimestre 2021 au 30/09/2021 :

Versions STIG mises à jour et STIGS appliqués pour la version du troisième trimestre 2021 comme suit :

Version 3.4.x de STIG-Build-Linux-Low

- RHEL 7 STIG Version 3, version 4
- RHEL 8 STIG Version 1, version 3

STIG-Build-Linux-Medium version 3.4.x

- RHEL 7 STIG Version 3, version 4
- RHEL 8 STIG Version 1, version 3

STIG-Build-Linux-High version 3.4.x

- RHEL 7 STIG Version 3, version 4
- RHEL 8 STIG Version 1, version 3

Composant de validation de conformité SCAP

Le protocole SCAP (Security Content Automation Protocol) est un ensemble de normes que les professionnels de l'informatique peuvent utiliser pour identifier les vulnérabilités de sécurité des applications à des fins de conformité. Le SCAP Compliance Checker (SCC) est un outil d'analyse validé par le SCAP, publié par le Naval Information Warfare Center (NIWC) Atlantic. Pour plus d'informations, voir Vérificateur de conformité au protocole SCAP (Security Content Automation Protocol) (SCC) sur le site Web de NIWC Atlantic.

Les `scap-compliance-checker-linux` composants AWSTOE `scap-compliance-checker-windows` et téléchargent et installent le scanner SCC sur les instances de construction et de test du pipeline. Lorsque le scanner s'exécute, il effectue des analyses de configuration authentifiées à l'aide des benchmarks DISA SCAP et fournit un rapport contenant les informations suivantes. AWSTOE écrit également les informations dans les journaux de vos applications.

- Paramètres STIG appliqués à l'instance.
- Un score de conformité global pour l'instance.

Nous vous recommandons d'exécuter la validation SCAP comme dernière étape de votre processus de création, afin de garantir l'exactitude des résultats de validation de conformité.

> **Note**
>
> Vous pouvez consulter les rapports à l'aide de l'un des outils de visualisation STIG. Ces outils sont disponibles en ligne via le DoD Cyber Exchange.

Les sections suivantes décrivent les benchmarks inclus dans les composants de validation SCAP.

scap-compliance-checker-linux version 2021.04.0

Le `scap-compliance-checker-linux` composant s'exécute sur les instances de build et de test du pipeline Image Builder. AWSTOE enregistre à la fois le rapport et le score produits par l'application SCC.

Le composant exécute les étapes de flux de travail suivantes :

1. Télécharge et installe l'application SCC.
2. Importe les repères de conformité.
3. Exécute la validation à l'aide de l'application SCC.
4. Enregistre le rapport de conformité et le score localement sur le poste de travail de l'instance de build.
5. Enregistre le score de conformité du rapport local dans les fichiers journaux de AWSTOE l'application.

> **Note**
>
> AWSTOE prend actuellement en charge la validation de conformité SCAP pour Windows Server 2012 R2, 2016 et 2019.

Le composant du vérificateur de conformité SCAP pour Windows inclut les tests de performance suivants :

Version SCC : 5.4.2

Points de référence du quatrième trimestre 2021 :

- U_MS_ _Framework_4-0_V2R1_Stig_Scap_1-2_Benchmark DotNet
- U_MS_IE11_V2R1_Stig_Scap_1-2_Benchmark
- U_MS_Windows_2012_et_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 version 2021.04.0

Le `scap-compliance-checker-linux` composant s'exécute sur les instances de build et de test du pipeline Image Builder. AWSTOE enregistre à la fois le rapport et le score produits par l'application SCC.

Le composant exécute les étapes de flux de travail suivantes :

1. Télécharge et installe l'application SCC.
2. Importe les repères de conformité.
3. Exécute la validation à l'aide de l'application SCC.
4. Enregistre le rapport de conformité et le score localement, à l'emplacement suivant sur l'instance de build :`/opt/scc/SCCResults`.
5. Enregistre le score de conformité du rapport local dans les fichiers journaux de AWSTOE l'application.

> **Note**
>
> AWSTOE prend actuellement en charge la validation de conformité SCAP pour RHEL 7/8 et Ubuntu 18. L'application SCC prend actuellement en charge l'architecture x86 pour la validation.

Le composant du vérificateur de conformité SCAP pour Linux inclut les tests de performance suivants :

Version SCC : 5.4.2

Points de référence du quatrième trimestre 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_ _Framework_4-0_V2R1_Stig_Scap_1-2_Benchmark DotNet
- U_MS_IE11_V2R1_Stig_Scap_1-2_Benchmark
- U_MS_Windows_2012_et_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

Historique des versions de SCAP

Le tableau suivant décrit les modifications importantes apportées à l'environnement et aux paramètres SCAP décrits dans ce document.

Modification	Description	Date
Composants SCAP ajoutés	Les composants SCAP suivants ont été introduits : • scap-compliance-checker-linux Version créée 2021.04.0 (Version SCC : 5.4.2) • scap-compliance-checker-linux Version créée 2021.04.0 (Version SCC : 5.4.2)	20 décembre 2021

AWSTOE référence de commande

AWSTOE est une application de gestion des composants qui s'exécute dans le AWS CLI.

> **Note**
>
> Certains modules AWSTOE d'action nécessitent des autorisations élevées pour s'exécuter sur un serveur Linux. Pour utiliser des autorisations élevées, préfixez la syntaxe de la commande par sudo ou sudo su exécutez-la une fois lorsque vous vous connectez avant d'exécuter les commandes liées ci-dessous. Pour plus d'informations sur les modules AWSTOE d'action, consultezModules d'action pris en charge par le gestionnaire de AWSTOE composants.

run

Utilisez la run commande pour exécuter les scripts de document YAML pour un ou plusieurs documents de composants.

valider

Exécutez la validate commande pour valider la syntaxe du document YAML pour un ou plusieurs documents de composants.

commande awstoe run

Cette commande exécute les scripts de document du composant YAML dans l'ordre dans lequel ils sont inclus dans le fichier de configuration spécifié par le `--config` paramètre ou dans la liste des documents du composant spécifiée par le `--documents` paramètre.

> **Note**
>
> Vous devez spécifier exactement l'un des paramètres suivants, jamais les deux :
> --configuration
> --documents

Syntaxe

```
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]
```

Paramètres et options

Paramètres

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

Forme abrégée : -c *./config-example.json*

Le fichier de configuration (conditionnel). Ce paramètre contient l'emplacement du fichier JSON contenant les paramètres de configuration des composants exécutés par cette commande. Si vous spécifiez des paramètres de run commande dans un fichier de configuration, vous ne devez pas spécifier le `--documents` paramètre. Pour plus d'informations sur la configuration des entrées, consultezConfigurer l'entrée pour la commande d' AWSTOE exécution.

Les emplacements valides incluent :

- Un chemin de fichier local (*./config-example.json*)
- Un URI S3 (`s3://`*bucket/key*)

--cw-ignore-failures

Forme abrégée : N/A

Ignorez les échecs de journalisation dans les CloudWatch journaux.

--cw-log-group

Forme abrégée : N/A

Le `LogGroup` nom des CloudWatch journaux.

--cw-log-region

Forme abrégée : N/A

La AWS région qui s'applique aux CloudWatch journaux.

--cw-log-stream

Forme abrégée : N/A

Le `LogStream` nom des CloudWatch journaux, qui indique AWSTOE où diffuser le `console.log` fichier.

--document-s3-bucket-owner

Forme abrégée : N/A

L'ID de compte du propriétaire du compartiment pour les documents basés sur l'URI S3.

--des documents./doc-1.yaml, ./doc-n.yaml

Forme abrégée : -d, *./doc-1.yaml./doc-n*

Les documents du composant (conditionnels). Ce paramètre contient une liste d'emplacements de fichiers séparés par des virgules pour les documents du composant YAML à exécuter. Si vous spécifiez des documents YAML pour la run commande à l'aide du `--documents` paramètre, vous ne devez pas le `--config` spécifier.

Les emplacements valides incluent :

- chemins de fichiers locaux (*. /component-doc-example.yaml).*
- URI S3 (`s3://`*bucket/key*).
- *ARN de version de build du composant Image Builder (arn:aws:imagebuilder:us-west-2:123456789012:component/ /2021.12.02/1). my-example-component*

> **Note**
>
> Il n'y a aucun espace entre les éléments de la liste, uniquement des virgules.

--identifiant d'exécution

Forme abrégée : -i

Il s'agit de l'identifiant unique qui s'applique à l'exécution de la run commande en cours. Cet identifiant est inclus dans les noms des fichiers de sortie et des fichiers journaux, afin d'identifier

ces fichiers de manière unique et de les lier à l'exécution de la commande en cours. Si ce paramètre est omis, AWSTOE génère un GUID.

--log-directory

Forme abrégée : -l

Le répertoire de destination où sont AWSTOE stockés tous les fichiers journaux de l'exécution de cette commande. Par défaut, ce répertoire se trouve dans le répertoire parent suivant :`TOE_<DATETIME>_<EXECUTIONID>`. Si vous ne spécifiez pas le répertoire du journal, AWSTOE utilise le répertoire de travail actuel (`.`).

--log-s3-bucket-name

Forme abrégée : -b

Si les journaux des composants sont stockés dans Amazon S3 (recommandé), AWSTOE télécharge les journaux des applications des composants dans le compartiment S3 nommé dans ce paramètre.

--log-s3-bucket-owner

Forme abrégée : N/A

Si les journaux des composants sont stockés dans Amazon S3 (recommandé), il s'agit de l'ID de compte propriétaire du compartiment dans lequel AWSTOE les fichiers journaux sont écrits.

--log-s3-key-prefix

Forme abrégée : -k

Si les journaux des composants sont stockés dans Amazon S3 (recommandé), il s'agit du préfixe de clé d'objet S3 pour l'emplacement du journal dans le compartiment.

--paramètres nom1 = valeur1, nom2 = valeur2...

Forme abrégée : N/A

Les paramètres sont des variables mutables définies dans le document du composant, avec des paramètres que l'application appelante peut fournir lors de l'exécution.

--phases

Forme abrégée : -p

Liste séparée par des virgules qui spécifie les phases à exécuter à partir des documents du composant YAML. Si le document d'un composant inclut des phases supplémentaires, celles-ci ne seront pas exécutées.

--répertoire d'état

Forme abrégée : -s

Le chemin du fichier dans lequel les fichiers de suivi d'état sont stockés.

--Version

Forme abrégée : -v

Spécifie la version du composant de l'application.

Options

--help

Forme abrégée : -h

Affiche un manuel d'aide pour utiliser les options de l'application de gestion des composants.

--tracer

Forme abrégée : -t

Active la journalisation détaillée sur la console.

commande awstoe validate

Lorsque vous exécutez cette commande, elle valide la syntaxe du document YAML pour chacun des documents de composant spécifiés par le `--documents` paramètre.

Syntaxe

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

Paramètres et options

Paramètres

--document-s3-bucket-owner

Forme abrégée : N/A

ID de compte source des documents basés sur l'URI S3 fournis.

--des documents./doc-1.yaml, ./doc-n.yaml

Forme abrégée : -d, *./doc-1.yaml./doc-n*

Les documents relatifs aux composants (obligatoires). Ce paramètre contient une liste d'emplacements de fichiers séparés par des virgules pour les documents du composant YAML à exécuter. Les emplacements valides incluent :

- chemins de fichiers locaux (*. /component-doc-example.yaml*)
- URI S3 () `s3://`*bucket/key*
- *ARN de version de construction du composant Image Builder (arn:aws:imagebuilder:us-west-2:123456789012:component/ /2021.12.02/1) my-example-component*

> **Note**
>
> Il n'y a aucun espace entre les éléments de la liste, uniquement des virgules.

Options

--help

Forme abrégée : -h

Affiche un manuel d'aide pour utiliser les options de l'application de gestion des composants.

--tracer

Forme abrégée : -t

Active la journalisation détaillée sur la console.

Gérer les ressources EC2 Image Builder

Les ressources sont les éléments constitutifs des pipelines d'images, ainsi que les images produites par ces pipelines. Ce chapitre traite de la création, de la maintenance et du partage des ressources Image Builder, notamment des composants, des recettes et des images, ainsi que de la configuration de l'infrastructure et des paramètres de distribution.

> **Note**
>
> Pour vous aider à gérer vos ressources Image Builder, vous pouvez attribuer vos propres métadonnées à chaque ressource sous forme de balises. Vous utilisez des balises pour classer vos AWS ressources de différentes manières, par exemple par objectif, propriétaire ou environnement. Cela est utile lorsque vous avez de nombreuses ressources du même type. Vous pouvez identifier plus facilement une ressource spécifique en fonction des balises que vous lui avez attribuées.
>
> Pour plus d'informations sur le balisage de vos ressources à l'aide des commandes Image Builder dans le AWS CLI, consultez la Balisage des ressources section de ce guide.

Table des matières

Gérez les composants avec Image Builder

Image Builder utilise l'application de gestion des composants AWS Task Orchestrator and Executor (AWSTOE) pour orchestrer des flux de travail complexes. Les composants de création et de test qui fonctionnent avec l' AWSTOE application sont basés sur des documents YAML qui définissent les scripts permettant de personnaliser ou de tester votre image. Pour les images AMI, Image Builder installe les composants et l'application de gestion des AWSTOE composants sur ses instances de build et de test Amazon EC2. Pour les images de conteneur, les composants et l'application de gestion des AWSTOE composants sont installés à l'intérieur du conteneur en cours d'exécution.

Image Builder permet AWSTOE d'effectuer toutes les activités sur instance. Aucune configuration supplémentaire n'est requise pour interagir AWSTOE lorsque vous exécutez des commandes Image Builder ou utilisez la console Image Builder.

> **Note**
>
> Lorsqu'un composant géré par Amazon atteint la fin de sa durée de support, il n'est plus maintenu. Environ quatre semaines avant que cela ne se produise, tous les comptes utilisant le composant reçoivent une notification et une liste des recettes concernées dans leur compte de la part de leur compte AWS Health Dashboard. Pour en savoir plus AWS Health, consultez le guide de AWS Health l'utilisateur.

Étapes du flux de travail pour créer une nouvelle image

Le flux de travail Image Builder pour créer de nouvelles images comprend les deux étapes distinctes suivantes.

1. Étape de création (pré-instantané) — Au cours de la phase de génération, vous apportez des modifications à l'instance de génération Amazon EC2 qui exécute votre image de base, afin de créer la base de référence pour votre nouvelle image. Par exemple, votre recette peut inclure des composants qui installent une application ou modifient les paramètres du pare-feu du système d'exploitation.

   Les phases de composants suivantes s'exécutent pendant la phase de construction :

   - build
   - valider

Une fois cette étape terminée avec succès, Image Builder crée un instantané ou une image conteneur qu'il utilise pour la phase de test et au-delà.

2. Phase de test (post-capture instantanée) — Au cours de la phase de test, il existe certaines différences entre les images qui créent des AMI et les images de conteneur. Pour les flux de travail AMI, Image Builder lance une instance EC2 à partir de l'instantané qu'il a créé comme dernière étape de la phase de création. Des tests sont exécutés sur la nouvelle instance pour valider les paramètres et s'assurer que l'instance fonctionne comme prévu. Pour les flux de travail de conteneurs, les tests s'exécutent sur la même instance que celle utilisée pour la création.

   La phase de composant suivante s'exécute pour chaque composant inclus dans la recette pendant la phase de test :

   - test

   Cette phase de composant s'applique à la fois aux types de composants de construction et de test. Une fois cette étape terminée avec succès, Image Builder peut créer et distribuer votre image finale à partir de l'instantané ou de l'image du conteneur.

> **Note**
>
> Bien qu'il vous AWSTOE permette de définir de nombreuses phases dans un document de composant, Image Builder applique des règles strictes concernant les phases à exécuter et au cours desquelles il les exécute. Pour qu'un composant s'exécute pendant la phase de construction, le document du composant doit définir au moins l'une des phases suivantes : `build` ou`validate`. Pour qu'un composant soit exécuté pendant la phase de test, le document du composant doit définir la `test` phase, et aucune autre phase.
> Image Builder exécute les étapes de manière indépendante, le chaînage des références dans les documents des composants ne peut pas dépasser les limites des étapes. Vous ne pouvez pas enchaîner une valeur d'une phase exécutée dans la phase de génération à une phase exécutée dans la phase de test. Vous pouvez toutefois définir des paramètres d'entrée pour la cible prévue et transmettre des valeurs via la ligne de commande. Pour plus d'informations sur la définition des paramètres des composants dans vos recettes Image Builder, consultezGérez les paramètres des AWSTOE composants avec EC2 Image Builder.

Pour faciliter le dépannage de votre instance de build ou de test, AWSTOE créez un dossier journal contenant le document d'entrée et les fichiers journaux pour suivre ce qui se passe à chaque

exécution d'un composant. Si vous avez configuré un compartiment Amazon S3 dans la configuration de votre pipeline, les journaux y sont également écrits. Pour plus d'informations sur les documents YAML et les résultats du journal, consultezUtiliser les documents relatifs aux composants dans AWSTOE.

> **Tip**
>
> Lorsque vous devez suivre de nombreux composants, le balisage vous aide à identifier un composant ou une version spécifique en fonction des balises que vous lui avez attribuées. Pour plus d'informations sur le balisage de vos ressources à l'aide des commandes Image Builder dans le AWS CLI, consultez la Balisage des ressources section de ce guide.

Cette section explique comment répertorier, afficher, créer et importer des composants à l'aide de la console Image Builder ou des commandes du AWS CLI.

Table des matières

Création d'un document de composant YAML

Pour créer un composant, fournissez un document de composant d'application YAML. Cela représente les phases et les étapes dont vous avez besoin pour créer le composant.

Les exemples présentés dans cette section créent un composant de génération qui appelle le module `Update0S` d'action dans l'application de gestion des AWSTOE composants. Le module met à jour le système d'exploitation. Pour plus d'informations sur le module `Update0S` d'action, consultezMettre à jour le système d'exploitation. Pour plus d'informations sur les phases, les étapes et la syntaxe des documents relatifs aux composants d'application AWSTOE YAML, consultez la section Utiliser des documents dans AWSTOE.

> **Note**
>
> Image Builder détermine les types de composants dans le flux de travail du pipeline. Ce flux de travail correspond à la phase de construction et à la phase de test du processus de génération. Image Builder détermine le type de composant comme suit :
>
> - Construire — Il s'agit du type de composant par défaut. Tout ce qui n'est pas classé comme composant de test est un composant de construction. Ce type de composant s'exécute pendant la phase de construction. Si une `test` phase est définie pour ce composant de construction, cette phase s'exécute pendant la phase de test.
> - Test — Pour être considéré comme un composant de test, le document du composant ne doit inclure qu'une seule phase, nomméetest. Pour les tests liés à la configuration des composants de génération, nous vous recommandons de ne pas utiliser de composant de test autonome. Utilisez plutôt la `test` phase dans le composant de construction associé.
>
> Pour plus d'informations sur la manière dont Image Builder utilise les étapes et les phases pour gérer le flux de travail des composants dans son processus de création, consultezGérez les composants avec Image Builder.

Pour créer un document de composant d'application YAML pour un exemple d'application, suivez les étapes de l'onglet correspondant à votre système d'exploitation d'image.

Linux

Création d'un fichier de composant YAML

Utilisez un outil d'édition de fichiers pour créer un fichier nommé*`update-linux-os.yaml`*. Incluez le contenu suivant :

```
# 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**
>
> Utilisez un outil tel que ce validateur YAML en ligne ou une extension YAML lint dans votre environnement de code pour vérifier que votre code YAML est bien formé.

Windows

Création d'un fichier de composant YAML

Utilisez un outil d'édition de fichiers pour créer un fichier nommé*update-windows-os.yaml*. Incluez le contenu suivant :

```
# 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

Utilisez un outil tel que ce validateur YAML en ligne ou une extension YAML lint dans votre environnement de code pour vérifier que votre code YAML est bien formé.

Gérez les paramètres des AWSTOE composants avec EC2 Image Builder

Vous pouvez gérer les AWSTOE composants, notamment en créant et en définissant les paramètres des composants, directement à partir de la console EC2 Image Builder, à AWS CLI l'aide de commandes ou de l'un des SDK d'Image Builder. Dans cette section, nous aborderons la création et l'utilisation de paramètres dans votre composant, ainsi que la définition des paramètres des composants via la console et AWS CLI les commandes Image Builder.

Important

Les paramètres des composants sont des valeurs en texte brut et sont connectés AWS CloudTrail. Nous vous recommandons d'utiliser AWS Secrets Manager le AWS Systems Manager Parameter Store pour stocker vos secrets. Pour plus d'informations sur Secrets Manager, voir Qu'est-ce que Secrets Manager ? dans le guide de AWS Secrets Manager

l'utilisateur. Pour plus d'informations sur AWS Systems Manager Parameter Store, voir AWS Systems Manager Parameter Store dans le guide de AWS Systems Manager l'utilisateur.

Utiliser des paramètres dans le document de votre composant YAML

Pour créer un composant, fournissez un document de composant d'application YAML. Cela représente les phases et les étapes dont vous avez besoin pour créer le composant. La recette qui fait référence au composant peut définir les paramètres pour personnaliser les valeurs lors de l'exécution, avec des valeurs par défaut qui prennent effet si le paramètre n'est pas défini sur une valeur spécifique.

Création d'un document de composant avec des paramètres d'entrée

Cette section explique comment définir et utiliser les paramètres d'entrée dans votre document de composant YAML.

Pour créer un document de composant d'application YAML qui utilise des paramètres et exécute des commandes dans vos instances de build ou de test Image Builder, suivez les étapes correspondant à votre système d'exploitation d'image :

Linux

Création d'un document de composant YAML

Utilisez un outil d'édition de fichiers pour créer un fichier nommé*hello-world-test.yaml*. Incluez le contenu suivant :

```
# 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**
>
> Utilisez un outil tel que ce validateur YAML en ligne ou une extension YAML lint dans votre environnement de code pour vérifier que votre code YAML est bien formé.

Windows

Création d'un document de composant YAML

Utilisez un outil d'édition de fichiers pour créer un fichier nommé*hello-world-test.yaml*. Incluez le contenu suivant :

```
# 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

Utilisez un outil tel que ce validateur YAML en ligne ou une extension YAML lint dans votre environnement de code pour vérifier que votre code YAML est bien formé.

Pour plus d'informations sur les phases, les étapes et la syntaxe des documents relatifs aux composants d'application AWSTOE YAML, consultez la section Utiliser des documents dans AWSTOE. Pour plus d'informations sur les paramètres et leurs exigences, consultez la Paramètres section de la page Définir et référencer les variables de AWSTOE la page.

Création d'un composant à partir du document du composant YAML

Quelle que soit la méthode que vous utilisez pour créer un AWSTOE composant, le document du composant d'application YAML est toujours requis comme référence.

- Pour utiliser la console Image Builder afin de créer un composant directement à partir de votre document YAML, consultezCréation d'un composant à l'aide de la console Image Builder.
- Pour utiliser les commandes Image Builder dans le AWS CLI pour créer votre composant, consultezCréez des AWSTOE composants avec Image Builder à l'aide du AWS CLI. Remplacez le nom du document YAML dans ces exemples par le nom de votre document YAML Hello World ()*`hello-world-test.yaml`*.

Définir les paramètres des composants dans une recette Image Builder (console)

La définition des paramètres des composants fonctionne de la même manière pour les recettes d'images et les recettes de conteneurs. Lorsque vous créez une nouvelle recette ou une nouvelle version d'une recette, vous choisissez les composants à inclure dans les listes Composants de construction et Composants de test. Les listes de composants incluent les composants applicables au système d'exploitation de base que vous avez choisi pour votre image.

Une fois que vous avez sélectionné un composant, il s'affiche dans la section Composants sélectionnés, directement sous les listes de composants. Les options de configuration sont affichées pour chaque composant sélectionné. Si des paramètres d'entrée sont définis pour votre composant, ils sont affichés sous la forme d'une section extensible appelée Paramètres d'entrée.

Les paramètres suivants sont affichés pour chaque paramètre défini pour votre composant :

- Nom du paramètre (non modifiable) : nom du paramètre.
- Description (non modifiable) — Description du paramètre
- Type (non modifiable) : type de données pour la valeur du paramètre.
- Valeur — La valeur du paramètre. Si vous utilisez ce composant pour la première fois dans cette recette et qu'une valeur par défaut a été définie pour le paramètre d'entrée, la valeur par défaut

apparaît dans la zone Valeur avec du texte grisé. Si aucune autre valeur n'est saisie, Image Builder utilise la valeur par défaut.

Répertorier et afficher les détails des composants

Cette section explique comment trouver des informations et afficher les détails des composants AWS Task Orchestrator and Executor (AWSTOE) que vous utilisez dans vos recettes EC2 Image Builder.

Détails des composants

- Lister AWSTOE les composants
- Répertorier les versions de construction des composants (AWS CLI)
- Obtenir les détails du composant (AWS CLI)
- Obtenir les détails de la politique relative aux composants (AWS CLI)

Lister AWSTOE les composants

Vous pouvez utiliser l'une des méthodes suivantes pour répertorier et filtrer AWSTOE les composants.

AWS Management Console

Pour afficher la liste des composants du AWS Management Console, procédez comme suit :

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Sélectionnez Composants dans le volet de navigation. Par défaut, Image Builder affiche la liste des composants que possède votre compte.
3. Vous pouvez éventuellement filtrer en fonction de la propriété des composants. Pour voir les composants que vous ne possédez pas, mais auxquels vous avez accès, développez la liste déroulante des types de propriétaires et sélectionnez l'une des valeurs. La liste des types de propriétaires se trouve dans la barre de recherche, à côté de la zone de texte de recherche. Vous pouvez choisir parmi les valeurs suivantes :

   - Démarrage rapide (géré par Amazon) : composants accessibles au public créés et gérés par Amazon.
   - Possédé par moi — Composants que vous avez créés. Il s'agit de la sélection par défaut.

- Partagé avec moi : composants que d'autres personnes ont créés et partagés avec vous depuis leur compte.
- Géré par un tiers : composants appartenant à un tiers et auxquels vous vous êtes abonné AWS Marketplace.

AWS CLI

L'exemple suivant montre comment utiliser la list-components commande pour renvoyer une liste des AWSTOE composants que possède votre compte.

```
aws imagebuilder list-components
```

Vous pouvez éventuellement filtrer en fonction de la propriété des composants. L'attribut owner définit à qui appartiennent les composants que vous souhaitez répertorier. Par défaut, cette demande renvoie une liste des composants que possède votre compte. Pour filtrer les résultats par propriétaire du composant, spécifiez l'une des valeurs suivantes avec le `--owner` paramètre lorsque vous exécutez la list-components commande.

Valeurs du propriétaire du composant

- Auto-utilisateur
- Amazon
- ThirdParty
- Partagé

Les exemples suivants montrent la list-components commande avec le `--owner` paramètre permettant de filtrer les résultats.

```
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
```

Répertorier les versions de construction des composants (AWS CLI)

L'exemple suivant montre comment utiliser la list-component-build-versions commande pour répertorier les versions de construction de composants qui ont une version sémantique spécifique. Pour en savoir plus sur le versionnement sémantique des ressources Image Builder, consultez. Gestion des versions sémantique

```
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"
            }
        }
    ]
}
```

Obtenir les détails du composant (AWS CLI)

L'exemple suivant montre comment utiliser la get-component commande pour obtenir les détails d'un composant lorsque vous spécifiez son Amazon Resource Name (ARN).

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

Obtenir les détails de la politique relative aux composants (AWS CLI)

L'exemple suivant montre comment utiliser la get-component-policy commande pour obtenir les détails d'une politique de composant lorsque vous spécifiez l'ARN du composant.

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

Création d'un composant à l'aide de la console Image Builder

Pour créer un composant d' AWSTOE application à partir de la console Image Builder, procédez comme suit :

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Sélectionnez Composants dans le volet de navigation. Sélectionnez ensuite Créer un composant.
3. Sur la page Créer un composant, sous Détails du composant, entrez ce qui suit :
   a. Système d'exploitation d'image (OS). Spécifiez le système d'exploitation avec lequel le composant est compatible.
   b. Catégorie de composant. Dans le menu déroulant, sélectionnez le type de composant de construction ou de test que vous créez.
   c. Nom du composant. Entrez un nom pour le composant.
   d. Version du composant. Entrez le numéro de version du composant.
   e. Description. Fournissez une description facultative pour vous aider à identifier le composant.
   f. Modifier la description. Fournissez une description facultative pour vous aider à comprendre les modifications apportées à cette version du composant.
4. Dans la section Définition du document, l'option par défaut est Définir le contenu du document. Le document du composant définit les actions qu'Image Builder exécute sur les instances de génération et de test pour créer votre image.

   Dans le champ Contenu, entrez le contenu du document de votre composant YAML. Pour commencer par un exemple de Hello World pour Linux, choisissez l'option Utiliser un exemple. Pour en savoir plus sur la création d'un document de composant YAML ou sur le copier-coller de l'exemple UpdateOS à partir de cette page, consultez. Création d'un document de composant YAML
5. Après avoir saisi les détails du composant, sélectionnez Créer un composant.

> **Note**
>
> Pour voir votre nouveau composant lorsque vous créez ou mettez à jour une recette, appliquez le filtre Owned by me à la liste des composants de construction ou de test. Le filtre se trouve en haut de la liste des composants, à côté du champ de recherche.

6. Pour supprimer un composant, sur la page Composants, cochez la case à côté du composant que vous souhaitez supprimer. Dans le menu déroulant Actions, sélectionnez Supprimer le composant.

Pour créer une nouvelle version de composant, procédez comme suit :

1. En fonction de votre point de départ :
   - Sur la page de liste des composants : cochez la case à côté du nom du composant, puis sélectionnez Créer une nouvelle version dans le menu Actions.
   - Sur la page détaillée du composant : cliquez sur le bouton Créer une nouvelle version dans le coin supérieur droit de l'en-tête.
2. Les informations du composant sont déjà renseignées avec les valeurs actuelles lorsque la page Créer un composant s'affiche. Suivez les étapes de création d'un composant pour le mettre à jour. Cela garantit que vous entrez une version sémantique unique dans la version du composant. Pour en savoir plus sur le versionnement sémantique des ressources Image Builder, consultez. Gestion des versions sémantique

Créez un composant à l'aide du AWS CLI

Cette section décrit comment utiliser les commandes Image Builder pour créer des composants AWS Task Orchestrator and Executor (AWSTOE) à partir du AWS Command Line Interface. Pour créer un composant, fournissez un document de composant d'application YAML. Cela représente les phases et les étapes dont vous avez besoin pour créer le composant. Pour créer un nouveau document de composant YAML, consultezCréation d'un document de composant YAML.

Créez des AWSTOE composants avec Image Builder à l'aide du AWS CLI

Dans cette section, vous allez apprendre à configurer et à utiliser les commandes Image Builder AWS CLI pour créer un composant d' AWSTOE application, comme suit.

- Téléchargez le document de votre composant YAML dans un compartiment S3 auquel vous pouvez faire référence depuis la ligne de commande.
- Créez le composant AWSTOE d'application à l'aide de la create-component commande.
- Répertoriez les versions des composants à l'aide de la list-components commande et d'un filtre de nom pour voir quelles versions existent déjà. Vous pouvez utiliser le résultat pour déterminer quelle sera la prochaine version pour les mises à jour.

Pour créer un composant d' AWSTOE application à partir d'un document YAML d'entrée, suivez les étapes correspondant à la plate-forme de votre système d'exploitation d'image.

Linux

Stockez le document relatif aux composants de votre application dans Amazon S3

Vous pouvez utiliser un compartiment S3 comme référentiel pour le document source des composants de votre AWSTOE application. Pour enregistrer le document de votre composant, procédez comme suit :

- Téléchargez le document sur Amazon S3

  Si la taille de votre document est inférieure à 64 Ko, vous pouvez ignorer cette étape. Les documents d'une taille supérieure ou égale à 64 Ko doivent être stockés dans Amazon S3.

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

Création d'un composant à partir du document YAML

Pour rationaliser la create-component commande que vous utilisez dans le AWS CLI, créez un fichier JSON contenant tous les paramètres de composant que vous souhaitez transmettre à la commande. Indiquez l'emplacement du *`update-linux-os.yaml`* document que vous avez créé lors des étapes précédentes. La paire `uri` clé-valeur contient la référence du fichier.

> **Note**
>
> La convention de dénomination des valeurs de données dans le fichier JSON suit le modèle spécifié pour les paramètres de demande d'action de l'API Image Builder. Pour consulter les paramètres de demande de commande d'API, consultez la CreateComponentcommande dans le manuel de référence de l'API EC2 Image Builder.

Pour fournir les valeurs de données sous forme de paramètres de ligne de commande, reportez-vous aux noms de paramètres spécifiés dans la référence des AWS CLI commandes.

1. Créer un fichier JSON d'entrée CLI

   Utilisez un outil d'édition de fichiers pour créer un fichier nommé*create-update-linux-os-component.json*. Incluez le contenu suivant :

```
{
 "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

   - Vous devez inclure l'option `file://` au début du chemin du fichier JSON.
   - Le chemin d'accès du fichier JSON doit suivre la convention appropriée pour le système d'exploitation de base sur lequel vous exécutez la commande. En effet, Windows utilise la barre oblique inverse (\) pour faire référence au chemin du répertoire, et Linux utilise la barre oblique directe (/).

2. Création du composant

   Utilisez la commande suivante pour créer le composant, en faisant référence au nom du fichier JSON que vous avez créé à l'étape précédente :

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

> **Note**
>
> - Vous devez inclure l'option `file://` au début du chemin du fichier JSON.
> - Le chemin d'accès du fichier JSON doit suivre la convention appropriée pour le système d'exploitation de base sur lequel vous exécutez la commande. En effet, Windows utilise la barre oblique inverse (\) pour faire référence au chemin du répertoire, et Linux utilise la barre oblique directe (/).

Windows

Stockez le document relatif aux composants de votre application dans Amazon S3

Vous pouvez utiliser un compartiment S3 comme référentiel pour le document source des composants de votre AWSTOE application. Pour enregistrer le document de votre composant, procédez comme suit :

- Téléchargez le document sur Amazon S3

  Si la taille de votre document est inférieure à 64 Ko, vous pouvez ignorer cette étape. Les documents d'une taille supérieure ou égale à 64 Ko doivent être stockés dans Amazon S3.

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

Création d'un composant à partir du document YAML

Pour rationaliser la create-component commande que vous utilisez dans le AWS CLI, créez un fichier JSON contenant tous les paramètres de composant que vous souhaitez transmettre à la commande. Indiquez l'emplacement du *`update-windows-os.yaml`* document que vous avez créé lors des étapes précédentes. La paire `uri` clé-valeur contient la référence du fichier.

> **Note**
>
> La convention de dénomination des valeurs de données dans le fichier JSON suit le modèle spécifié pour les paramètres de demande d'action de l'API Image Builder.

Pour consulter les paramètres de demande de commande d'API, consultez la CreateComponentcommande dans le manuel de référence de l'API EC2 Image Builder. Pour fournir les valeurs de données sous forme de paramètres de ligne de commande, reportez-vous aux noms de paramètres spécifiés dans la référence des AWS CLI commandes.

1. Créer un fichier JSON d'entrée CLI

   Utilisez un outil d'édition de fichiers pour créer un fichier nommé*create-update-windows-os-component.json*. Incluez le contenu suivant :

```
{
 "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

   - Vous devez inclure l'option `file://` au début du chemin du fichier JSON.
   - Le chemin d'accès du fichier JSON doit suivre la convention appropriée pour le système d'exploitation de base sur lequel vous exécutez la commande. En effet, Windows utilise la barre oblique inverse (\) pour faire référence au chemin du répertoire, et Linux utilise la barre oblique directe (/).

2. Création du composant

   Utilisez la commande suivante pour créer le composant, en faisant référence au nom du fichier JSON que vous avez créé à l'étape précédente :

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

> **Note**
>
> - Vous devez inclure l'option `file://` au début du chemin du fichier JSON.
> - Le chemin d'accès du fichier JSON doit suivre la convention appropriée pour le système d'exploitation de base sur lequel vous exécutez la commande. En effet, Windows utilise la barre oblique inverse (\) pour faire référence au chemin du répertoire, et Linux utilise la barre oblique directe (/).

AWSTOE gestion des versions des composants pour les mises à jour ()AWS CLI

AWSTOE les noms et versions des composants sont intégrés dans le nom de ressource Amazon (ARN) du composant, après le préfixe du composant. Chaque nouvelle version d'un composant possède son propre ARN unique. Les étapes de création d'une nouvelle version sont exactement les mêmes que celles de création d'un nouveau composant, à condition que la version sémantique soit unique pour le nom de ce composant. Pour en savoir plus sur le versionnement sémantique des ressources Image Builder, consultez. Gestion des versions sémantique

Pour être sûr d'attribuer la version logique suivante, obtenez d'abord une liste des versions existantes du composant que vous souhaitez modifier. Utilisez la list-components commande avec le AWS CLI, et filtrez sur le nom.

Dans cet exemple, vous filtrez sur le nom du composant que vous avez créé dans les exemples Linux précédents. Pour répertorier le composant que vous avez créé, utilisez la valeur du `name` paramètre du fichier JSON que vous avez utilisé dans la create-component commande.

```
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"
        }
    ]
}
```

Sur la base de vos résultats, vous pouvez déterminer quelle devrait être la prochaine version.

Importer un composant (AWS CLI)

Dans certains scénarios, il peut être plus facile de démarrer avec un script préexistant. Pour ce scénario, vous pouvez utiliser l'exemple suivant.

Cet exemple suppose que vous disposez d'un fichier appelé *import-component.json* (comme indiqué). Notez que le fichier fait directement référence à un PowerShell script appelé `AdminConfig.ps1` qui a déjà été téléchargé vers*my-s3-bucket*. Actuellement, `SHELL` est pris en charge pour le composant`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"
}
```

Pour importer le composant, exécutez la commande suivante.

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

Nettoyage des ressources

Pour éviter des frais imprévus, veillez à nettoyer les ressources et les pipelines que vous avez créés à partir des exemples de ce guide. Pour plus d'informations sur la suppression de ressources dans Image Builder, consultezSupprimer les ressources EC2 Image Builder.

Gérez les recettes

Une recette EC2 Image Builder définit l'image de base à utiliser comme point de départ pour créer une nouvelle image, ainsi que l'ensemble des composants que vous ajoutez pour personnaliser votre image et vérifier que tout fonctionne comme prévu. Image Builder propose des choix de version automatiques pour chaque composant. Le nombre de composants que vous pouvez appliquer à une recette est limité à 20 composants au total. Cela inclut à la fois les composants de construction et de test.

Une fois que vous avez créé une recette, vous ne pouvez ni la modifier ni la remplacer. Pour mettre à jour les composants après avoir créé une recette, vous devez créer une nouvelle recette ou une nouvelle version de recette. Vous pouvez toujours appliquer des balises à vos recettes existantes. Pour plus d'informations sur le balisage de vos ressources à l'aide des commandes Image Builder dans le AWS CLI, consultez la Balisage des ressources section de ce guide.

> **Tip**
>
> Vous pouvez utiliser des composants gérés par Amazon dans vos recettes ou développer vos propres composants personnalisés avec l'application AWS Task Orchestrator and Executor (AWSTOE). Consultez Commencez avec AWSTOE pour démarrer.

Cette section explique comment répertorier, afficher et créer des recettes.

Table des matières

- Création d'une nouvelle version d'une recette d'image
- Création d'une nouvelle version d'une recette de conteneur
- Nettoyage des ressources

Répertorier et afficher les détails de la recette en image

Cette section décrit les différentes manières de trouver des informations et d'afficher les détails de vos recettes d'images EC2 Image Builder.

Détails de la recette en image

- Répertorier les recettes d'images (console)
- Répertorier les recettes d'images (AWS CLI)
- Afficher les détails de la recette en image (console)
- Obtenir les détails de la recette en image (AWS CLI)
- Afficher les détails de la politique relative aux recettes par image (AWS CLI)

Répertorier les recettes d'images (console)

Pour consulter la liste des recettes d'images créées sous votre compte dans la console Image Builder, procédez comme suit :

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Choisissez Image recipes dans le volet de navigation. Cela affiche la liste des recettes d'images créées sous votre compte.
3. Pour afficher les détails ou créer une nouvelle version de recette, cliquez sur le lien Nom de la recette. Cela ouvre la vue détaillée de la recette.

> **Note**
>
> Vous pouvez également cocher la case à côté du nom de la recette, puis choisir Afficher les détails.

Répertorier les recettes d'images (AWS CLI)

L'exemple suivant montre comment répertorier toutes vos recettes d'images à l'aide du AWS CLI.

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

Afficher les détails de la recette en image (console)

Pour afficher les détails d'une recette d'image spécifique à l'aide de la console Image Builder, sélectionnez la recette d'image à consulter, en suivant les étapes décrites dansRépertorier les recettes d'images (console).

Sur la page détaillée de la recette, vous pouvez :

- Supprimez la recette. Pour plus d'informations sur la suppression de ressources dans Image Builder, consultezSupprimer les ressources EC2 Image Builder.
- Créez une nouvelle version.
- Créez un pipeline à partir de la recette. Après avoir sélectionné Créer un pipeline à partir de cette recette, vous êtes redirigé vers l'assistant de pipeline. Pour plus d'informations sur la création d'un pipeline Image Builder à l'aide de l'assistant de pipeline, voir Créez un pipeline d'images à l'aide de l'assistant de console EC2 Image Builder

> **Note**
>
> Lorsque vous créez un pipeline à partir d'une recette existante, l'option permettant de créer une nouvelle recette n'est pas disponible.

Obtenir les détails de la recette en image (AWS CLI)

L'exemple suivant montre comment utiliser une commande imagebuilder CLI pour obtenir les détails d'une recette d'image en spécifiant son Amazon Resource Name (ARN).

```
aws imagebuilder get-image-recipe --image-recipe-arn arn:aws:imagebuilder:us-
west-2:123456789012:image-recipe/my-example-recipe/2020.12.03
```

Afficher les détails de la politique relative aux recettes par image (AWS CLI)

L'exemple suivant montre comment utiliser une commande imagebuilder CLI pour obtenir les détails d'une politique de recette d'image en spécifiant son 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
```

Répertorier et afficher les détails des recettes du contenant

Cette section décrit les méthodes permettant de trouver des informations et d'afficher les détails de vos recettes de conteneurs EC2 Image Builder.

Détails de la recette du contenant

- Répertorier les recettes de conteneurs dans la console
- Répertoriez les recettes de contenants avec le AWS CLI
- Afficher les détails de la recette du conteneur dans la console
- Obtenez les détails de la recette du contenant avec le AWS CLI
- Obtenez les détails de la politique en matière de recettes en contenants grâce au AWS CLI

Répertorier les recettes de conteneurs dans la console

Pour consulter la liste des recettes de conteneurs créées sous votre compte dans la console Image Builder, procédez comme suit :

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Choisissez Container recipes dans le volet de navigation. Cela affiche la liste des recettes de conteneurs créées sous votre compte.
3. Pour afficher les détails ou créer une nouvelle version de recette, cliquez sur le lien Nom de la recette. Cela ouvre la vue détaillée de la recette.

> **Note**
>
> Vous pouvez également cocher la case à côté du nom de la recette, puis choisir Afficher les détails.

Répertoriez les recettes de contenants avec le AWS CLI

L'exemple suivant montre comment répertorier toutes vos recettes de contenants à l'aide du AWS CLI.

```
aws imagebuilder list-container-recipes
```

Afficher les détails de la recette du conteneur dans la console

Pour afficher les détails d'une recette de conteneur spécifique à l'aide de la console Image Builder, sélectionnez la recette de conteneur à consulter et suivez les étapes décrites dansRépertorier les recettes de conteneurs dans la console.

Sur la page détaillée de la recette, vous pouvez effectuer les opérations suivantes :

- Supprimez la recette. Pour plus d'informations sur la suppression de ressources dans Image Builder, consultezSupprimer les ressources EC2 Image Builder.
- Créez une nouvelle version.
- Créez un pipeline à partir de la recette. Après avoir sélectionné Créer un pipeline à partir de cette recette, vous êtes redirigé vers l'assistant de pipeline. Pour plus d'informations sur la création d'un pipeline Image Builder à l'aide de l'assistant de pipeline, voir Créez un pipeline d'images à l'aide de l'assistant de console EC2 Image Builder

> **Note**
>
> Lorsque vous créez un pipeline à partir d'une recette existante, l'option permettant de créer une nouvelle recette n'est pas disponible.

Obtenez les détails de la recette du contenant avec le AWS CLI

L'exemple suivant montre comment utiliser une commande imagebuilder CLI pour obtenir les détails d'une recette de conteneur en spécifiant son ARN.

```
aws imagebuilder get-container-recipe --container-recipe-arn arn:aws:imagebuilder:us-
west-2:123456789012:container-recipe/my-example-recipe/2020.12.03
```

Obtenez les détails de la politique en matière de recettes en contenants grâce au AWS CLI

L'exemple suivant montre comment utiliser une commande imagebuilder CLI pour obtenir les détails d'une politique de recette de conteneur en spécifiant son 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
```

Création d'une nouvelle version d'une recette d'image

Cette section explique comment créer une nouvelle version d'une recette d'image.

Table des matières

Création d'une nouvelle version de recette imagée (console)

Lorsque vous créez une nouvelle version de recette, cela revient pratiquement à créer une nouvelle recette. La différence est que certains détails sont présélectionnés pour correspondre à la recette de base, dans la plupart des cas. La liste suivante décrit les différences entre la création d'une nouvelle recette et la création d'une nouvelle version d'une recette existante.

Détails de la recette de base dans la nouvelle version

- Nom — Non modifiable.
- Version — Obligatoire. Ce détail de base n'est pas prérempli avec la version actuelle ni avec aucun type de séquence. Entrez le numéro de version que vous souhaitez créer dans le format <major>. <minor>. <patch>. Si la version existe déjà, vous rencontrez une erreur.
- L'option Sélectionner une image : présélectionnée, mais vous pouvez la modifier. Si vous modifiez votre choix de source pour votre image de base, vous risquez de perdre d'autres détails qui dépendent de l'option d'origine que vous avez choisie.

  Pour voir les détails associés à votre sélection d'images de base, choisissez l'onglet correspondant à votre sélection.

  Managed image

  - Système d'exploitation d'image (OS) : non modifiable.

- Nom de l'image : présélectionné, en fonction de la combinaison des choix d'images de base que vous avez effectués pour la recette existante. Toutefois, si vous modifiez l'option Sélectionner une image, vous perdez le nom de l'image présélectionné.
- Options de gestion automatique des versions : ne correspondent pas à votre recette de base. Cette option d'image prend par défaut l'option Utiliser la version du système d'exploitation sélectionnée.

> **Important**
>
> Si vous utilisez le versionnement sémantique pour lancer les builds de pipeline, assurez-vous de modifier cette valeur en Utiliser la dernière version disponible du système d'exploitation. Pour en savoir plus sur le versionnement sémantique des ressources Image Builder, consultez. Gestion des versions sémantique

AWS Marketplace image

- Abonnements — Cet onglet doit être ouvert et l'image à partir de laquelle vous êtes abonné AWS Marketplace doit être présélectionnée pour correspondre à votre recette de base. Si vous modifiez l'image que votre recette utilise comme image de base, vous risquez de perdre d'autres détails qui dépendent de l'image d'origine que vous avez choisie.

Pour plus d'informations sur les AWS Marketplace produits, consultez la section Acheter des produits dans le Guide de AWS Marketplace l'acheteur.

Custom AMI

- ID AMI — Obligatoire. Toutefois, ce paramètre n'est pas prérempli avec votre saisie d'origine. Vous devez saisir l'ID AMI de votre image de base.

- Configuration de l'instance : les paramètres sont présélectionnés, mais vous pouvez les modifier.
  - Agent Systems Manager : vous pouvez activer ou désactiver cette case à cocher pour contrôler l'installation de l'agent Systems Manager sur la nouvelle image. La case est décochée par défaut pour inclure l'agent Systems Manager dans votre nouvelle image. Pour supprimer l'agent Systems Manager de l'image finale, cochez la case afin que l'agent ne soit pas inclus dans votre AMI.
  - Données utilisateur : vous pouvez utiliser cette zone pour fournir des commandes ou un script de commande à exécuter lorsque vous lancez votre instance de build. Toutefois, cette valeur remplace toutes les commandes qu'Image Builder aurait pu ajouter pour garantir l'installation de Systems Manager. Ces commandes incluent le script de nettoyage qu'Image Builder exécute normalement pour les images Linux avant de créer la nouvelle image.

> **Note**
>
> - Si vous entrez des données utilisateur, assurez-vous que l'agent Systems Manager est préinstallé sur votre image de base ou que vous incluez l'installation dans vos données utilisateur.
> - Pour les images Linux, assurez-vous que les étapes de nettoyage sont exécutées en incluant une commande permettant de créer un fichier vide nommé `perform_cleanup` dans votre script de données utilisateur. Image Builder détecte ce fichier et exécute le script de nettoyage avant de créer la nouvelle image. Pour plus d'informations et un exemple de script, consultezBonnes pratiques de sécurité pour EC2 Image Builder.

- Répertoire de travail : présélectionné, mais vous pouvez le modifier.
- Composants : les composants déjà inclus dans la recette sont affichés dans la section Composants sélectionnés à la fin de chacune des listes de composants (construction et test). Vous pouvez supprimer ou réorganiser les composants sélectionnés en fonction de vos besoins.

  Les composants de durcissement CIS ne respectent pas les règles de classement des composants standard des recettes Image Builder. Les composants de renforcement CIS sont toujours exécutés en dernier pour garantir que les tests de référence s'exécutent par rapport à votre image de sortie.

  > **Note**
  >
  > Les listes de composants de génération et de test affichent les composants disponibles en fonction du type de propriétaire du composant. Pour ajouter ou mettre à jour des composants pour votre recette, sélectionnez le type de propriétaire du composant que vous recherchez. Par exemple, si vous souhaitez ajouter un composant associé à une image de base à laquelle vous vous êtes abonné AWS Marketplace, sélectionnez-le dans la liste des types `Third party managed` de propriétaire située à côté de la barre de recherche.

  Vous pouvez configurer les paramètres suivants pour le composant sélectionné :

  - Options de gestion des versions : présélectionnées, mais vous pouvez les modifier. Nous vous recommandons de choisir l'option Utiliser la dernière version de composant disponible pour vous assurer que vos compilations d'images reprennent toujours la dernière version du composant. Si vous devez utiliser une version de composant spécifique dans votre recette, vous pouvez choisir

Spécifier la version du composant et saisir la version dans la zone Version du composant qui apparaît.

- Paramètres d'entrée : affiche les paramètres d'entrée acceptés par le composant. La valeur est préremplie avec la valeur de la version précédente de la recette. Si vous utilisez ce composant pour la première fois dans cette recette et qu'une valeur par défaut a été définie pour le paramètre d'entrée, la valeur par défaut apparaît dans la zone Valeur avec du texte grisé. Si aucune autre valeur n'est saisie, Image Builder utilise la valeur par défaut.

  Si un paramètre d'entrée est obligatoire, mais qu'aucune valeur par défaut n'est définie dans le composant, vous devez fournir une valeur. Image Builder ne créera pas la version de la recette s'il manque des paramètres requis et si aucune valeur par défaut n'est définie.

  > **Important**
  >
  > Les paramètres des composants sont des valeurs en texte brut et sont connectés AWS CloudTrail. Nous vous recommandons d'utiliser AWS Secrets Manager le AWS Systems Manager Parameter Store pour stocker vos secrets. Pour plus d'informations sur Secrets Manager, voir Qu'est-ce que Secrets Manager ? dans le guide de AWS Secrets Manager l'utilisateur. Pour plus d'informations sur AWS Systems Manager Parameter Store, voir AWS Systems Manager Parameter Store dans le guide de AWS Systems Manager l'utilisateur.

Pour développer les paramètres des options de version ou des paramètres d'entrée, vous pouvez cliquer sur la flèche à côté du nom du paramètre. Pour étendre tous les paramètres de tous les composants sélectionnés, vous pouvez activer ou désactiver le bouton Tout étendre.

- Stockage (volumes) — sont préremplis. Le nom du périphérique, le snapshot et les sélections d'IOPS du volume racine ne sont pas modifiables. Cependant, vous pouvez modifier tous les autres paramètres, tels que la taille. Vous pouvez également ajouter de nouveaux volumes et chiffrer des volumes nouveaux ou existants.

  Pour chiffrer les volumes des images créées par Image Builder sous votre compte dans la région source (où s'exécute la génération), vous devez utiliser le chiffrement des volumes de stockage dans la recette d'image. Le chiffrement exécuté pendant la phase de distribution de la compilation concerne uniquement les images distribuées à d'autres comptes ou régions.

> **Note**
>
> Si vous utilisez le chiffrement pour vos volumes, vous devez sélectionner la clé pour chaque volume séparément, même s'il s'agit de la même clé que celle utilisée pour le volume racine.

Pour créer une nouvelle version de recette imagée :

1. En haut de la page des détails de la recette, choisissez Créer une nouvelle version. Cela vous amène à la page Créer une recette d'image.
2. Pour créer la nouvelle version, apportez vos modifications, puis choisissez Créer une recette d'image.

Pour plus d'informations sur la création d'une recette d'image lorsque vous créez un pipeline d'images, consultez Étape 2 : Choisissez la recette la section Démarrage de ce guide.

Créez une recette imagée à l'aide du AWS CLI

Pour créer une recette d'image à l'aide de la `create-image-recipe` commande Image Builder dans le AWS CLI, procédez comme suit :

Prérequis

Avant d'exécuter les commandes Image Builder de cette section pour créer une recette d'image à partir de AWS CLI, vous devez créer les composants utilisés par la recette. L'exemple de recette illustré à l'étape suivante fait référence à des exemples de composants créés dans la Créez un composant à l'aide du AWS CLI section de ce guide.

Après avoir créé vos composants, ou si vous utilisez des composants existants, notez les ARN que vous souhaitez inclure dans la recette.

1. Créer un fichier JSON d'entrée CLI

   Vous pouvez fournir toutes les entrées de la create-image-recipe commande avec des paramètres de commande intégrés. Cependant, la commande qui en résulte peut être assez longue. Pour rationaliser la commande, vous pouvez à la place fournir un fichier JSON contenant tous les paramètres de la recette.

> **Note**
>
> La convention de dénomination des valeurs de données dans le fichier JSON suit le modèle spécifié pour les paramètres de demande d'action de l'API Image Builder. Pour consulter les paramètres de demande de commande d'API, consultez la CreateImageRecipecommande dans le manuel de référence de l'API EC2 Image Builder. Pour fournir les valeurs de données sous forme de paramètres de ligne de commande, reportez-vous aux noms de paramètres spécifiés dans la référence des AWS CLI commandes.

Voici un résumé des paramètres spécifiés dans ces exemples :

- name (chaîne, obligatoire) — Le nom de la recette d'image.
- description (chaîne) — Description de la recette d'image.
- parentImage (chaîne, obligatoire) — L'image que la recette d'image utilise comme base pour votre image personnalisée. La valeur peut être l'ARN de l'image de base ou un ID d'AMI.

  > **Note**
  >
  > L'exemple Linux utilise une AMI Image Builder, tandis que l'exemple Windows utilise un ARN.

- SemanticVersion <major>(chaîne, obligatoire) — Version sémantique de la recette d'image, exprimée dans le format suivant, avec des valeurs numériques à chaque position pour indiquer une version spécifique :. <minor>. <patch>. Par exemple, une valeur peut être`1.0.0`. Pour en savoir plus sur le versionnement sémantique des ressources Image Builder, consultez. Gestion des versions sémantique
- components (tableau, obligatoire) — Contient un tableau d'`ComponentConfiguration`objets. Au moins un composant de construction doit être spécifié :

  > **Note**
  >
  > Image Builder installe les composants dans l'ordre dans lequel vous les avez spécifiés dans la recette. Cependant, les composants de renforcement CIS s'exécutent toujours

en dernier pour garantir que les tests de référence s'exécutent par rapport à votre image de sortie.

- componentArn (string, obligatoire) — L'ARN du composant.

  > **Tip**
  >
  > Pour utiliser l'un des exemples afin de créer votre propre recette d'image, vous devez remplacer les exemples d'ARN par les ARN des composants que vous utilisez pour votre recette.

- paramètres (tableau d'objets) — Contient un tableau d'`ComponentParameter`objets. Si un paramètre d'entrée est obligatoire, mais qu'aucune valeur par défaut n'est définie dans le composant, vous devez fournir une valeur. Image Builder ne créera pas la version de la recette s'il manque des paramètres requis et si aucune valeur par défaut n'est définie.

  > **Important**
  >
  > Les paramètres des composants sont des valeurs en texte brut et sont connectés AWS CloudTrail. Nous vous recommandons d'utiliser AWS Secrets Manager le AWS Systems Manager Parameter Store pour stocker vos secrets. Pour plus d'informations sur Secrets Manager, voir Qu'est-ce que Secrets Manager ? dans le guide de AWS Secrets Manager l'utilisateur. Pour plus d'informations sur AWS Systems Manager Parameter Store, voir AWS Systems Manager Parameter Store dans le guide de AWS Systems Manager l'utilisateur.

  - name (chaîne, obligatoire) — Nom du paramètre du composant à définir.
  - valeur (tableau de chaînes, obligatoire) — Contient un tableau de chaînes pour définir la valeur du paramètre de composant nommé. Si une valeur par défaut est définie pour le composant et qu'aucune autre valeur n'est fournie, AWSTOE utilise la valeur par défaut.

- additionalInstanceConfiguration(objet) — Spécifiez des paramètres supplémentaires et lancez des scripts pour vos instances de build.
  - systemsManagerAgent(objet) — Contient les paramètres de l'agent Systems Manager sur votre instance de build.

- uninstallAfterBuild(Boolean) — Contrôle si l'agent Systems Manager est supprimé de votre image de build finale avant de créer la nouvelle AMI. Si cette option est définie sur`true`, l'agent est supprimé de l'image finale. Si l'option est définie sur`false`, l'agent reste actif afin qu'il soit inclus dans la nouvelle AMI. La valeur par défaut est `false`.

  > **Note**
  >
  > Si l'`uninstallAfterBuild`attribut n'est pas inclus dans le fichier JSON et que les conditions suivantes sont remplies, Image Builder supprime l'agent Systems Manager de l'image finale afin qu'il ne soit pas disponible dans l'AMI :
  >
  > - Le `userDataOverride` est vide ou a été omis dans le fichier JSON.
  > - Image Builder a automatiquement installé l'agent Systems Manager sur l'instance de build pour un système d'exploitation sur lequel l'agent n'était pas préinstallé sur l'image de base.

- userDataOverride(string) — Fournissez des commandes ou un script de commande à exécuter lorsque vous lancez votre instance de build.

  > **Note**
  >
  > Les données utilisateur sont toujours codées en base 64.
  > Par exemple, les commandes suivantes sont codées comme `IyEvYmluL2Jhc2gKbWtkaXIgLXAgL3Zhci9iYi8KdG91Y2ggL3Zhcg==` :
  >
  > ```
  > #!/bin/bash
  > mkdir -p /var/bb/
  > touch /var
  > ```
  >
  > L'exemple Linux utilise cette valeur codée.

Linux

L'image de base (`parentImage`propriété) de l'exemple suivant est une AMI. Lorsque vous utilisez une AMI, vous devez avoir accès à l'AMI, et l'AMI doit se trouver dans la région source (la même région où Image Builder exécute la commande). Enregistrez le fichier sous `create-image-recipe.json` et utilisez-le dans la create-image-recipe commande.

```
{
"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

L'exemple suivant fait référence à la dernière version de l'image de base complète en anglais de Windows Server 2016. Dans cet exemple, l'ARN fait référence à la dernière image du SKU en fonction des filtres de version sémantiques que vous avez spécifiés :`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**
>
> Pour en savoir plus sur le versionnement sémantique des ressources Image Builder, consultez. Gestion des versions sémantique

2. Créez la recette

   Utilisez la commande suivante pour créer la recette. Indiquez le nom du fichier JSON que vous avez créé à l'étape précédente dans le `--cli-input-json` paramètre :

```
aws imagebuilder create-image-recipe --cli-input-json file://create-image-recipe.json
```

> **Note**
>
> - Vous devez inclure l'option `file://` au début du chemin du fichier JSON.
> - Le chemin d'accès du fichier JSON doit suivre la convention appropriée pour le système d'exploitation de base sur lequel vous exécutez la commande. En effet, Windows utilise la barre oblique inverse (\) pour faire référence au chemin du répertoire, et Linux utilise la barre oblique directe (/).

Importer une machine virtuelle comme image de base dans la console

Dans cette section, nous nous concentrons sur la façon d'importer une machine virtuelle (VM) comme image de base pour votre recette d'image. Nous ne couvrons pas les autres étapes liées à la création d'une recette ou d'une version de recette ici. Pour connaître les étapes supplémentaires permettant de créer une nouvelle recette d'image à l'aide de l'assistant de création de pipeline de la console Image Builder, consultezCréation d'un pipeline d'images (AMI). Pour connaître les étapes supplémentaires permettant de créer une nouvelle recette sous forme d'image ou une nouvelle version de recette, voirCréation d'une nouvelle version d'une recette d'image.

Pour importer une machine virtuelle comme image de base pour votre recette d'image dans la console Image Builder, suivez ces étapes, ainsi que toute autre étape requise, pour créer votre recette ou version de recette.

1. Dans la section Sélectionner une image pour l'image de base, sélectionnez l'option Importer l'image de base.
2. Choisissez le système d'exploitation des images (OS) et la version du système d'exploitation comme vous le feriez normalement.

Configuration de l'importation de machines virtuelles

Lorsque vous exportez votre machine virtuelle depuis son environnement de virtualisation, ce processus crée un ensemble d'un ou plusieurs fichiers de conteneur de disque qui agissent comme des instantanés de l'environnement, des paramètres et des données de votre machine virtuelle. Vous pouvez utiliser ces fichiers pour importer votre machine virtuelle en tant qu'image de base pour votre recette d'image. Pour plus d'informations sur l'importation de machines virtuelles dans Image Builder, voir Importation et exportation d'images de machines virtuelles

Pour spécifier l'emplacement de votre source d'importation, procédez comme suit :

Source d'importation

Spécifiez la source du premier conteneur de disque ou d'instantané d'image de machine virtuelle à importer dans la section Conteneur de disque 1.

1. Source : il peut s'agir d'un compartiment S3 ou d'un instantané EBS.
2. Sélectionnez l'emplacement S3 du disque : entrez l'emplacement dans Amazon S3 où vos images de disque sont stockées. Pour rechercher l'emplacement, choisissez Parcourir S3.
3. Pour ajouter un conteneur de disques, choisissez Ajouter un conteneur de disques.

Rôle IAM

Pour associer un rôle IAM à la configuration d'importation de votre machine virtuelle, sélectionnez le rôle dans la liste déroulante des rôles IAM ou choisissez Créer un nouveau rôle pour en créer un nouveau. Si vous créez un nouveau rôle, la page de console IAM Roles s'ouvre dans un onglet distinct.

Réglages avancés — optionnel

Les paramètres suivants sont facultatifs. Avec ces paramètres, vous pouvez configurer le chiffrement, les licences, les balises, etc., pour l'image de base créée par l'importation.

Général

1. Spécifiez un nom unique pour l'image de base. Si vous n'entrez aucune valeur, l'image de base hérite du nom de la recette.
2. Spécifiez une version pour l'image de base. Utilisez le format suivant : *<major>.<minor>.<patch>*. Si vous n'entrez aucune valeur, l'image de base hérite de la version de la recette.
3. Vous pouvez également saisir une description pour l'image de base.

Architecture d'image de base

Pour spécifier l'architecture de la source d'importation de votre machine virtuelle, sélectionnez une valeur dans la liste Architecture.

Chiffrement

Si les images de disque de votre machine virtuelle sont chiffrées, vous devez fournir une clé à utiliser pour le processus d'importation. Pour spécifier un AWS KMS key pour l'importation, sélectionnez une valeur dans la liste Chiffrement (clé KMS). La liste contient les clés KMS auxquelles votre compte a accès dans la région actuelle.

Gestion des licences

Lorsque vous importez une machine virtuelle, le processus d'importation détecte automatiquement le système d'exploitation de la machine virtuelle et applique la licence appropriée à l'image de base. Selon la plate-forme de votre système d'exploitation, les types de licence sont les suivants :

- Licence incluse : une AWS licence adaptée à votre plateforme est appliquée à votre image de base.
- Apportez votre propre licence (BYOL) : conserve la licence de votre machine virtuelle, le cas échéant.

Pour associer des configurations de licence créées avec AWS License Manager à votre image de base, sélectionnez-les dans la liste des noms de configuration de licence. Pour plus d'informations sur License Manager, consultez Working with AWS License Manager

> **Note**
>
> - Les configurations de licence contiennent des règles de licence basées sur les termes de vos contrats d'entreprise.
> - Linux ne prend en charge que les licences BYOL.

Balises (image de base)

Les balises utilisent des paires clé-valeur pour attribuer du texte consultable à votre ressource Image Builder. Pour spécifier des balises pour l'image de base importée, entrez des paires clé-valeur dans les cases Clé et Valeur.

Pour ajouter une identification, choisissez Ajouter une identification. Pour supprimer une balise, choisissez Remove tag (Supprimer une balise).

Création d'une nouvelle version d'une recette de conteneur

Cette section explique comment créer une nouvelle version d'une recette de conteneur.

Table des matières

Créez une nouvelle version de recette de conteneur avec la console

La création d'une nouvelle version d'une recette contenant est pratiquement identique à la création d'une nouvelle recette. La différence est que certains détails sont présélectionnés pour correspondre à la recette de base, dans la plupart des cas. La liste suivante décrit les différences entre la création d'une nouvelle recette et la création d'une nouvelle version d'une recette existante.

Détails de la recette

- Nom : non modifiable.

- Version — Obligatoire. Ce détail n'est pas prérempli avec la version actuelle ni avec aucun type de séquence. Entrez le numéro de version que vous souhaitez créer au format major.minor.patch. Si la version existe déjà, vous rencontrez une erreur.

Image de base

- Sélectionnez l'option d'image : présélectionnée, mais modifiable. Si vous modifiez votre choix de source pour votre image de base, vous risquez de perdre d'autres détails qui dépendent de l'option d'origine que vous avez choisie.

  Pour voir les détails associés à votre sélection d'images de base, choisissez l'onglet correspondant à votre sélection.

  Managed images

  - Système d'exploitation d'image (OS) : non modifiable.
  - Nom de l'image : présélectionné, en fonction de la combinaison des choix d'images de base que vous avez effectués pour la recette existante. Toutefois, si vous modifiez l'option Sélectionner une image, vous perdez le nom de l'image présélectionné.
  - Options de gestion automatique des versions : ne correspondent pas à votre recette de base. Les options de gestion automatique des versions sont définies par défaut sur l'option Utiliser la version du système d'exploitation sélectionnée.

    > **Important**
    >
    > Si vous utilisez le versionnement sémantique pour lancer les builds de pipeline, assurez-vous de modifier cette valeur en Utiliser la dernière version disponible du système d'exploitation. Pour en savoir plus sur le versionnement sémantique des ressources Image Builder, consultez. Gestion des versions sémantique

  ECR image

  - Système d'exploitation d'image (OS) : présélectionné, mais modifiable.
  - Version du système d'exploitation : présélectionnée, mais modifiable.
  - ID d'image ECR : prérempli, mais modifiable.

  Docker Hub image

  - Système d'exploitation d'image (OS) : non modifiable.
  - Version du système d'exploitation : présélectionnée, mais modifiable.

- ID d'image Docker : prérempli, mais modifiable.

Configuration des instances

- ID AMI : prérempli, mais modifiable.
- Stockage (volumes)

  EBS volume 1 (racine AMI) — Prérempli. Vous ne pouvez pas modifier le nom du périphérique du volume racine, le snapshot ou les sélections d'IOPS. Cependant, vous pouvez modifier tous les autres paramètres, tels que la taille. Vous pouvez également ajouter de nouveaux volumes.

> **Note**
>
> Si vous avez spécifié une AMI de base qui a été partagée avec vous depuis un autre compte, les instantanés de tous les volumes secondaires spécifiés doivent également être partagés avec votre compte.

Répertoire de travail

- Chemin du répertoire de travail : pré-rempli, mais modifiable.

Composants

- Composants : les composants déjà inclus dans la recette sont affichés dans la section Composants sélectionnés à la fin de chacune des listes de composants (construction et test). Vous pouvez supprimer ou réorganiser les composants sélectionnés en fonction de vos besoins.

  Les composants de durcissement CIS ne respectent pas les règles de classement des composants standard des recettes Image Builder. Les composants de renforcement CIS sont toujours exécutés en dernier pour garantir que les tests de référence s'exécutent par rapport à votre image de sortie.

> **Note**
>
> Les listes de composants de génération et de test affichent les composants disponibles en fonction du type de propriétaire du composant. Pour ajouter ou mettre à jour des composants pour votre recette, sélectionnez le type de propriétaire du composant que vous recherchez. Par exemple, si vous souhaitez ajouter un composant associé à une image de

base à laquelle vous vous êtes abonné AWS Marketplace, sélectionnez-le dans la liste des types `Third party managed` de propriétaire située à côté de la barre de recherche.

Vous pouvez configurer les paramètres suivants pour le composant sélectionné :

- Options de gestion des versions : présélectionnées, mais vous pouvez les modifier. Nous vous recommandons de choisir l'option Utiliser la dernière version de composant disponible pour vous assurer que vos compilations d'images reprennent toujours la dernière version du composant. Si vous devez utiliser une version de composant spécifique dans votre recette, vous pouvez choisir Spécifier la version du composant et saisir la version dans la zone Version du composant qui apparaît.
- Paramètres d'entrée : affiche les paramètres d'entrée acceptés par le composant. La valeur est préremplie avec la valeur de la version précédente de la recette. Si vous utilisez ce composant pour la première fois dans cette recette et qu'une valeur par défaut a été définie pour le paramètre d'entrée, la valeur par défaut apparaît dans la zone Valeur avec du texte grisé. Si aucune autre valeur n'est saisie, Image Builder utilise la valeur par défaut.

  Si un paramètre d'entrée est obligatoire, mais qu'aucune valeur par défaut n'est définie dans le composant, vous devez fournir une valeur. Image Builder ne créera pas la version de la recette s'il manque des paramètres requis et si aucune valeur par défaut n'est définie.

  > **Important**
  >
  > Les paramètres des composants sont des valeurs en texte brut et sont connectés AWS CloudTrail. Nous vous recommandons d'utiliser AWS Secrets Manager le AWS Systems Manager Parameter Store pour stocker vos secrets. Pour plus d'informations sur Secrets Manager, voir Qu'est-ce que Secrets Manager ? dans le guide de AWS Secrets Manager l'utilisateur. Pour plus d'informations sur AWS Systems Manager Parameter Store, voir AWS Systems Manager Parameter Store dans le guide de AWS Systems Manager l'utilisateur.

Pour développer les paramètres des options de version ou des paramètres d'entrée, vous pouvez cliquer sur la flèche à côté du nom du paramètre. Pour étendre tous les paramètres de tous les composants sélectionnés, vous pouvez activer ou désactiver le bouton Tout étendre.

Modèle Dockerfile

- Modèle Dockerfile : pré-rempli, mais modifiable. Vous pouvez spécifier n'importe laquelle des variables contextuelles suivantes, qu'Image Builder remplace par des informations de build lors de l'exécution.

  ParentImage (obligatoire)

  Au moment de la création, cette variable devient l'image de base de votre recette.

  Exemple :

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

  environnements (obligatoire si des composants sont spécifiés)

  Cette variable sera convertie en un script qui exécute des composants.

  Exemple :

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

  composants (facultatif)

  Image Builder résout les scripts de création et de test des composants pour les composants inclus dans la recette du conteneur. Cette variable peut être placée n'importe où dans le Dockerfile, après la variable d'environnement.

  Exemple :

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

Référentiel cible

- Nom du référentiel cible : le référentiel Amazon ECR dans lequel votre image de sortie est stockée si aucun autre référentiel n'est spécifié dans la configuration de distribution de votre pipeline pour la région où le pipeline s'exécute (région 1).

Pour créer une nouvelle version de recette de conteneur :

1. En haut de la page des détails de la recette du conteneur, choisissez Créer une nouvelle version. Vous êtes redirigé vers la page Créer une recette pour les recettes par conteneur.
2. Pour créer la nouvelle version, apportez vos modifications, puis choisissez Créer une recette.

Pour plus d'informations sur la création d'une recette de conteneur lorsque vous créez un pipeline d'images, consultez la section Mise Étape 2 : Choisissez la recette en route de ce guide.

Créez une recette en pot avec le AWS CLI

Pour créer une recette de conteneur Image Builder à l'aide de la `imagebuilder create-container-recipe` commande AWS CLI, procédez comme suit :

Prérequis

Avant d'exécuter les commandes Image Builder de cette section pour créer une recette de conteneur avec le AWS CLI, vous devez créer les composants que la recette utilisera. L'exemple de recette de conteneur présenté à l'étape suivante fait référence à des exemples de composants créés dans la Créez un composant à l'aide du AWS CLI section de ce guide.

Après avoir créé vos composants, ou si vous utilisez des composants existants, notez les ARN que vous souhaitez inclure dans la recette.

1. Créer un fichier JSON d'entrée CLI

   Vous pouvez fournir toutes les entrées de la create-container-recipe commande avec des paramètres de commande intégrés. Cependant, la commande qui en résulte peut être assez longue. Pour rationaliser la commande, vous pouvez plutôt fournir un fichier JSON contenant tous les paramètres des recettes du conteneur

   > **Note**
   >
   > La convention de dénomination des valeurs de données dans le fichier JSON suit le modèle spécifié pour les paramètres de demande d'action de l'API Image Builder. Pour consulter les paramètres de demande de commande d'API, consultez la CreateContainerRecipecommande dans le manuel de référence de l'API EC2 Image Builder.

Pour fournir les valeurs de données sous forme de paramètres de ligne de commande, reportez-vous aux noms de paramètres spécifiés dans la référence des AWS CLI commandes.

Voici un résumé des paramètres de cet exemple :

- composants (tableau d'objets, obligatoire) — Contient un tableau d'`ComponentConfiguration`objets. Au moins un composant de construction doit être spécifié :

  Note

  Image Builder installe les composants dans l'ordre dans lequel vous les avez spécifiés dans la recette. Cependant, les composants de renforcement CIS s'exécutent toujours en dernier pour garantir que les tests de référence s'exécutent par rapport à votre image de sortie.

  - componentArn (string, obligatoire) — L'ARN du composant.

    Tip

    Pour utiliser cet exemple afin de créer votre propre recette de conteneur, remplacez les ARN d'exemple par les ARN des composants que vous utilisez pour votre recette,. Il s'agit Région AWS notamment du nom et du numéro de version de chacun.

  - paramètres (tableau d'objets) — Contient un tableau d'`ComponentParameter`objets. Si un paramètre d'entrée est obligatoire, mais qu'aucune valeur par défaut n'est définie dans le composant, vous devez fournir une valeur. Image Builder ne créera pas la version de la recette s'il manque des paramètres requis et si aucune valeur par défaut n'est définie.

    Important

    Les paramètres des composants sont des valeurs en texte brut et sont connectés AWS CloudTrail. Nous vous recommandons d'utiliser AWS Secrets Manager le AWS Systems Manager Parameter Store pour stocker vos secrets. Pour plus

d'informations sur Secrets Manager, voir Qu'est-ce que Secrets Manager ? dans le guide de AWS Secrets Manager l'utilisateur. Pour plus d'informations sur AWS Systems Manager Parameter Store, voir AWS Systems Manager Parameter Store dans le guide de AWS Systems Manager l'utilisateur.

  - name (chaîne, obligatoire) — Nom du paramètre du composant à définir.
  - valeur (tableau de chaînes, obligatoire) — Contient un tableau de chaînes pour définir la valeur du paramètre de composant nommé. Si une valeur par défaut est définie pour le composant et qu'aucune autre valeur n'est fournie, AWSTOE utilise la valeur par défaut.
- containerType (chaîne, obligatoire) — Type de conteneur à créer. Les valeurs valides incluentDOCKER.
- dockerfileTemplateData(string) — Le modèle Dockerfile utilisé pour créer votre image, exprimé sous la forme d'un blob de données intégré.
- name (chaîne, obligatoire) — Le nom de la recette du conteneur.
- description (chaîne) — Description de la recette du conteneur.
- parentImage (chaîne, obligatoire) — Image que la recette du conteneur utilise comme base pour votre image personnalisée. La valeur peut être l'ARN de l'image de base ou un ID d'AMI.
- PlatforMoverride (chaîne) — Spécifie la plate-forme du système d'exploitation lorsque vous utilisez une image de base personnalisée.
- SemanticVersion <major>(chaîne, obligatoire) — La version sémantique de la recette de conteneur spécifiée au format suivant, avec des valeurs numériques à chaque position pour indiquer une version spécifique :. <minor>. <patch>. Un exemple serait `1.0.0`. Pour en savoir plus sur le versionnement sémantique des ressources Image Builder, consultez. Gestion des versions sémantique
- tags (string map) — Tags attachés à la recette du conteneur.
- InstanceConfiguration (object) — Groupe d'options qui peuvent être utilisées pour configurer une instance afin de créer et de tester des images de conteneurs.
  - image (chaîne) — L'ID d'AMI à utiliser comme image de base pour une instance de construction et de test de conteneur. Si vous ne spécifiez pas cette valeur, Image Builder utilise l'AMI optimisée Amazon ECS appropriée comme image de base.
  - blockDeviceMappings(ensemble d'objets) — Définit les périphériques en mode bloc à associer pour créer une instance à partir de l'AMI Image Builder spécifiée dans le image paramètre.

    - DeviceName (chaîne) : périphérique auquel ces mappages s'appliquent.
    - ebs (object) — Utilisé pour gérer la configuration spécifique d'Amazon EBS pour ce mappage.
        - deleteOnTermination(Booléen) — Utilisé pour configurer la suppression à la fin de l'appareil associé.
        - crypté (booléen) — Utilisé pour configurer le chiffrement de l'appareil.
        - VolumeSize (entier) — Utilisé pour remplacer la taille du volume de l'appareil.
        - VolumeType (string) — Utilisé pour remplacer le type de volume du périphérique.
- TargetRepository (objet, obligatoire) — Le référentiel de destination pour l'image du conteneur si aucun autre référentiel n'est spécifié dans la configuration de distribution de votre pipeline pour la région où le pipeline s'exécute (région 1).
    - RepositoryName (chaîne, obligatoire) — Nom du référentiel de conteneurs dans lequel l'image de conteneur de sortie est stockée. Ce nom est doté du préfixe de l'emplacement du référentiel.
    - service (chaîne, obligatoire) — Spécifie le service dans lequel cette image a été enregistrée.
- WorkingDirectory (string) — Le répertoire de travail à utiliser lors des workflows de génération et de test.

```
{
 "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. Créez la recette

Utilisez la commande suivante pour créer la recette. Indiquez le nom du fichier JSON que vous avez créé à l'étape précédente dans le `--cli-input-json` paramètre :

```
aws imagebuilder create-container-recipe --cli-input-json file://create-container-recipe.json
```

> **Note**
>
> - Vous devez inclure l'option `file://` au début du chemin du fichier JSON.
> - Le chemin d'accès du fichier JSON doit suivre la convention appropriée pour le système d'exploitation de base sur lequel vous exécutez la commande. En effet, Windows utilise la barre oblique inverse (\) pour faire référence au chemin du répertoire, et Linux utilise la barre oblique directe (/).

Nettoyage des ressources

Pour éviter des frais imprévus, veillez à nettoyer les ressources et les pipelines que vous avez créés à partir des exemples de ce guide. Pour plus d'informations sur la suppression de ressources dans Image Builder, consultezSupprimer les ressources EC2 Image Builder.

Gérer les images EC2 Image Builder

Après avoir créé des ressources d'image pour les images AMI ou de conteneur avec Image Builder, vous pouvez les gérer à l'aide de la console Image Builder, via l'API Image Builder ou à l'aide des imagebuilder commandes du AWS CLI.

> **Tip**
>
> Lorsque vous disposez de plusieurs ressources du même type, le balisage vous aide à identifier une ressource spécifique en fonction des balises que vous lui avez attribuées. Pour plus d'informations sur le balisage de vos ressources à l'aide des commandes Image Builder dans le AWS CLI, consultez la Balisage des ressources section de ce guide.

Cette section explique comment répertorier, afficher et créer des images. Pour plus d'informations sur les flux de production d'images et sur la façon de les gérer, consultezGérez les flux de travail de création et de test pour les images EC2 Image Builder.

Table des matières

Répertorier les images et créer des versions

Sur la page Images de la console Image Builder, vous pouvez consulter la liste de toutes les ressources d'image Image Builder que vous possédez, qui sont partagées avec vous et auxquelles vous avez accès. Les résultats de la liste incluent des informations clés sur ces ressources.

Vous pouvez également voir toutes les images de votre compte pour lesquelles des actions de flux de travail sont en attente.

Table des matières

Liste des images

Cette section décrit les différentes manières de répertorier les informations relatives à vos images.

Vous pouvez utiliser l'une des méthodes suivantes pour répertorier les ressources d'image Image Builder auxquelles vous avez accès. Pour l'action de l'API, voir le ListImagesmanuel de référence de l'API EC2 Image Builder. Pour la demande de SDK associée, reportez-vous au lien Voir aussi sur la même page.

Table des matières

Répertorier les images dans la console

Pour ouvrir la page de liste des images dans la console, procédez comme suit :

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Choisissez Images dans le volet de navigation.

La page Images de la console est divisée en onglets, en fonction de la propriété de l'image ou des actions de flux de travail en attente. Cette section couvre les trois premiers onglets qui affichent les images que vous possédez ou auxquelles vous avez accès.

Onglet console : Je suis propriétaire

Dans l'onglet Owned by me, vous pouvez utiliser les filtres suivants pour rationaliser les résultats de la liste d'images.

- Vous pouvez rechercher tout ou partie du nom dans la barre de recherche.
- Vous pouvez filtrer les images en fonction de la plate-forme de leur système d'exploitation (Windows ou Linux).
- Vous pouvez filtrer les images en fonction du type de sortie qu'elles produisent (AMI ou image de conteneur).
- Vous pouvez utiliser le filtre source pour rechercher des images importées depuis une machine virtuelle avec le VMIE.

Après les contrôles de filtre, l'onglet Owned by me affiche la liste des images Image Builder que vous avez créées, avec les détails suivants pour les ressources répertoriées :

Nom/Version

Les noms des ressources d'image Image Builder commencent par le nom de la recette et la version à partir de laquelle elles ont été créées. Sélectionnez le lien pour voir toutes les versions de génération d'image associées.

Type

Type d'image de sortie créée par Image Builder pour cette ressource d'image (une AMI ou une image de conteneur).

Plateforme

La plate-forme du système d'exploitation de la version de la ressource d'image, par exemple « Windows » ou « Linux ».

Source de l'image

Origine de l'image de base utilisée par Image Builder pour créer cette ressource d'image. Ceci est principalement utilisé pour filtrer les résultats des images importées depuis une machine virtuelle (VMIE).

Heure de création

Date et heure auxquelles Image Builder a créé la version actuelle de la ressource d'image.

ARN

Le nom de ressource Amazon (ARN) de la version actuelle de la ressource image.

Onglet console : partagé avec moi

Dans l'onglet Partagé avec moi, vous pouvez utiliser les filtres suivants pour rationaliser les résultats de la liste d'images.

- Vous pouvez rechercher tout ou partie du nom dans la barre de recherche.
- Vous pouvez filtrer les images en fonction de la plate-forme de leur système d'exploitation (Windows ou Linux).
- Vous pouvez filtrer les images en fonction du type de sortie qu'elles produisent (AMI ou image de conteneur).
- Vous pouvez utiliser le filtre source pour rechercher des images importées depuis une machine virtuelle avec le VMIE.

Après les contrôles de filtre, l'onglet Shared with me affiche une liste des images Image Builder qui ont été partagées avec vous, avec les détails suivants pour les ressources répertoriées :

Nom de l'image

Le nom de la ressource d'image qui a été partagée avec vous. Pour utiliser une image partagée dans une recette, vous devez sélectionner l'option Sélectionner les images gérées et modifier l'origine de l'image en Images partagées avec moi.

Type

Type d'image de sortie créée par Image Builder pour cette ressource d'image (une AMI ou une image de conteneur).

Version

La plate-forme du système d'exploitation de la version de la ressource d'image, par exemple « Windows » ou « Linux ».

Source de l'image

Origine de l'image de base utilisée par Image Builder pour créer cette ressource d'image, le cas échéant. Ceci est principalement utilisé pour filtrer les résultats des images importées depuis une machine virtuelle (VMIE).

Plateforme

La plate-forme du système d'exploitation de la version de la ressource d'image, par exemple « Windows » ou « Linux ».

Heure de création

Date et heure auxquelles Image Builder a créé la version de la ressource d'image qui a été partagée avec vous.

Propriétaire

Le propriétaire de la ressource d'image partagée.

ARN

Le nom de ressource Amazon (ARN) de la version de la ressource image qui a été partagée avec vous.

Onglet console : géré par Amazon

Dans l'onglet Managed by Amazon, vous pouvez utiliser les filtres suivants pour rationaliser les résultats des listes d'images.

- Vous pouvez rechercher tout ou partie du nom dans la barre de recherche.
- Vous pouvez filtrer les images en fonction de la plate-forme de leur système d'exploitation (Windows ou Linux).
- Vous pouvez filtrer les images en fonction du type de sortie qu'elles produisent (AMI ou image de conteneur).
- Vous pouvez utiliser le filtre source pour rechercher des images importées depuis une machine virtuelle avec le VMIE.

Après les contrôles de filtre, l'onglet Managed by Amazon affiche une liste d'images Image Builder gérées par Amazon que vous pouvez utiliser comme images de base pour vos recettes. Image Builder affiche les informations suivantes pour les ressources répertoriées :

Nom de l'image

Nom de l'image gérée. Lorsque vous créez une recette, l'image de base par défaut est Quick start (géré par Amazon). Les images répertoriées dans cet onglet alimentent la liste des noms

d'images associée à la plate-forme du système d'exploitation que vous choisissez pour votre image de base lorsque vous créez une recette.

Type

Type d'image de sortie créée par Image Builder pour cette ressource d'image (une AMI ou une image de conteneur).

Version

La plate-forme du système d'exploitation de la version de la ressource d'image, par exemple « Windows » ou « Linux ».

Plateforme

La plate-forme du système d'exploitation de la version de la ressource d'image, par exemple « Windows » ou « Linux ».

Heure de création

Date et heure auxquelles Image Builder a créé la version de la ressource d'image qui a été partagée avec vous.

Propriétaire

Amazon est propriétaire des images gérées.

ARN

Le nom de ressource Amazon (ARN) de la version de la ressource image qui a été partagée avec vous.

Répertorier les images avec des AWS CLI commandes

Lorsque vous exécutez la list-images commande dans le AWS CLI, vous pouvez obtenir une liste des images que vous possédez ou auxquelles vous avez accès.

L'exemple de commande suivant montre comment utiliser la list-images commande sans filtre pour répertorier toutes les ressources d'image Image Builder que vous possédez.

Exemple : liste toutes les images

```
aws imagebuilder list-images
```

Sortie :

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


Lorsque vous exécutez la list-images commande, vous pouvez appliquer des filtres pour rationaliser les résultats, comme le montre l'exemple suivant. Pour plus d'informations sur le filtrage des résultats, consultez la commande list-images dans la référence des AWS CLI commandes.

Exemple : filtre pour images Linux

```
aws imagebuilder list-images --filters name="platform",values="Linux"
```

Sortie :


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

Répertorier les images en attente d'action

Lorsque vous utilisez l'action d'`WaitForAction`étape dans votre flux de travail d'images, le flux de travail est suspendu jusqu'à ce que vous lui envoyiez un signal lui demandant de reprendre le traitement ou d'échouer. Vous pouvez utiliser cette action d'étape si un processus externe doit être exécuté avant de continuer. Vous pouvez ensuite utiliser le `SendWorkflowStepAction` pour envoyer un signal à l'étape interrompue vers `RESUME` ou`STOP`. Vous pouvez également arrêter ou reprendre votre flux de travail depuis la console.

Les onglets suivants indiquent comment obtenir une liste de toutes les ressources d'image de votre compte avec les étapes du flux de travail actuellement suspendues pour attendre la reprise ou l'arrêt d'un signal. Les onglets couvrent les étapes de la console et la AWS CLI commande.

Vous pouvez également utiliser l'API ou un SDK pour obtenir une liste des étapes du flux de travail en attente d'une action. Pour l'action de l'API, voir le ListWaitingWorkflowStepsmanuel de référence de l'API EC2 Image Builder. Pour la demande de SDK associée, reportez-vous au lien Voir aussi sur la même page.

Console

Pour accéder à l'onglet En attente d'action de la console, procédez comme suit :

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Choisissez Images dans le volet de navigation. Cela ouvre la page de liste des images.
3. Sélectionnez l'onglet En attente d'action dans la page de liste.
4. (facultatif) Pour arrêter ou reprendre une étape, cochez la case à côté du nom, puis choisissez Arrêter l'étape ou Reprendre l'étape. Vous pouvez cocher plusieurs cases pour exécuter la même action pour toutes les étapes sélectionnées.

Détails des étapes du flux de travail en attente

Les détails du flux de travail pour l'étape en attente sont les suivants :

- Nom de l'image : nom de la ressource d'image dont l'étape est en attente. Vous pouvez sélectionner le lien du nom pour afficher la page détaillée de cette image.
- Nom de l'étape en attente : nom de l'étape du flux de travail en attente d'une action.
- ID d'exécution de l'étape : identifie de manière unique l'instance d'exécution de l'étape du flux de travail. Vous pouvez sélectionner l'ID lié pour afficher les détails d'exécution de l'étape.
- Début de l'étape : horodatage du début de l'instance d'exécution de l'étape du flux de travail.
- ARN du flux de travail : nom de ressource Amazon (ARN) du flux de travail avec l'étape en attente.
- Actions : action d'étape en état d'attente.

AWS CLI

Lorsque vous exécutez la list-waiting-workflow-steps commande dans le AWS CLI, vous obtenez une liste de toutes les images de votre compte dont les étapes du flux de travail sont en attente d'une action avant de terminer le processus de création d'image.

L'exemple de commande suivant montre comment utiliser la list-waiting-workflow-steps commande pour répertorier toutes les images de votre compte avec les étapes du flux de travail en attente d'une action.

Exemple : répertorier les images de votre compte avec les étapes du flux de travail en attente

```
aws imagebuilder list-waiting-workflow-steps
```

Sortie :

Le résultat de cet exemple montre une image du compte avec une étape en attente d'action.

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

Lister les versions de génération d'images

Sur la page des versions de génération d'images de la console Image Builder, vous pouvez consulter la liste des versions de génération et des informations supplémentaires sur une ressource d'image dont vous êtes le propriétaire. Vous pouvez également utiliser des commandes ou des actions avec l'API Image Builder, les SDK ou AWS CLI pour répertorier les versions de génération d'image.

Vous pouvez utiliser l'une des méthodes suivantes pour répertorier les versions de génération d'image pour les ressources d'images que vous possédez. Pour l'action de l'API, voir le ListImageBuildVersionsmanuel de référence de l'API EC2 Image Builder. Pour la demande de SDK associée, reportez-vous au lien Voir aussi sur la même page.

Console

Détails de la version

La page des versions d'Image build de la console Image Builder contient les informations suivantes :

- Version — Version de génération de la ressource image. Dans la console Image Builder, la version renvoie à une page détaillée de l'image.
- Type : type de sortie distribué par Image Builder lors de la création de cette ressource d'image (une AMI ou une image de conteneur).
- Date de création : date et heure auxquelles Image Builder a créé la version de génération de l'image.
- État de l'image — État actuel de la version créée par l'image. L'état peut être lié à la création ou à la disposition de l'image. Par exemple, pendant le processus de création, vous pouvez voir le statut `Building` ou`Distributing`. Pour la disposition de l'image, le statut `Deprecated` ou peut s'afficher`Deleted`.
- Raison de l'échec : raison de l'état de l'image. La console Image Builder affiche uniquement la raison de l'échec de la génération (le statut de l'image est égal à`Failed`).
- Résultats de sécurité : résultats de numérisation d'images agrégés pour la version de génération d'image référencée.

- ARN — Le nom de ressource Amazon (ARN) pour la version référencée de la ressource image.
- Flux de journal : lien vers le détail du flux de journal pour la version de génération de l'image référencée.

Versions de liste

Pour répertorier les versions de génération d'image dans la console Image Builder, effectuez les opérations suivantes :

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Choisissez Images dans le volet de navigation. Par défaut, la liste des images affiche la version actuelle de chacune des images que vous possédez.
3. Pour voir la liste de toutes les versions d'une image, cliquez sur le lien de la version actuelle. Le lien ouvre la page des versions de génération d'image qui répertorie toutes les versions de génération pour une image spécifique.

AWS CLI

Lorsque vous exécutez la list-image-build-versions commande dans le AWS CLI, vous obtenez une liste complète des versions de build pour la ressource d'image spécifiée. Vous devez posséder l'image pour exécuter cette commande.

L'exemple de commande suivant montre comment utiliser la list-image-build-versions commande pour répertorier toutes les versions de compilation pour l'image spécifiée.

Exemple : liste des versions de compilation pour une image spécifique

```
aws imagebuilder list-image-build-versions --image-version-arn
 arn:aws:imagebuilder:us-west-2:123456789012:image/image-recipe-name/1.0.0
```

Sortie :

La sortie de cet exemple inclut deux versions de génération pour la recette d'image spécifiée.

```
{
  "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**
>
> Le résultat de la list-image-build-versions commande n'inclut pas les résultats de sécurité ni les flux de journaux pour le moment.

Afficher les détails de l'image

Sur la page des détails de l'image de la console Image Builder, vous pouvez consulter les détails d'une ressource d'image spécifique dont vous êtes le propriétaire. Vous pouvez également utiliser des commandes ou des actions avec l'API Image Builder, les SDK ou AWS CLI pour obtenir des informations détaillées sur les images.

Pour plus d'informations sur les ressources Compte AWS partagées avec vous par le biais d'un partage de ressources AWS Resource Access Manager (AWS RAM), consultez la section Accès aux AWS ressources partagées avec vous dans le Guide de AWS RAM l'utilisateur.

Table des matières

Afficher les détails de l'image dans la console Image Builder

La page détaillée de l'image dans la console Image Builder inclut une section récapitulative, avec des informations supplémentaires regroupées sous forme d'onglets. L'en-tête de page correspond au nom et à la version de compilation de la recette qui a créé l'image.

Sections et onglets détaillés de la console

- Onglet Ressources de sortie
- Onglet Configuration de l'infrastructure
- Onglet Paramètres de distribution
- Onglet Workflow
- Onglet Constatations relatives
- Onglet Balises

Section récapitulative

La section récapitulative couvre toute la largeur de la page et inclut les détails suivants. Ces informations sont toujours affichées.

Recette

Le nom de la recette et la version qui n'incluent pas la version de compilation. Par exemple, si la version de construction l'est`sample-linux-recipe | 1.0.1/2`, la recette l'est `sample-linux-recipe | 1.0.1` et la version de construction l'est2.

Date de création

Date et heure auxquelles Image Builder a créé la version de génération de l'image.

État de l'image

État actuel de la version de création de l'image. L'état peut être lié à la création ou à la disposition de l'image. Par exemple, pendant le processus de création, vous pouvez voir le statut `Building` ou`Distributing`. Pour la disposition de l'image, le statut `Deprecated` ou peut s'afficher`Deleted`.

Raison de l'échec

La raison de l'état de l'image. La console Image Builder affiche uniquement la raison de l'échec de la génération (le statut de l'image est égal à`Failed`).

Onglet Ressources de sortie

L'onglet Ressources de sortie répertorie les détails de sortie et de distribution de la ressource image actuellement affichée. Les informations affichées par Image Builder dépendent du type de recette utilisé par le pipeline pour créer l'image, comme suit.

Recette d'images

- Région : région de distribution pour l'Amazon Machine Image (AMI) en sortie spécifiée dans la colonne Image.
- Image : ID de l'AMI qu'Image Builder a distribuée à la destination. Cet identifiant est lié à la page Amazon Machine Images (AMI) dans la console Amazon EC2.

> **Note**
>
> Image Builder crée l'AMI après avoir créé la ressource d'image de sortie et avant de distribuer l'AMI à la destination.

- Nom : nom de l'AMI qu'Image Builder a distribuée à la destination.
- Description — Description facultative de la recette d'image utilisée par le pipeline pour créer la ressource d'image de sortie.
- Compte : propriétaire Compte AWS de la ressource d'image Image Builder actuellement affichée.

Recette de contenant

Image Builder affiche les détails suivants pour la sortie créée à partir d'une recette de conteneur.

- Région : région de distribution de l'image du conteneur spécifiée dans la colonne URI de l'image.
- URI de l'image : URI de l'image du conteneur de sortie qu'Image Builder a distribuée au référentiel ECR de la région de destination.

> **Note**
>
> Image Builder affiche une ligne par destination. L'image de sortie comporte toujours au moins une entrée destinée à être distribuée au compte qui a créé l'image. Les destinations supplémentaires peuvent inclure des distributions entre les régions Comptes AWS, ou AWS Organizations. Pour plus d'informations, consultez Gérer les paramètres de distribution d'EC2 Image Builder.

Onglet Configuration de l'infrastructure

L'onglet Configuration de l'infrastructure affiche les paramètres d'infrastructure Amazon EC2 utilisés par Image Builder pour créer et tester l'image actuellement affichée. Image Builder affiche toujours

le nom de la ressource de configuration de l'infrastructure (nom de configuration) et son Amazon Resource Name (ARN). Si la configuration de votre infrastructure définit les valeurs, les détails supplémentaires de l'infrastructure peuvent inclure les éléments suivants

- Types d'instances
- Un profil d'instance
- Infrastructure réseau
- Paramètres du groupe de sécurité
- Un emplacement Amazon S3 où Image Builder stocke les journaux des applications
- Une paire de clés Amazon EC2 pour le dépannage
- Rubrique Amazon SNS pour les notifications d'événements

Pour plus d'informations, consultez Gérer la configuration de l'infrastructure EC2 Image Builder.

Onglet Paramètres de distribution

L'onglet Paramètres de distribution affiche les paramètres utilisés par Image Builder pour distribuer vos images de sortie. Image Builder affiche toujours le nom de la ressource de configuration de distribution (nom de configuration) et son Amazon Resource Name (ARN). Les détails de distribution supplémentaires dépendent du type de recette utilisé par le pipeline Image Builder pour créer l'image, comme suit :

Recette d'images

Si votre ressource de configuration de distribution définit les valeurs, les détails de distribution supplémentaires peuvent inclure les éléments suivants :

- Région : région de distribution pour l'Amazon Machine Image (AMI) en sortie.
- Nom de l'AMI de sortie : nom de l'AMI qu'Image Builder a distribuée à la destination.
- Chiffrement (clé KMS) : si configuré, Image Builder utilise pour chiffrer l'image afin de la distribuer dans la région cible. AWS KMS key
- Comptes cibles pour la distribution : si vous avez configuré la distribution entre comptes, cette colonne affiche une liste séparée par des virgules des personnes avec Comptes AWS lesquelles partager l'image de sortie dans la région cible.
- Principaux dotés d'autorisations partagées : liste séparée par des virgules des AWS principaux autorisés à lancer votre image, par exemple, des groupes Comptes AWS ou des unités organisationnelles (AWS Organizations UO).

> **Note**
>
> Lorsque vous autorisez d'autres personnes à lancer votre image, vous êtes toujours propriétaire de l'image. AWS facture votre compte pour toutes les instances lancées par Amazon EC2 à partir de votre image.

- Comptes cibles pour une configuration de lancement plus rapide :
- Configurations de licence associées : les ARN de configuration de licence du License Manager à associer à l'AMI dans la région spécifiée.
- Configuration du modèle de lancement —
- Définir la version par défaut du modèle de lancement —

Recette de contenant

Les distributions de conteneurs incluent toujours les informations suivantes :

- Région : région de distribution pour l'image du conteneur spécifiée dans la colonne URI de l'image.
- URI de l'image : URI de l'image du conteneur de sortie qu'Image Builder a distribuée au référentiel Amazon ECR dans la région de destination.

> **Note**
>
> Image Builder affiche une ligne par destination. L'image de sortie comporte toujours au moins une entrée destinée à être distribuée au compte qui a créé l'image. Les destinations supplémentaires peuvent inclure des distributions entre les régions Comptes AWS, ou AWS Organizations. Pour plus d'informations, consultez Gérer les paramètres de distribution d'EC2 Image Builder.

Onglet Workflow

Les flux de travail définissent la séquence d'étapes qu'Image Builder exécute lorsqu'il crée une nouvelle image. Toutes les images ont des flux de travail de création et de test. Les conteneurs disposent d'un flux de travail supplémentaire pour la distribution. L'onglet Workflow affiche les flux de travail applicables exécutés par Image Builder pour votre image.

Types de flux de travail de filtrage

Image Builder affiche initialement le résumé du flux de travail de création et les étapes du flux de travail par défaut. Cependant, le filtre de flux de travail affiche tous les flux de travail en cours ou terminés pour votre image. Pour afficher un autre flux de travail, sélectionnez-le dans la liste comme suit :

Workflows d'imagerie (sortie AMI)

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

Flux de travail liés aux conteneurs (sortie des conteneurs)

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

> **Note**
>
> Si le flux de travail n'a pas encore démarré, il n'apparaît pas dans la liste. Par exemple, si la création de votre image vient de commencer, `build-image` c'est le seul type de flux de travail qui apparaît dans la liste. Lorsque le flux de travail suivant commence, `test-image` dans ce cas, Image Builder l'ajoute à la liste.

Après le filtre de flux de travail, le flux de travail sélectionné affiche un résumé de l'exécution qui inclut les détails suivants pour chaque type de flux de travail :

État du flux de travail

État d'exécution actuel de ce flux de travail. Les valeurs peuvent inclure les éléments suivants :

- En attente
- Ignoré
- En cours d'exécution
- Terminé
- Échec

- R ollback-in-progress
- Annulation terminée

ID d'exécution

Identifiant unique attribué par Image Builder pour suivre les ressources d'exécution chaque fois qu'il exécute un flux de travail.

Démarrer

Horodatage du démarrage de l'instance d'exécution de ce flux de travail.

Fin

Horodatage auquel cette instance d'exécution du flux de travail s'est terminée.

Nombre total d'étapes

Nombre total d'étapes du flux de travail. Cela doit être égal à la somme du nombre d'étapes réussies, ignorées et échouées.

Étapes réussies

Nombre d'exécution correspondant au nombre d'étapes du flux de travail exécutées avec succès.

Les étapes ont échoué

Nombre d'exécution correspondant au nombre d'étapes du flux de travail qui ont échoué.

Étapes ignorées

Nombre d'exécution correspondant au nombre d'étapes du flux de travail qui ont été ignorées.

Les détails de la liste suivante indiquent l'état actuel de toutes les étapes de cette instance d'exécution du flux de travail. Image Builder affiche les mêmes détails pour tous les types d'images.

Étape #

Un nombre qui représente l'ordre dans lequel Image Builder exécute les étapes du flux de travail.

ID de l'étape

Identifiant unique pour l'étape du flux de travail, attribué lors de l'exécution.

État de l'étape

État d'exécution actuel de l'étape de flux de travail spécifiée.

État de la rétrogradation

État actuel de la restauration en cas d'échec de cette instance d'exécution du flux de travail.

Nom de l'étape

Nom de l'étape de flux de travail spécifiée.

Démarrer

Horodatage du début de l'étape spécifiée pour cette instance d'exécution du flux de travail.

Fin

Horodatage de fin de l'étape spécifiée pour cette instance d'exécution du flux de travail.

Onglet Constatations relatives

Si vous avez activé le scan, l'onglet Résultats de sécurité affiche les résultats relatifs aux vulnérabilités et expositions courantes (CVE). Amazon Inspector a identifié ces résultats sur l'instance de test lancée par Image Builder pour créer votre nouvelle image. Pour garantir qu'Image Builder capture les résultats de votre image, vous devez configurer la numérisation comme suit :

1. Activez les scans Amazon Inspector pour votre compte. Pour plus d'informations, consultez Getting started with Amazon Inspector dans le guide de l'utilisateur d'Amazon Inspector.
2. Activez les résultats de sécurité pour le pipeline qui crée cette image. Lorsque vous activez les résultats de sécurité pour votre pipeline, Image Builder enregistre un instantané des résultats avant de mettre fin à l'instance de test. Pour plus d'informations, consultez Configurez les scans de sécurité pour les images Image Builder dans AWS Management Console.

L'onglet Résultats de sécurité inclut les informations suivantes pour chaque vulnérabilité identifiée par Amazon Inspector pour votre image.

Sévérité

Le niveau de gravité du résultat de la CVE. Les valeurs sont les suivantes :

- Non trié
- Informationnel
- Faible
- Medium

- Élevée
- Critique

ID de résultat

Identifiant unique du résultat CVE détecté par Amazon Inspector pour votre image lors de la numérisation de l'instance de test. L'ID est lié à la page Résultats de sécurité > Par vulnérabilité. Pour plus d'informations, consultez Gérez les résultats de sécurité relatifs aux images Image Builder dans AWS Management Console.

Source

Source des informations sur la vulnérabilité utilisées pour la découverte de la CVE.

Âge

Nombre de jours écoulés depuis que le résultat a été observé pour la première fois pour votre image.

Note de l'Inspecteur

Le score attribué par Amazon Inspector à la recherche CVE.

Onglet Balises

L'onglet Tags affiche tous les tags que vous avez définis pour votre image.

Obtenir les détails de la politique en matière d'image (AWS CLI)

L'exemple suivant montre comment obtenir les détails d'une politique relative aux images à l'aide de son Amazon Resource Name (ARN).

```
aws imagebuilder get-image-policy --image-arn arn:aws:imagebuilder:us-
west-2:123456789012:image/example-image/2019.12.02
```

Créez des images

Cette section explique comment créer des images Image Builder et annuler une compilation en cours.

Table des matières

- Annuler la création de l'image (AWS CLI)

Créer une image

Vous pouvez créer une nouvelle image Image Builder de différentes manières. Par exemple, vous pouvez utiliser l'une des méthodes suivantes pour créer une image avec le AWS Management Console ou AWS CLI. Vous pouvez également utiliser l'action CreateImageAPI. Pour la demande de SDK associée, vous pouvez vous référer au lien Voir aussi correspondant à cette commande dans le manuel EC2 Image Builder API Reference.

AWS Management Console

Pour créer une nouvelle image à partir d'un pipeline existant, vous pouvez exécuter le pipeline manuellement, comme suit. Vous pouvez également utiliser l'assistant de pipeline pour créer une nouvelle image à partir de zéro. Voir Création d'un pipeline d'images (AMI) ouCréation d'un pipeline d'images (Docker), selon le type d'image que vous souhaitez créer.

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Choisissez Image pipelines dans le volet de navigation.
3. Cochez la case à côté du nom du pipeline que vous souhaitez exécuter.
4. Pour créer l'image, sélectionnez Exécuter le pipeline dans le menu Actions. Cela démarre le pipeline.

Vous pouvez également définir un calendrier pour exécuter votre pipeline ou utiliser Amazon EventBridge pour exécuter votre pipeline en fonction des règles que vous configurez.

AWS CLI

Avant d'exécuter la create-image commande dans le AWS CLI, vous devez créer les ressources suivantes si elles n'existent pas déjà :

Ressources requises

- Recette — Vous devez spécifier exactement une recette pour votre image, comme suit :

  Recette d'images

  Spécifiez le nom de ressource Amazon (ARN) pour votre ressource de recette d'image à l'aide du `--image-recipe-arn` paramètre.

Recette de contenant

Spécifiez l'ARN de votre ressource de recette de conteneur à l'aide du `--container-recipe-arn` paramètre.

- Configuration de l'infrastructure — Spécifiez l'ARN de votre ressource de configuration d'infrastructure à l'aide du `--infrastructure-configuration-arn` paramètre.

Vous pouvez également spécifier l'une des ressources suivantes dont votre image a besoin :

Ressources et configuration facultatives

- Configuration de distribution — Par défaut, Image Builder distribue la ressource d'image de sortie à votre compte dans la région où vous exécutez la create-image commande. Pour fournir des destinations ou une configuration supplémentaires pour votre distribution, spécifiez l'ARN de votre ressource de configuration de distribution à l'aide du `--distribution-configuration-arn` paramètre.
- Numérisation d'images : pour configurer des instantanés correspondant aux résultats d'Amazon Inspector sur votre instance de test d'image ou de conteneur, utilisez le `--image-scanning-configuration` paramètre. Pour les images de conteneurs, vous spécifiez également le référentiel ECR qu'Amazon Inspector utilise pour ses scans.
- Tests d'image : pour supprimer la phase de test Image Builder, utilisez le `--image-tests-configuration` paramètre. Vous pouvez également définir un délai d'expiration pour sa durée d'exécution.
- Balises d'image : utilisez le `--tags` paramètre pour ajouter des balises à votre image de sortie.
- Flux de travail d'images : si vous ne spécifiez aucun flux de création ou de test, Image Builder crée votre image avec son flux de travail d'image par défaut. Pour spécifier les flux de travail que vous avez créés, utilisez le `--workflows` paramètre.

> **Note**
>
> Si vous spécifiez des flux de travail d'image, vous devez également fournir le nom ou l'ARN du rôle IAM qu'Image Builder utilise pour exécuter vos actions de flux de travail dans le `--execution-role` paramètre.

L'exemple suivant montre comment créer une image à l'aide de la commande create-image AWS CLI . Pour plus d'informations, consultez la référence de la commande AWS CLI .

Exemple : création d'une image de base avec une distribution par défaut

```
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
```

Sortie :

```
{
    "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",
            ...
        }
    ]
}
```

Annuler la création de l'image (AWS CLI)

Pour annuler une création d'image en cours, utilisez la cancel-image-creation commande suivante :

```
aws imagebuilder cancel-image-creation --image-build-version-arn
 arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-recipe/2019.12.03/1
```

Importer une image de machine virtuelle

Image Builder s'intègre à l'API Amazon EC2 VM Import/Export pour permettre au processus d'importation de s'exécuter de manière asynchrone en arrière-plan. Image Builder fait référence à l'ID de tâche issu de l'importation de la machine virtuelle pour suivre sa progression, et crée une ressource image Image Builder en sortie. Cela vous permet de référencer la ressource image Image Builder dans vos recettes avant la fin de l'importation de la machine virtuelle.

Importer une machine virtuelle (console)

Pour importer une machine virtuelle avec la console Image Builder, procédez comme suit :

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Choisissez Images dans le volet de navigation.
3. Choisissez Importer une image.
4. Fournissez des informations sur les sections suivantes sur la page Importer une image. Choisissez ensuite Importer une image lorsque vous avez terminé.

Général

1. Spécifiez un nom unique pour l'image de base.
2. Spécifiez une version pour l'image de base. Utilisez le format suivant : *major*.*minor*.*patch*.
3. Vous pouvez également saisir une description facultative pour l'image de base.

Système d'exploitation de l'image de base

1. Sélectionnez l'option Système d'exploitation d'image (OS) correspondant à votre plate-forme VM OS.
2. Sélectionnez la version du système d'exploitation qui correspond à la version de votre machine virtuelle dans la liste.

Configuration de l'importation de machines virtuelles

Lorsque vous exportez votre machine virtuelle depuis son environnement de virtualisation, ce processus crée un ensemble d'un ou plusieurs fichiers de conteneur de disque. Ils agissent comme des instantanés de l'environnement, des paramètres et des données de votre machine virtuelle. Vous pouvez utiliser ces fichiers pour importer votre machine virtuelle en tant qu'image de base pour votre recette d'image. Pour plus d'informations sur l'importation de machines virtuelles dans Image Builder, consultezImportation et exportation d'images de machines virtuelles.

Pour spécifier l'emplacement de votre source d'importation, procédez comme suit :

Source d'importation

Spécifiez la source du premier conteneur de disque ou d'instantané d'image de machine virtuelle à importer dans la section Conteneur de disque 1.

1. Source : il peut s'agir d'un compartiment S3 ou d'un instantané EBS.
2. Sélectionnez l'emplacement S3 du disque : entrez l'emplacement dans Amazon S3 où vos images de disque sont stockées. Pour rechercher l'emplacement, choisissez Parcourir S3.
3. Pour ajouter un conteneur de disques, choisissez Ajouter un conteneur de disques.

Rôle IAM

Pour associer un rôle IAM à la configuration d'importation de votre machine virtuelle, sélectionnez le rôle dans la liste déroulante des rôles IAM ou choisissez Créer un nouveau rôle pour en créer un nouveau. Si vous créez un nouveau rôle, la page de console IAM Roles s'ouvre dans un onglet distinct.

Réglages avancés — optionnel

Les paramètres suivants sont facultatifs. Avec ces paramètres, vous pouvez configurer le chiffrement, les licences, les balises, etc., pour l'image de base créée par l'importation.

Architecture d'image de base

Pour spécifier l'architecture de la source d'importation de votre machine virtuelle, sélectionnez une valeur dans la liste Architecture.

Chiffrement

Si les images de disque de votre machine virtuelle sont chiffrées, vous devez fournir une clé à utiliser pour le processus d'importation. Pour spécifier une clé KMS pour l'importation, sélectionnez une valeur dans la liste Chiffrement (clé KMS). La liste contient les clés KMS auxquelles votre compte a accès dans la région actuelle.

Gestion des licences

Lorsque vous importez une machine virtuelle, le processus d'importation détecte automatiquement le système d'exploitation de la machine virtuelle et applique la licence appropriée à l'image de base. Selon la plate-forme de votre système d'exploitation, les types de licence sont les suivants :

- Licence incluse : une AWS licence adaptée à votre plateforme est appliquée à votre image de base.

- Apportez votre propre licence (BYOL) : conserve la licence de votre machine virtuelle, le cas échéant.

Pour associer des configurations de licence créées avec AWS License Manager à votre image de base, sélectionnez-les dans la liste des noms de configuration de licence. Pour plus d'informations sur License Manager, consultez Working with AWS License Manager

> **Note**
>
> - Les configurations de licence contiennent des règles de licence basées sur les termes de vos contrats d'entreprise.
> - Linux ne prend en charge que les licences BYOL.

Balises (image de base)

Les balises utilisent des paires clé-valeur pour attribuer du texte consultable à votre ressource Image Builder. Pour spécifier des balises pour l'image de base importée, entrez des paires clé-valeur à l'aide des cases Clé et Valeur.

Pour ajouter une identification, choisissez Ajouter une identification. Pour supprimer une balise, choisissez Remove tag (Supprimer une balise).

Importer une machine virtuelle (AWS CLI)

Pour importer une machine virtuelle depuis des disques vers une AMI et créer une ressource d'image Image Builder à laquelle vous pouvez immédiatement faire référence, procédez comme suit à partir du AWS CLI :

1. Lancez une importation de machine virtuelle à l'aide de la commande Amazon EC2 VM Import/Export import-image dans le. AWS CLI Notez l'ID de tâche renvoyé dans la réponse à la commande. Vous en aurez besoin pour la prochaine étape. Pour plus d'informations, consultez la section Importation d'une machine virtuelle sous forme d'image à l'aide de VM Import/Export dans le guide de l'utilisateur de VM Import/Export.

2. Créer un fichier JSON d'entrée CLI

Pour rationaliser la import-vm-image commande Image Builder utilisée dans le AWS CLI, nous créons un fichier JSON contenant toutes les configurations d'importation que nous voulons transmettre à la commande.

> **Note**
>
> La convention de dénomination des valeurs de données dans le fichier JSON suit le modèle spécifié pour les paramètres de demande d'action de l'API Image Builder. Pour consulter les paramètres de demande de commande d'API, consultez la ImportVmImagecommande dans le manuel de référence de l'API EC2 Image Builder. Pour fournir les valeurs de données sous forme de paramètres de ligne de commande, reportez-vous aux noms de paramètres spécifiés dans la AWS CLI Command Reference. À la import-vm-image commande Image Builder en tant qu'options.

Voici un résumé des paramètres que nous indiquons dans cet exemple :

- name (chaîne, obligatoire) : nom de la ressource d'image Image Builder à créer en sortie de l'importation.
- SemanticVersion <major>(chaîne, obligatoire) — Version sémantique de l'image de sortie qui spécifie la version au format suivant, avec des valeurs numériques à chaque position pour indiquer une version spécifique :. <minor>. <patch>. Par exemple, `1.0.0`. Pour en savoir plus sur le versionnement sémantique des ressources Image Builder, consultez. Gestion des versions sémantique
- description (chaîne) — Description de la recette d'image.
- platform (string, obligatoire) : plate-forme du système d'exploitation de la machine virtuelle importée.
- vmImportTaskId (chaîne, obligatoire) — Le `ImportTaskId` (AWS CLI) issu du processus d'importation de la machine virtuelle Amazon EC2. Image Builder surveille le processus d'importation pour intégrer l'AMI qu'il crée et créer une ressource d'image Image Builder qui peut être utilisée immédiatement dans des recettes.
- ClientToken (chaîne, obligatoire) — Identifiant unique distinguant majuscules et minuscules que vous fournissez pour garantir l'idempuissance de la demande. Pour plus d'informations, consultez la section Garantir l'idempotencie dans le manuel Amazon EC2 API Reference.

- tags (chaîne de caractères) — Les tags sont des paires clé-valeur associées aux ressources d'importation. Jusqu'à 50 paires clé-valeur sont autorisées.

Enregistrez le fichier sous`import-vm-image.json`, pour l'utiliser dans la import-vm-image commande 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. Importer l'image

Exécutez la import-vm-image commande en utilisant le fichier que vous avez créé en entrée :

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

> **Note**
>
> - Vous devez inclure l'option `file://` au début du chemin du fichier JSON.
> - Le chemin d'accès du fichier JSON doit suivre la convention appropriée pour le système d'exploitation de base sur lequel vous exécutez la commande. En effet, Windows utilise la barre oblique inverse (\) pour faire référence au chemin du répertoire, et Linux utilise la barre oblique directe (/).

Gestion des résultats de sécurité pour les images Image Builder

Lorsque vous activez le scan de sécurité avec Amazon Inspector, celui-ci analyse en permanence les images des machines et les instances en cours d'exécution de votre compte pour détecter les vulnérabilités du système d'exploitation et du langage de programmation. Lorsqu'elle est activée,

l'analyse de sécurité est automatique et Image Builder peut enregistrer un instantané des résultats de votre instance de test lorsque vous créez une nouvelle image. Amazon Inspector est un service payant.

Lorsqu'Amazon Inspector découvre des vulnérabilités dans votre logiciel ou dans vos paramètres réseau, il prend les mesures suivantes :

- Vous informe qu'une découverte a été faite.
- Évalue la gravité du résultat. L'indice de gravité classe les vulnérabilités afin de vous aider à hiérarchiser vos découvertes, et inclut les valeurs suivantes :
  - Non trié
  - Informationnel
  - Faible
  - Medium
  - Élevée
  - Critique
- Fournit des informations sur la découverte et des liens vers des ressources supplémentaires pour plus de détails.
- Fournit des conseils de correction pour vous aider à résoudre les problèmes à l'origine du résultat.

Configurez les scans de sécurité pour les images Image Builder dans AWS Management Console

Si vous avez activé Amazon Inspector pour votre compte, Amazon Inspector scanne automatiquement les instances EC2 lancées par Image Builder pour créer et tester une nouvelle image. Ces instances ont une courte durée de vie pendant le processus de création et de test, et leurs résultats expirent normalement dès leur fermeture. Pour vous aider à étudier et corriger les résultats de votre nouvelle image, Image Builder peut éventuellement enregistrer sous forme de capture instantanée tous les résultats identifiés par Amazon Inspector sur votre instance de test pendant le processus de création.

Étape 1 : activer les scans de sécurité Amazon Inspector pour votre compte

Pour activer les scans de sécurité Amazon Inspector pour votre compte depuis la console Image Builder, procédez comme suit :

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.

2. Choisissez les paramètres d'analyse de sécurité dans le volet de navigation. Cela ouvre la boîte de dialogue d'analyse de sécurité.

   La boîte de dialogue affiche l'état de numérisation de votre compte. Si Amazon Inspector est déjà activé pour votre compte, le statut indique Activé.

3. Suivez les étapes 1 et 2 des instructions pour activer le scan Amazon Inspector.

> **Note**
>
> Amazon Inspector encourt des frais. Pour plus d'informations, consultez la Tarification d'Amazon Inspector.

Si vous avez activé la numérisation pour votre pipeline, Image Builder prend un instantané des résultats pour votre instance de build lorsque vous créez une nouvelle image. De cette façon, vous pouvez accéder aux résultats une fois qu'Image Builder a mis fin à l'instance de génération.

Étape 2 : configurer votre pipeline pour enregistrer des instantanés afin de détecter les vulnérabilités

Pour configurer les instantanés de détection des vulnérabilités pour votre pipeline, effectuez les opérations suivantes :

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Choisissez Image pipelines dans le volet de navigation.
3. Choisissez l'une des méthodes suivantes pour spécifier les détails du pipeline :

   Création d'un nouveau pipeline

   1. Sur la page Pipelines d'images, choisissez Créer un pipeline d'images. Cela ouvre la page Spécifier les détails du pipeline dans l'assistant du pipeline.

   Mettre à jour un pipeline existant

   1. Sur la page Pipelines d'images, cliquez sur le lien du nom du pipeline que vous souhaitez mettre à jour. Cela ouvre la page détaillée du pipeline.

> **Note**
>
> Vous pouvez également cocher la case à côté du nom du pipeline que vous souhaitez mettre à jour, puis choisir Afficher les détails.

2. Sur la page des détails du pipeline, sélectionnez Modifier le pipeline dans le menu Action. Cela vous amène à la page Modifier le pipeline.

4. Dans la section Général de l'assistant de pipeline ou de la page Modifier le pipeline, cochez la case Activer le scan de sécurité.

> **Note**
>
> Si vous souhaitez désactiver les instantanés ultérieurement, vous pouvez modifier votre pipeline pour désactiver la case à cocher. Cela ne désactive pas le scan Amazon Inspector pour votre compte. Pour désactiver le scan Amazon Inspector, consultez la section Désactivation d'Amazon Inspector dans le guide de l'utilisateur d'Amazon Inspector.

Gérez les résultats de sécurité relatifs aux images Image Builder dans AWS Management Console

Les pages de liste des résultats de sécurité affichent des informations de haut niveau sur les résultats relatifs à vos ressources, avec des vues basées sur plusieurs filtres différents que vous pouvez appliquer. Chaque vue inclut les options suivantes en haut pour modifier votre vue :

- Tous les résultats de sécurité : il s'agit de l'affichage par défaut si vous choisissez la page Résultats de sécurité dans le volet de navigation de la console Image Builder.
- Par vulnérabilité : cette vue affiche une liste détaillée de toutes les ressources d'images de votre compte contenant des résultats. L'ID de recherche est lié à des informations plus détaillées sur le résultat. Ces informations apparaissent sur un panneau qui s'ouvre sur le côté droit de la page. Le panneau contient les informations suivantes :
  - Description détaillée de la découverte.
  - Un onglet Détails de recherche. Cet onglet inclut un aperçu des résultats, les packages concernés, des conseils de correction sommaires, des informations détaillées sur les vulnérabilités et les vulnérabilités associées. L'identifiant de vulnérabilité renvoie à des

informations détaillées sur les vulnérabilités dans la base de données nationale sur les vulnérabilités.

- Un onglet de répartition des scores. Cet onglet inclut une side-by-side comparaison des scores CVSS et Amazon Inspector afin que vous puissiez voir où Amazon Inspector a modifié un score, le cas échéant.

- Par pipeline d'images : cette vue indique le nombre de résultats pour chaque pipeline d'images de votre compte. Image Builder affiche le nombre de résultats de gravité moyenne et supérieure, ainsi que le total de tous les résultats. Toutes les données de la liste sont liées comme suit :
  - La colonne Nom du pipeline d'images renvoie à la page détaillée du pipeline d'images spécifié.
  - Les liens des colonnes du niveau de gravité ouvrent la vue Tous les résultats de sécurité, filtrée en fonction du nom du pipeline d'images et du niveau de gravité associés.

  Vous pouvez également utiliser des critères de recherche pour affiner vos résultats.
- Par image : cette vue indique le nombre de résultats pour chaque image créée dans votre compte. Image Builder affiche le nombre de résultats de gravité moyenne et supérieure, ainsi que le total de tous les résultats. Toutes les données de la liste sont liées comme suit :
  - La colonne Nom de l'image renvoie à la page détaillée de l'image pour la génération d'image spécifiée. Pour plus d'informations, consultez Afficher les détails de l'image.
  - Les liens des colonnes du niveau de gravité ouvrent la vue Tous les résultats de sécurité, filtrée en fonction du nom de version de l'image et du niveau de gravité associés.

  Vous pouvez également utiliser des critères de recherche pour affiner vos résultats.

Image Builder affiche les détails suivants dans la section Liste des résultats de la vue par défaut Tous les résultats de sécurité.

Sévérité

Niveau de gravité du résultat de la CVE. Les valeurs sont les suivantes :

- Non trié
- Informationnel
- Faible
- Medium
- Élevée
- Critique

ID de résultat

Identifiant unique du résultat CVE détecté par Amazon Inspector pour votre image lorsqu'il a scanné l'instance de build. L'ID est lié à la page Résultats de sécurité > Par vulnérabilité.

ARN de l'image

Le nom de ressource Amazon (ARN) de l'image dont le résultat est spécifié dans la colonne Identifiant de recherche.

Pipeline

Le pipeline qui a créé l'image spécifiée dans la colonne Image ARN.

Description

Brève description de la découverte.

Note de l'Inspecteur

Le score attribué par Amazon Inspector à la recherche CVE.

Assainissement

Liens vers des informations sur le plan d'action recommandé pour remédier à la constatation.

Date de publication

Date et heure auxquelles cette vulnérabilité a été ajoutée pour la première fois à la base de données du fournisseur.

Nettoyage des ressources

Pour éviter des frais imprévus, veillez à nettoyer les ressources et les pipelines que vous avez créés à partir des exemples de ce guide. Pour plus d'informations sur la suppression de ressources dans Image Builder, consultezSupprimer les ressources EC2 Image Builder.

Gérer la configuration de l'infrastructure EC2 Image Builder

Vous pouvez utiliser les configurations d'infrastructure pour spécifier l'infrastructure Amazon EC2 utilisée par Image Builder pour créer et tester votre image EC2 Image Builder. Les paramètres d'infrastructure incluent :

- Types d'instances pour votre infrastructure de création et de test. Nous vous recommandons de spécifier plusieurs types d'instance, car cela permet à Image Builder de lancer une instance à partir d'un pool doté d'une capacité suffisante. Cela peut réduire vos échecs de compilation transitoires.
- Un profil d'instance qui fournit à vos instances de création et de test les autorisations nécessaires pour effectuer des activités de personnalisation. Par exemple, si vous avez un composant qui récupère des ressources depuis Amazon S3, le profil d'instance nécessite des autorisations pour accéder à ces fichiers. Le profil d'instance nécessite également un ensemble minimal d'autorisations pour qu'EC2 Image Builder puisse communiquer correctement avec l'instance. Pour plus d'informations, consultez Prérequis.
- Le VPC, le sous-réseau et les groupes de sécurité pour les instances de création et de test de votre pipeline.
- L'emplacement Amazon S3 où Image Builder stocke les journaux d'applications de votre compilation et de vos tests. Si vous configurez la journalisation, le profil d'instance spécifié dans la configuration de votre infrastructure doit disposer d'`s3:PutObject`autorisations pour le compartiment cible (`arn:aws:s3:::BucketName/*`).
- Une paire de clés Amazon EC2 qui vous permet de vous connecter à votre instance pour résoudre les problèmes si votre build échoue et que vous le configurez. `terminateInstanceOnFailure false`
- Rubrique SNS dans laquelle Image Builder envoie des notifications d'événements. Pour plus d'informations sur la façon dont Image Builder s'intègre à Amazon SNS, consultez. Intégration d'Amazon SNS dans Image Builder

> **Note**
>
> Si votre rubrique SNS est chiffrée, la clé qui chiffre cette rubrique doit résider dans le compte sur lequel le service Image Builder est exécuté. Image Builder ne peut pas envoyer de notifications aux rubriques SNS chiffrées à l'aide de clés provenant d'autres comptes.

Vous pouvez créer et gérer des configurations d'infrastructure à l'aide de la console Image Builder, via l'API Image Builder ou à l'aide de imagebuilder commandes dans le AWS CLI.

Table des matières

- EC2 Image Builder et points de terminaison VPC d'interface ()AWS PrivateLink

> **Tip**
>
> Lorsque vous disposez de plusieurs ressources du même type, le balisage vous aide à identifier une ressource spécifique en fonction des balises que vous lui avez attribuées. Pour plus d'informations sur le balisage de vos ressources à l'aide des commandes Image Builder dans le AWS CLI, consultez la Balisage des ressources section de ce guide.

Répertorier et afficher les détails de configuration de l'infrastructure

Cette section décrit les différentes manières de trouver des informations et d'afficher les détails de vos configurations d'infrastructure EC2 Image Builder.

Détails de configuration de l'infrastructure

- Répertorier les configurations d'infrastructure (AWS CLI)
- Obtenir les détails de configuration de l'infrastructure (AWS CLI)

Répertorier les configurations d'infrastructure (AWS CLI)

L'exemple suivant montre comment répertorier toutes les configurations de votre infrastructure à l'aide de la list-infrastructure-configurations commande du AWS CLI.

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

Obtenir les détails de configuration de l'infrastructure (AWS CLI)

L'exemple suivant montre comment utiliser la get-infrastructure-configuration commande du AWS CLI pour obtenir les détails d'une configuration d'infrastructure en spécifiant son 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
```

Créer une configuration d'infrastructure

Cette section décrit comment vous pouvez utiliser la console Image Builder ou imagebuilder les commandes du AWS CLI pour créer une configuration d'infrastructure,

Console

Pour créer une ressource de configuration d'infrastructure à partir de la console Image Builder, procédez comme suit :

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Dans le volet de navigation, choisissez Configuration de l'infrastructure.
3. Choisissez Créer une configuration d'infrastructure.
4. Dans la section Général, entrez les informations obligatoires suivantes :
   - Entrez le nom de votre ressource de configuration d'infrastructure.
   - Sélectionnez un rôle IAM que vous souhaitez associer au profil d'instance pour obtenir des autorisations de composant sur vos instances de génération et de test. Image Builder utilise ces autorisations pour télécharger et exécuter vos composants, télécharger des journaux et effectuer toutes les actions supplémentaires spécifiées par les composants de votre recette. CloudWatch
5. Dans le panneau AWS d'infrastructure, vous pouvez configurer les paramètres d'infrastructure restants disponibles. Entrez les informations obligatoires suivantes :
   - Type d'instance : vous pouvez spécifier un ou plusieurs types d'instance à utiliser pour cette version. Le service choisira l'un de ces types d'instances en fonction de sa disponibilité.
   - Rubrique SNS (facultatif) : sélectionnez une rubrique SNS pour recevoir des notifications et des alertes d'EC2 Image Builder.

   Si vous ne fournissez pas de valeurs pour les paramètres suivants, ils utilisent des valeurs par défaut spécifiques au service, le cas échéant.

   - VPC, sous-réseau et groupes de sécurité : Image Builder utilise votre VPC et votre sous-réseau par défaut. Pour plus d'informations sur la configuration des points de terminaison de l'interface VPC, consultez. EC2 Image Builder et points de terminaison VPC d'interface ()AWS PrivateLink

- Dans la section Paramètres de résolution des problèmes, vous pouvez configurer les valeurs suivantes :
  - Par défaut, la case Terminate instance en cas d'échec est cochée. Toutefois, lorsqu'une compilation échoue, vous pouvez vous connecter à l'instance EC2 pour résoudre les problèmes. Si vous souhaitez que votre instance continue de s'exécuter après un échec de compilation, décochez la case.
  - Paire de clés : si votre instance EC2 continue de fonctionner après un échec de compilation, vous pouvez créer une paire de clés ou utiliser une paire de clés existante pour vous connecter à l'instance et résoudre les problèmes.
  - Journaux : vous pouvez spécifier un compartiment S3 dans lequel Image Builder peut écrire des journaux d'applications pour vous aider à résoudre les problèmes liés à votre build et à vos tests. Si vous ne spécifiez pas de compartiment S3, Image Builder écrit les journaux de l'application sur l'instance.
- Dans la section Paramètres des métadonnées de l'instance, vous pouvez configurer les valeurs suivantes à appliquer aux instances EC2 utilisées par Image Builder pour créer et tester votre image :
  - Sélectionnez la version des métadonnées pour déterminer si EC2 nécessite un en-tête de jeton signé pour les demandes de récupération de métadonnées d'instance.
    - V1 et V2 (jeton facultatif) : valeur par défaut si vous ne sélectionnez rien.
    - V2 (jeton requis)

    > **Note**
    >
    > Nous vous recommandons de configurer toutes les instances EC2 lancées par Image Builder à partir d'un pipeline de manière à utiliser IMDSv2 afin que les demandes de récupération de métadonnées d'instance nécessitent un en-tête de jeton signé.

  - Limite de sauts de réponse du jeton de métadonnées : nombre de sauts réseau que le jeton de métadonnées peut effectuer. Nombre minimum de sauts : 1, nombre maximum de sauts : 64, avec un saut par défaut.

6. Dans la section Balises d'infrastructure (facultatif), vous pouvez attribuer des balises de métadonnées à l'instance Amazon EC2 lancée par Image Builder pendant le processus de création. Les balises sont saisies sous forme de paires clé-valeur.

7. Dans la section Tags (facultatif), vous pouvez attribuer des balises de métadonnées à la ressource de configuration d'infrastructure créée par Image Builder en sortie. Les balises sont saisies sous forme de paires clé-valeur.

AWS CLI

L'exemple suivant montre comment configurer l'infrastructure de votre image à l'aide de la create-infrastructure-configuration commande Image Builder dans le AWS CLI.

1. Créer un fichier JSON d'entrée CLI

   Cet exemple de configuration d'infrastructure spécifie deux types d'instances, `m5.large` et`m5.xlarge`. Nous vous recommandons de spécifier plusieurs types d'instance, car cela permet à Image Builder de lancer une instance à partir d'un pool doté d'une capacité suffisante. Cela peut réduire vos échecs de compilation transitoires.

   `instanceProfileName`Spécifie le profil d'instance qui fournit à l'instance les autorisations dont le profil a besoin pour effectuer des activités de personnalisation. Par exemple, si vous avez un composant qui récupère des ressources depuis Amazon S3, le profil d'instance nécessite des autorisations pour accéder à ces fichiers. Le profil d'instance nécessite également un ensemble minimal d'autorisations pour qu'EC2 Image Builder puisse communiquer correctement avec l'instance. Pour plus d'informations, consultez Prérequis.

   Utilisez un outil d'édition de fichier pour créer un fichier JSON avec les clés illustrées dans l'exemple suivant, auxquelles s'ajoutent des valeurs valides pour votre environnement. Cet exemple utilise un fichier nommé`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. Utilisez le fichier que vous avez créé comme entrée lorsque vous exécutez la commande suivante.

```
aws imagebuilder create-infrastructure-configuration --cli-input-json
 file://create-infrastructure-configuration.json
```

Mettre à jour une configuration d'infrastructure

Cette section explique comment utiliser la console Image Builder ou imagebuilder les commandes du AWS CLI pour mettre à jour une ressource de configuration d'infrastructure.

Console

Vous pouvez modifier les détails de configuration de l'infrastructure suivants depuis la console Image Builder :

- Description de la configuration de votre infrastructure.
- Rôle IAM à associer au profil d'instance.
- AWS infrastructure, y compris le type d'instance et une rubrique SNS pour les notifications.
- VPC, sous-réseau et groupes de sécurité.
- Paramètres de résolution des problèmes, notamment l'arrêt de l'instance en cas de défaillance, la paire de clés pour la connexion et un emplacement du compartiment S3 facultatif pour les journaux d'instance.

Pour mettre à jour une ressource de configuration d'infrastructure depuis la console Image Builder, procédez comme suit :

Choisissez une configuration d'infrastructure Image Builder existante

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Pour consulter la liste des ressources de configuration de l'infrastructure associées à votre compte, choisissez Configuration de l'infrastructure dans le volet de navigation.
3. Pour afficher les détails ou modifier une configuration d'infrastructure, cliquez sur le lien Nom de la configuration. Cela ouvre la vue détaillée de la configuration de l'infrastructure.

   > **Note**
   >
   > Vous pouvez également cocher la case à côté du nom de la configuration, puis choisir Afficher les détails.

4. Dans le coin supérieur droit du panneau des détails de l'infrastructure, choisissez Modifier.
5. Lorsque vous êtes prêt à enregistrer les mises à jour que vous avez apportées à la configuration de votre infrastructure, choisissez Enregistrer les modifications.

AWS CLI

L'exemple suivant montre comment mettre à jour la configuration de l'infrastructure de votre image à l'aide de la update-infrastructure-configuration commande Image Builder dans le AWS CLI.

1. Créer un fichier JSON d'entrée CLI

   Cet exemple de configuration d'infrastructure utilise les mêmes paramètres que l'exemple de création, sauf que nous avons mis à jour le `terminateInstanceOnFailure` paramètre sur`false`. Une fois la update-infrastructure-configuration commande exécutée, les pipelines qui utilisent cette configuration d'infrastructure mettent fin à la construction et testent les instances en cas d'échec de la construction.

   Utilisez un outil d'édition de fichier pour créer un fichier JSON avec les clés illustrées dans l'exemple suivant, auxquelles s'ajoutent des valeurs valides pour votre environnement. Cet exemple utilise un fichier nommé`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. Utilisez le fichier que vous avez créé comme entrée lorsque vous exécutez la commande suivante.

```
aws imagebuilder update-infrastructure-configuration --cli-input-json
 file://update-infrastructure-configuration.json
```

EC2 Image Builder et points de terminaison VPC d'interface ()AWS PrivateLink

Vous pouvez établir une connexion privée entre votre VPC et EC2 Image Builder en créant un point de terminaison VPC d'interface. Les points de terminaison de l'interface sont alimentés par AWS PrivateLinkune technologie qui vous permet d'accéder en privé aux API Image Builder sans passerelle Internet, appareil NAT, connexion VPN ou AWS Direct Connect connexion. Les instances de votre VPC n'ont pas besoin d'adresses IP publiques pour communiquer avec les API Image Builder. Le trafic entre votre VPC et Image Builder ne quitte pas le réseau Amazon.

Chaque point de terminaison d'interface est représenté par une ou plusieurs interfaces réseau Elastic dans vos sous-réseaux. Lorsque vous créez une nouvelle image, vous pouvez spécifier l'identifiant de sous-réseau VPC dans la configuration de votre infrastructure.

> **Note**
>
> Chaque service auquel vous accédez depuis un VPC possède son propre point de terminaison d'interface, avec sa propre politique de point de terminaison. Image Builder télécharge l'application de gestion des AWSTOE composants et accède aux ressources gérées à partir de compartiments S3 pour créer des images personnalisées. Pour accorder l'accès à ces compartiments, vous devez mettre à jour la politique du point de terminaison S3 afin de l'autoriser. Pour plus d'informations, consultez Politiques personnalisées pour l'accès au compartiment S3.

Pour plus d'informations sur les points de terminaison d'un VPC, consultez Points de terminaison de VPC d'interface (AWS PrivateLink) dans le Guide de l'utilisateur Amazon VPC.

Considérations relatives aux points de terminaison VPC Image Builder

Avant de configurer un point de terminaison VPC d'interface pour Image Builder, assurez-vous de consulter les propriétés et les limites du point de terminaison d'interface dans le guide de l'utilisateur Amazon VPC.

Image Builder permet d'appeler toutes ses actions d'API depuis votre VPC.

Création d'un point de terminaison VPC d'interface pour Image Builder

Pour créer un point de terminaison VPC pour le service Image Builder, vous pouvez utiliser la console Amazon VPC ou le (). AWS Command Line Interface AWS CLI Pour plus d'informations, consultez Création d'un point de terminaison d'interface dans le Guide de l'utilisateur Amazon VPC.

Créez un point de terminaison VPC pour Image Builder en utilisant le nom de service suivant :

- com.amazonaws.*region*.imagebuilder

Si vous activez le DNS privé pour le point de terminaison, vous pouvez envoyer des demandes d'API à Image Builder en utilisant son nom DNS par défaut pour la région, par exemple :`imagebuilder.us-east-1.amazonaws.com`. Pour rechercher le point de terminaison qui s'applique à votre région cible, consultez la section Points de terminaison et quotas d'EC2 Image Builder dans le. Référence générale d'Amazon Web Services

Pour plus d'informations, consultez Accès à un service via un point de terminaison d'interface dans le Guide de l'utilisateur Amazon VPC.

Création d'une politique de point de terminaison VPC pour Image Builder

Vous pouvez associer une politique de point de terminaison à votre point de terminaison VPC qui contrôle l'accès à Image Builder. La politique spécifie les informations suivantes :

- Le principal qui peut exécuter des actions.
- Les actions qui peuvent être effectuées.
- Les ressources sur lesquelles les actions peuvent être exécutées.

Si vous utilisez des composants gérés par Amazon dans votre recette, le point de terminaison VPC pour Image Builder doit autoriser l'accès à la bibliothèque de composants appartenant au service suivante :

`arn:aws:imagebuilder:`*`region`*`:aws:component/*`

> **Important**
>
> Lorsqu'une politique autre que celle par défaut est appliquée à un point de terminaison VPC d'interface pour EC2 Image Builder, certaines demandes d'API ayant échoué, telles que celles provenant de, peuvent `RequestLimitExceeded` ne pas être enregistrées AWS CloudTrail sur Amazon ou sur Amazon. CloudWatch

Pour plus d'informations, consultez Contrôle de l'accès aux services avec points de terminaison d'un VPC dans le Guide de l'utilisateur Amazon VPC.

Politiques personnalisées pour l'accès au compartiment S3

Image Builder utilise un compartiment S3 accessible au public pour stocker et accéder aux ressources gérées, telles que les composants. Il télécharge également l'application de gestion des AWSTOE composants à partir d'un compartiment S3 distinct. Si vous utilisez un point de terminaison VPC pour Amazon S3 dans votre environnement, vous devez vous assurer que votre politique de point de terminaison VPC S3 autorise Image Builder à accéder aux compartiments S3 suivants. Les noms des compartiments sont uniques par AWS région (*`région`*) et par environnement d'application (*`environnement`*). Image Builder AWSTOE prend en charge les environnements d'applications suivants : `prod`, `preprod`, et `beta`.

- Le bucket AWSTOE du gestionnaire de composants :

```
s3://ec2imagebuilder-toe-region-environment
```

Exemple : s3://ec2 imagebuilder-toe-us-west -2-prod/*

- Le bucket de ressources gérées par Image Builder :

```
s3://ec2imagebuilder-managed-resources-region-environment/components
```

Exemple : s3://ec2 imagebuilder-managed-resources-us -west-2-prod/components/*

Exemples de stratégie de point de terminaison de VPC

Cette section inclut des exemples de politiques de point de terminaison VPC personnalisées.

Politique générale des points de terminaison VPC pour les actions Image Builder

L'exemple de politique de point de terminaison suivant pour Image Builder refuse l'autorisation de supprimer des images et des composants d'Image Builder. L'exemple de politique accorde également l'autorisation d'effectuer toutes les autres actions d'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": "*",
```

```
    }]
}
```

Restreindre l'accès par organisation, autoriser l'accès aux composants gérés

L'exemple de politique de point de terminaison suivant montre comment restreindre l'accès aux identités et aux ressources appartenant à votre organisation et fournir un accès aux composants gérés par Amazon AWSTOE . Remplacez *la principal-org-idrégion* et *resource-org-id*par les valeurs de votre organisation.

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

```
}
```

Politique de point de terminaison VPC pour l'accès au compartiment Amazon S3

L'exemple de politique de point de terminaison S3 suivant montre comment fournir un accès aux compartiments S3 utilisés par Image Builder pour créer des images personnalisées. Remplacez *la région* et *l'environnement* par les valeurs de votre organisation. Ajoutez toute autre autorisation requise à la politique en fonction des exigences de votre application.

> **Note**
>
> Pour les images Linux, si vous ne spécifiez pas de données utilisateur dans votre recette d'image, Image Builder ajoute un script pour télécharger et installer l'agent Systems Manager sur les instances de compilation et de test de votre image. Pour télécharger l'agent, Image Builder accède au compartiment S3 de votre région de construction.
> Pour garantir qu'Image Builder puisse démarrer les instances de génération et de test, ajoutez la ressource supplémentaire suivante à votre politique de point de terminaison 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/
*"
      ]
    }
  ]
}
```

Gérer les paramètres de distribution d'EC2 Image Builder

Après avoir créé les paramètres de distribution avec Image Builder, vous pouvez les gérer à l'aide de la console Image Builder, de l'API Image Builder ou des imagebuilder commandes du AWS CLI. Les paramètres de distribution vous permettent d'effectuer les actions suivantes :

Diffusion d'AMI

- Spécifiez le nom et la description de votre AMI de sortie.
- Autorisez Comptes AWS d'autres organisations et unités d'organisation à lancer l'AMI depuis le compte du propriétaire. Le compte du propriétaire est facturé pour les frais associés à l'AMI.

  > **Note**
  >
  > Pour rendre une AMI publique, définissez les comptes autorisés pour l'autorisation de lancement sur `all`. Consultez les exemples de publication d'une AMI sur EC2. ModifyImageAttribute
- Créez une copie de l'AMI de sortie pour chacun des comptes, organisations et unités d'organisation cibles spécifiés dans la région de destination. Les comptes cibles, les organisations et les unités d'organisation sont propriétaires de leurs copies d'AMI et sont facturés pour tous les frais associés. Pour plus d'informations sur la distribution de votre AMI aux unités d'organisation AWS Organizations et aux unités d'organisation, voir Partager une AMI avec des organisations ou des unités d'organisation.
- Copiez l'AMI sur le compte du propriétaire dans un autre compte Régions AWS.
- Exportez des disques d'images de machine virtuelle vers Amazon Simple Storage Service (Amazon S3). Pour plus d'informations, consultez Création de paramètres de distribution pour les disques de machine virtuelle de sortie (AWS CLI).

Distribution d'images de conteneurs

- Spécifiez le référentiel ECR dans lequel Image Builder stocke l'image de sortie dans la région de distribution.

Vous pouvez utiliser vos paramètres de distribution de la manière suivante pour fournir des images aux régions, comptes AWS Organizations et unités organisationnelles (UO) cibles une seule fois, ou à chaque création de pipeline :

- Pour fournir automatiquement des images mises à jour à des régions, des comptes, des organisations et des unités d'organisation spécifiques, utilisez les paramètres de distribution avec un pipeline Image Builder qui s'exécute selon un calendrier.
- Pour créer une nouvelle image et la diffuser aux régions, comptes, organisations et unités d'organisation spécifiés, utilisez les paramètres de distribution avec un pipeline Image Builder que vous exécutez une fois depuis la console Image Builder, en utilisant Run pipeline dans le menu Actions.
- Pour créer une nouvelle image et la diffuser aux Régions, comptes, Organisations et UO spécifiés, utilisez les paramètres de distribution avec l'action d'API ou la commande Image Builder suivante dans le AWS CLI :
  - L'CreateImageaction dans l'API Image Builder.
  - La create-image commande contenue dans le AWS CLI.
- Exporter des disques d'image de machine virtuelle (VM) vers des compartiments S3 dans les régions cibles dans le cadre de votre processus de création d'image habituel.

> **Tip**
>
> Lorsque vous disposez de plusieurs ressources du même type, le balisage vous aide à identifier une ressource spécifique en fonction des balises que vous lui avez attribuées. Pour plus d'informations sur le balisage de vos ressources à l'aide des commandes Image Builder dans le AWS CLI, consultez la Balisage des ressources section de ce guide.

Cette rubrique explique comment répertorier, afficher et créer des paramètres de distribution.

Table des matières

Répertorier et afficher le détail des paramètres de distribution

Cette section décrit les différentes manières de trouver des informations et d'afficher les détails relatifs à vos paramètres de distribution EC2 Image Builder.

Détails des paramètres de distribution

- Configurations de distribution de listes (console)
- Afficher les détails de configuration de distribution (console)
- Répertorier les distributions (AWS CLI)
- Obtenir le détail de la configuration de distribution (AWS CLI)

Configurations de distribution de listes (console)

Pour consulter la liste des configurations de distribution créées sous votre compte dans la console Image Builder, procédez comme suit :

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Choisissez Paramètres de distribution dans le volet de navigation. Cela affiche la liste des configurations de distribution créées sous votre compte.
3. Pour afficher les détails ou créer une nouvelle configuration de distribution, cliquez sur le lien Nom de la configuration. Cela ouvre la vue détaillée des paramètres de distribution.

> **Note**
>
> Vous pouvez également cocher la case à côté du nom de la configuration, puis choisir Afficher les détails.

Afficher les détails de configuration de distribution (console)

Pour afficher les détails d'une configuration de distribution spécifique à l'aide de la console Image Builder, sélectionnez la configuration à examiner en suivant les étapes décrites dansConfigurations de distribution de listes (console).

Sur la page détaillée de la distribution, vous pouvez :

- Supprimez la configuration de distribution. Pour plus d'informations sur la suppression de ressources dans Image Builder, consultezSupprimer les ressources EC2 Image Builder.
- Modifiez les informations relatives à la distribution.

Répertorier les distributions (AWS CLI)

L'exemple suivant montre comment utiliser la list-distribution-configurations commande AWS CLI pour répertorier toutes vos distributions.

```
aws imagebuilder list-distribution-configurations
```

Obtenir le détail de la configuration de distribution (AWS CLI)

L'exemple suivant montre comment utiliser la get-distribution-configuration commande du AWS CLI pour obtenir les détails d'une configuration de distribution en spécifiant son 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
```

Création et mise à jour des configurations de distribution d'AMI

Cette section traite de la création et de la mise à jour des configurations de distribution pour une AMI Image Builder.

Table des matières

Création d'une configuration de distribution AMI (console)

Les configurations de distribution incluent le nom de l'AMI de sortie, les paramètres régionaux spécifiques pour le chiffrement, les autorisations de lancement Comptes AWS, les organisations et unités organisationnelles (UO) qui peuvent lancer l'AMI de sortie, ainsi que les configurations de licence.

Pour créer une nouvelle configuration de distribution d'AMI, procédez comme suit :

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Choisissez Paramètres de distribution dans le volet de navigation. Cela affiche la liste des configurations de distribution créées sous votre compte.
3. Choisissez Créer des paramètres de distribution en haut du panneau des paramètres de distribution.
4. Dans la section Type d'image, choisissez le type de sortie Amazon Machine Image (AMI).
5. Dans la section Général, entrez un nom pour votre configuration de distribution, ainsi qu'une description facultative.
6. Dans la section Paramètres régionaux, entrez les informations suivantes pour chaque région dans laquelle vous distribuez votre AMI :

    a. L'AMI est distribuée à la région actuelle (région 1), par défaut. La région 1 est la source de la distribution. Certains paramètres de la région 1 ne peuvent pas être modifiés. Pour toutes les régions que vous ajoutez, vous pouvez choisir une région dans la liste déroulante des régions.

    La clé Kms identifie celle AWS KMS key qui est utilisée pour chiffrer les volumes EBS de votre image dans la région cible. Il est important de noter que cela ne s'applique pas à l'AMI d'origine créée par le build sous votre compte dans la région source (région 1). Le chiffrement exécuté pendant la phase de distribution de la compilation concerne uniquement les images distribuées à d'autres comptes ou régions.

    Pour chiffrer les volumes EBS pour l'AMI créée dans la région source de votre compte, vous devez définir la clé KMS dans le mappage des périphériques par blocs de recettes d'image (stockage (volumes) dans la console).

    Image Builder copie l'AMI sur les comptes cibles que vous spécifiez pour la région.

> **Prérequis**
>
> Pour copier une image entre comptes, vous devez créer le `EC2ImageBuilderDistributionCrossAccountRole` rôle dans tous les comptes cibles des régions cibles et associer la politique Stratégie Ec2ImageBuilderCrossAccountDistributionAccess gérée au rôle.

Le nom de l'AMI de sortie est facultatif. Si vous fournissez un nom, le nom final de l'AMI de sortie inclut un horodatage indiquant la date de création de l'AMI. Si vous ne spécifiez aucun nom, Image Builder ajoute l'horodatage de construction au nom de la recette. Cela garantit des noms d'AMI uniques pour chaque build.

i. Grâce au partage d'AMI, vous pouvez accorder l'accès à des entités AWS principales spécifiques afin de lancer des instances depuis votre AMI. Si vous développez la section de partage d'AMI, vous pouvez saisir les informations suivantes :

   - Autorisations de lancement : sélectionnez Privé si vous souhaitez conserver la confidentialité de votre AMI et autoriser l'accès à des AWS principaux spécifiques pour lancer une instance depuis votre AMI privée. Sélectionnez Public si vous souhaitez rendre votre AMI publique. N'importe quel AWS principal peut lancer une instance depuis votre AMI publique.
   - Principaux : vous pouvez accorder l'accès aux types de AWS principaux suivants pour lancer des instances :
     - AWS compte — Accorder l'accès à un AWS compte spécifique
     - Unité organisationnelle (UO) : accordez l'accès à une UO et à toutes ses entités enfants. Les entités enfants incluent les unités d'organisation et AWS les comptes.
     - Organisation : accordez l'accès à votre AWS Organizations entité et à toutes ses entités enfants. Les entités enfants incluent les unités d'organisation et AWS les comptes.

       Sélectionnez d'abord le type principal. Entrez ensuite l'ID du AWS principal auquel vous souhaitez accorder l'accès dans le champ situé à droite de la liste déroulante. Vous pouvez saisir plusieurs identifiants de différents types.

ii. Vous pouvez développer la section Configuration des licences pour joindre les configurations de licence créées avec AWS License Manager à vos images Image

Builder. Les configurations de licence contiennent des règles de licence basées sur les termes de vos contrats d'entreprise. Image Builder inclut automatiquement les configurations de licence associées à votre AMI de base.

iii. Vous pouvez développer la section de configuration du modèle de lancement pour spécifier un modèle de lancement EC2 à utiliser pour lancer des instances à partir de l'AMI que vous créez.

Si vous utilisez un modèle de lancement EC2, vous pouvez demander à Image Builder de créer une nouvelle version de votre modèle de lancement qui inclut le dernier ID AMI une fois la compilation terminée. Pour mettre à jour le modèle de lancement, configurez les paramètres comme suit :

- Nom du modèle de lancement : sélectionnez le nom du modèle de lancement que vous souhaitez qu'Image Builder mette à jour.
- Définir la version par défaut : cochez cette case pour mettre à jour la version par défaut du modèle de lancement vers la nouvelle version.

Pour ajouter une autre configuration de modèle de lancement, choisissez Ajouter une configuration de modèle de lancement. Vous pouvez avoir jusqu'à cinq configurations de modèles de lancement par région.

b. Pour ajouter des paramètres de distribution pour une autre région, choisissez Ajouter une région.

7. Choisissez Créer des paramètres lorsque vous avez terminé.

Création de paramètres de distribution pour les AMI de sortie (AWS CLI)

Une configuration de distribution vous permet de spécifier le nom et la description de votre AMI de sortie, d'autoriser d'autres utilisateurs Comptes AWS à lancer l'AMI, de copier l'AMI sur d'autres comptes et de répliquer l'AMI dans d'autres AWS régions. Il vous permet également d'exporter l'AMI vers Amazon Simple Storage Service (Amazon S3) ou de configurer EC2 Fast Launch pour les AMI Windows de sortie. Pour rendre une AMI publique, définissez les comptes autorisés pour l'autorisation de lancement sur `all`. Consultez les exemples de publication d'une AMI sur EC2. ModifyImageAttribute

L'exemple suivant montre comment utiliser la create-distribution-configuration commande pour créer une nouvelle configuration de distribution pour votre AMI, à l'aide du AWS CLI.

1. Créer un fichier JSON d'entrée CLI

   Utilisez un outil d'édition de fichiers pour créer un fichier JSON avec les clés illustrées dans l'un des exemples suivants et des valeurs valides pour votre environnement. Ces exemples définissent quelles Comptes AWS unités organisationnelles (UO) sont autorisées à lancer l'AMI que vous distribuez dans les régions spécifiées. AWS Organizations Nommez le fichier`create-ami-distribution-configuration.json`, à utiliser à l'étape suivante :

   Accounts

   Cet exemple distribue une AMI à deux régions et indique celles Comptes AWS qui disposent d'autorisations de lancement dans chaque région.

```
{
    "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

Cet exemple distribue une AMI à la région source et spécifie les autorisations de lancement de l'organisation et de l'unité d'organisation.

```
{
  "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. Exécutez la commande suivante en utilisant le fichier que vous avez créé en entrée.

```
aws imagebuilder create-distribution-configuration --cli-input-json file://create-
ami-distribution-configuration.json
```

> **Note**
>
> - Vous devez inclure l'option `file://` au début du chemin du fichier JSON.
> - Le chemin d'accès du fichier JSON doit suivre la convention appropriée pour le système d'exploitation de base sur lequel vous exécutez la commande. En effet, Windows utilise la barre oblique inverse (\) pour faire référence au chemin du répertoire, et Linux utilise la barre oblique directe (/).

Pour des informations plus détaillées, reportez-vous create-distribution-configuration à la référence des AWS CLI commandes.

Mettre à jour les paramètres de distribution de l'AMI (console)

Vous pouvez modifier les paramètres de distribution de votre AMI à l'aide de la console Image Builder. Les paramètres de distribution mis à jour sont utilisés pour tous les déploiements de pipelines automatisés et manuels à l'avenir. Toutefois, les modifications que vous apportez ne s'appliquent pas aux ressources déjà distribuées par Image Builder. Par exemple, si vous avez distribué une AMI dans une région que vous supprimez ultérieurement de votre distribution, l'AMI déjà distribuée reste dans cette région jusqu'à ce que vous la supprimiez manuellement.

Mettre à jour la configuration de distribution de l'AMI

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Choisissez Paramètres de distribution dans le volet de navigation. Cela affiche la liste des configurations de distribution créées sous votre compte.
3. Pour afficher les détails ou mettre à jour une configuration de distribution, cliquez sur le lien Nom de la configuration. Cela ouvre la vue détaillée des paramètres de distribution.

> **Note**
>
> Vous pouvez également cocher la case à côté du nom de la configuration, puis choisir Afficher les détails.

4. Pour modifier la configuration de distribution, choisissez Modifier dans le coin supérieur droit de la section Détails de distribution. Certains champs sont verrouillés, tels que le nom de la

configuration de distribution et la région par défaut affichée sous la forme Région 1. Pour plus d'informations sur les paramètres de configuration de distribution, consultezCréation d'une configuration de distribution AMI (console).

5. Lorsque vous avez terminé, choisissez Save changes (Enregistrer les modifications).

Création de paramètres de distribution pour une AMI Windows avec EC2 Fast Launch activé ()AWS CLI

L'exemple suivant montre comment utiliser la create-distribution-configuration commande pour créer des paramètres de distribution dans lesquels EC2 Fast Launch est configuré pour votre AMI, à l'aide du AWS CLI.

1. Créer un fichier JSON d'entrée CLI

   Utilisez un outil d'édition de fichiers pour créer un fichier JSON avec des clés, comme indiqué dans l'exemple suivant, ainsi que des valeurs valides pour votre environnement.

   Cet exemple lance des instances pour toutes ses ressources cibles simultanément, car le nombre maximum de lancements parallèles est supérieur au nombre de ressources cibles. Ce fichier est nommé `ami-dist-config-win-fast-launch.json` dans l'exemple de commande présenté à l'étape suivante.

```
{
"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**
>
> Vous pouvez spécifier le `launchTemplateName` au lieu de `launchTemplateId` dans la `launchTemplate` section, mais vous ne pouvez pas spécifier à la fois le nom et l'identifiant.

2. Exécutez la commande suivante en utilisant le fichier que vous avez créé en entrée.

```
aws imagebuilder create-distribution-configuration --cli-input-json file://ami-dist-config-win-fast-launch.json
```

> **Note**
>
> - Vous devez inclure l'option `file://` au début du chemin du fichier JSON.
> - Le chemin d'accès du fichier JSON doit suivre la convention appropriée pour le système d'exploitation de base sur lequel vous exécutez la commande. En effet, Windows utilise la barre oblique inverse (\) pour faire référence au chemin du répertoire, et Linux utilise la barre oblique directe (/).

Pour des informations plus détaillées, reportez-vous create-distribution-configuration à la référence des AWS CLI commandes.

Création de paramètres de distribution pour les disques de machine virtuelle de sortie (AWS CLI)

L'exemple suivant montre comment utiliser la create-distribution-configuration commande pour créer des paramètres de distribution qui exporteront les disques d'images de machine virtuelle vers Amazon S3 à chaque création d'image.

1. Créer un fichier JSON d'entrée CLI

   Vous pouvez rationaliser la create-distribution-configuration commande que vous utilisez dans le AWS CLI. Pour ce faire, créez un fichier JSON contenant l'ensemble de la configuration d'exportation que vous souhaitez transmettre à la commande.

   > **Note**
   >
   > La convention de dénomination des valeurs de données dans le fichier JSON suit le modèle spécifié pour les paramètres de demande d'action de l'API Image Builder. Pour consulter les paramètres de demande de commande d'API, consultez la CreateDistributionConfiguration commande dans le manuel de référence de l'API EC2 Image Builder.
   > Pour fournir les valeurs de données sous forme de paramètres de ligne de commande, reportez-vous aux noms de paramètres spécifiés dans la référence de AWS CLI commande. À la create-distribution-configuration commande sous forme d'options.

   Voici un résumé des paramètres que nous avons spécifiés dans l'objet `s3ExportConfiguration` JSON pour cet exemple :

   - RoleName (chaîne, obligatoire) : nom du rôle qui accorde à VM Import/Export l'autorisation d'exporter des images vers votre compartiment S3.
   - diskImageFormat(chaîne, obligatoire) — Exportez l'image disque mise à jour vers l'un des formats pris en charge suivants :
     - Disque dur virtuel (VHD) : compatible avec les produits de virtualisation Citrix Xen et Microsoft Hyper-V.
     - Disque de machine virtuelle ESX (VMDK) optimisé pour les flux : compatible avec les versions 4, 5 et 6 de VMware ESX et VMware vSphere.
     - Raw — Format brut.

- S3bucket (chaîne, obligatoire) — Le compartiment S3 dans lequel stocker les images de disque de sortie pour votre machine virtuelle.

Enregistrez le fichier sous le nom `export-vm-disks.json`. Utilisez le nom du fichier dans la create-distribution-configuration commande.

```
{
    "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. Exécutez la commande suivante en utilisant le fichier que vous avez créé en entrée.

```
aws imagebuilder create-distribution-configuration --cli-input-json file://export-vm-disks.json
```

> **Note**
>
> - Vous devez inclure l'option `file://` au début du chemin du fichier JSON.
> - Le chemin d'accès du fichier JSON doit suivre la convention appropriée pour le système d'exploitation de base sur lequel vous exécutez la commande. En effet, Windows utilise la barre oblique inverse (\) pour faire référence au chemin du répertoire, et Linux utilise la barre oblique directe (/).

Pour des informations plus détaillées, reportez-vous create-distribution-configuration à la référence des AWS CLI commandes.

Mettre à jour les paramètres de distribution de l'AMI (AWS CLI)

L'exemple suivant montre comment utiliser la update-distribution-configuration commande pour mettre à jour les paramètres de distribution de votre AMI à l'aide du AWS CLI.

1. Créer un fichier JSON d'entrée CLI

   Utilisez votre outil d'édition de fichiers préféré pour créer un fichier JSON avec les clés illustrées dans l'exemple suivant, auxquelles s'ajoutent des valeurs valides pour votre environnement. Cet exemple utilise un fichier nomméupdate-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. Exécutez la commande suivante en utilisant le fichier que vous avez créé en entrée.

```
aws imagebuilder update-distribution-configuration --cli-input-json file://update-ami-distribution-configuration.json
```

> **Note**
>
> - Vous devez inclure l'option `file://` au début du chemin du fichier JSON.
> - Le chemin d'accès du fichier JSON doit suivre la convention appropriée pour le système d'exploitation de base sur lequel vous exécutez la commande. En effet, Windows utilise la barre oblique inverse (\) pour faire référence au chemin du répertoire, et Linux utilise la barre oblique directe (/).

Pour des informations plus détaillées, reportez-vous update-distribution-configuration à la référence des AWS CLI commandes. Pour mettre à jour les balises de votre ressource de configuration de distribution, consultez la Balisage des ressources section.

Création et mise à jour des paramètres de distribution pour les images de conteneurs

Cette section traite de la création et de la mise à jour des paramètres de distribution pour les images de conteneur Image Builder.

Table des matières

- Mettre à jour les paramètres de distribution pour votre image de conteneur (AWS CLI)

Création de paramètres de distribution pour les images de conteneur Image Builder (AWS CLI)

Une configuration de distribution vous permet de spécifier le nom et la description de votre image de conteneur de sortie et de répliquer l'image de conteneur dans d'autres AWS régions. Vous pouvez également appliquer des balises distinctes à la ressource de configuration de distribution et aux images de conteneur au sein de chaque région.

1. Créer un fichier JSON d'entrée CLI

   Utilisez votre outil d'édition de fichiers préféré pour créer un fichier JSON avec les clés illustrées dans l'exemple suivant, auxquelles s'ajoutent des valeurs valides pour votre environnement. Cet exemple utilise un fichier nommé`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. Exécutez la commande suivante en utilisant le fichier que vous avez créé en entrée.

```
aws imagebuilder create-distribution-configuration --cli-input-json file://create-container-distribution-configuration.json
```

> **Note**
>
> - Vous devez inclure l'option `file://` au début du chemin du fichier JSON.
> - Le chemin d'accès du fichier JSON doit suivre la convention appropriée pour le système d'exploitation de base sur lequel vous exécutez la commande. En effet, Windows utilise la barre oblique inverse (\) pour faire référence au chemin du répertoire, et Linux utilise la barre oblique directe (/).

Pour des informations plus détaillées, reportez-vous create-distribution-configuration à la référence des AWS CLI commandes.

Mettre à jour les paramètres de distribution pour votre image de conteneur (AWS CLI)

L'exemple suivant montre comment utiliser la update-distribution-configuration commande pour mettre à jour les paramètres de distribution de votre image de conteneur, à l'aide du AWS CLI. Vous pouvez également mettre à jour les balises pour les images des conteneurs dans chaque région.

1. Créer un fichier JSON d'entrée CLI

   Utilisez votre outil d'édition de fichiers préféré pour créer un fichier JSON avec les clés illustrées dans l'exemple suivant, auxquelles s'ajoutent des valeurs valides pour votre

environnement. Cet exemple utilise un fichier nomméupdate-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. Exécutez la commande suivante en utilisant le fichier que vous avez créé comme entrée :

```
aws imagebuilder update-distribution-configuration --cli-input-json file://update-
container-distribution-configuration.json
```

> **Note**
>
> - Vous devez inclure l'option `file://` au début du chemin du fichier JSON.
> - Le chemin d'accès du fichier JSON doit suivre la convention appropriée pour le système d'exploitation de base sur lequel vous exécutez la commande. En effet, Windows utilise la barre oblique inverse (\) pour faire référence au chemin du répertoire, et Linux utilise la barre oblique directe (/).

Pour des informations plus détaillées, reportez-vous update-distribution-configuration à la référence des AWS CLI commandes. Pour mettre à jour les balises de votre ressource de configuration de distribution, consultez la Balisage des ressources section.

Configurer la distribution d'AMI entre comptes avec Image Builder

Cette section décrit comment configurer les paramètres de distribution pour fournir une AMI Image Builder aux autres comptes que vous spécifiez.

Le compte de destination peut ensuite lancer ou modifier l'AMI, selon les besoins.

> **Note**
>
> AWS CLI les exemples de commandes présentés dans cette section supposent que vous avez déjà créé des fichiers JSON de recette d'image et de configuration d'infrastructure. Pour créer le fichier JSON pour une recette d'image, consultezCréez une recette imagée à l'aide du AWS CLI. Pour créer le fichier JSON pour une configuration d'infrastructure, consultezCréer une configuration d'infrastructure.

Prérequis

Pour que les comptes cibles puissent lancer correctement des instances à partir de votre image Image Builder, vous devez configurer les autorisations appropriées pour tous les comptes de destination dans toutes les régions.

Si vous chiffrez votre AMI à l'aide de AWS Key Management Service (AWS KMS), vous devez configurer un AWS KMS key pour votre compte qui sera utilisé pour chiffrer la nouvelle image.

Lorsque Image Builder effectue une distribution entre comptes pour les AMI chiffrées, l'image du compte source est déchiffrée et envoyée vers la région cible, où elle est rechiffrée à l'aide de la clé désignée pour cette région. Image Builder agissant pour le compte cible et utilise un rôle IAM que vous créez dans la région de destination, ce compte doit avoir accès aux clés dans les régions source et de destination.

Clés de chiffrement

Les conditions préalables suivantes sont requises si votre image est cryptée à l'aide AWS KMS de. Les conditions préalables à l'IAM sont abordées dans la section suivante.

Exigences relatives au compte source

- Créez une clé KMS dans votre compte dans toutes les régions où vous créez et distribuez votre AMI. Vous pouvez également utiliser une clé existante.
- Mettez à jour la politique clé pour toutes ces clés afin de permettre aux comptes de destination d'utiliser votre clé.

Exigences relatives au compte de destination

- Ajoutez une politique en ligne `EC2ImageBuilderDistributionCrossAccountRole` qui permet au rôle d'effectuer les actions requises pour distribuer une AMI chiffrée. Pour les étapes de configuration IAM, consultez la section sur les Politiques IAM conditions préalables.

Pour plus d'informations sur l'utilisation de l'accès entre comptes AWS KMS, voir Autoriser les utilisateurs d'autres comptes à utiliser une clé KMS dans le Guide du AWS Key Management Service développeur.

Spécifiez votre clé de chiffrement dans la recette de l'image, comme suit :

- Si vous utilisez la console Image Builder, choisissez votre clé de chiffrement dans la liste déroulante Encryption (alias KMS) de la section Stockage (volumes) de votre recette.
- Si vous utilisez l'action CreateImageRecipe API ou la create-image-recipe commande contenue dans le AWS CLI, configurez votre clé dans la `ebs` section `blockDeviceMappings` ci-dessous de votre entrée JSON.

  L'extrait de code JSON suivant indique les paramètres de chiffrement d'une recette d'image. En plus de fournir votre clé de chiffrement, vous devez également définir l'`encrypted`indicateur sur`true`.

```
{
 ...
 "blockDeviceMappings": [
 {
  "deviceName": "Example root volume",
  "ebs": {
   "deleteOnTermination": true,
   "encrypted": true,
   "iops": 100,
   "kmsKeyId": "image-owner-key-id",
   ...
  },
  ...
 }],
 ...
}
```

Politiques IAM

Pour configurer les autorisations de distribution entre comptes dans AWS Identity and Access Management (IAM), procédez comme suit :

1. Pour utiliser les AMI Image Builder réparties entre les comptes, le propriétaire du compte de destination doit créer un nouveau rôle IAM dans son compte appelé`EC2ImageBuilderDistributionCrossAccountRole`.
2. Ils doivent associer le Stratégie Ec2ImageBuilderCrossAccountDistributionAccess au rôle pour permettre la distribution entre comptes. Pour plus d'informations sur les politiques gérées, voir Politiques gérées et politiques intégrées dans le guide de l'AWS Identity and Access Management utilisateur.
3. Vérifiez que l'ID du compte source est ajouté à la politique de confiance attachée au rôle IAM du compte de destination. Pour plus d'informations sur les politiques de confiance, consultez la section Politiques basées sur les ressources dans le guide de l'AWS Identity and Access Management utilisateur.
4. Si l'AMI que vous distribuez est cryptée, le propriétaire du compte de destination doit ajouter la politique `EC2ImageBuilderDistributionCrossAccountRole` en ligne suivante à son compte afin de pouvoir utiliser vos clés KMS. La `Principal` section contient leur numéro de compte. Image Builder peut ainsi agir en leur nom lorsqu'il AWS KMS crypte et déchiffre l'AMI avec les clés appropriées pour chaque région.

```
{
 "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": "*"
  }
 ]
}
```

Pour plus d'informations sur les politiques intégrées, consultez la section Politiques intégrées dans le guide de l'AWS Identity and Access Management utilisateur.

5. Si vous utilisez `launchTemplateConfigurations` pour spécifier un modèle de lancement Amazon EC2, vous devez également ajouter la politique suivante `EC2ImageBuilderDistributionCrossAccountRole` à votre compte de destination.

```
{
    "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 pour la distribution entre comptes

La distribution d'images Image Builder sur plusieurs comptes comporte certaines limites :

- Le compte de destination est limité à 50 copies simultanées de l'AMI pour chaque région de destination.
- Si vous souhaitez copier une AMI de virtualisation paravirtuelle (PV) vers une autre région, la région de destination doit prendre en charge les AMI de virtualisation PV. Pour plus d'informations, consultez Types de virtualisations AMI Linux.
- Vous ne pouvez pas créer de copie non chiffrée d'un instantané chiffré. Si vous ne spécifiez pas de clé gérée par le client AWS Key Management Service (AWS KMS) pour le `KmsKeyId` paramètre, Image Builder utilise la clé par défaut pour Amazon Elastic Block Store (Amazon EBS). Pour de plus amples informations, veuillez consulter Chiffrement Amazon EBS dans le Guide de l'utilisateur Amazon Elastic Compute Cloud.

Pour plus d'informations, consultez le manuel CreateDistributionConfigurationde référence de l'API EC2 Image Builder.

Configuration de la distribution entre comptes pour une AMI Image Builder (console)

Cette section explique comment créer et configurer les paramètres de distribution pour la distribution entre comptes de vos AMI Image Builder à l'aide du AWS Management Console. La configuration de la distribution entre comptes nécessite des autorisations IAM spécifiques. Vous devez remplir le formulaire Prérequis correspondant à cette section avant de continuer.

Pour créer des paramètres de distribution dans la console Image Builder, procédez comme suit :

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Choisissez Paramètres de distribution dans le volet de navigation. Cela affiche la liste des paramètres de distribution créés sous votre compte.
3. En haut de la page des paramètres de distribution, choisissez Créer des paramètres de distribution. Cela vous amène à la page Créer des paramètres de distribution.
4. Dans la section Type d'image, choisissez Amazon Machine Image (AMI) comme type de sortie. Il s'agit du paramètre par défaut.
5. Dans la section Général, entrez le nom de la ressource de paramètres de distribution que vous souhaitez créer (obligatoire).
6. Dans la section Paramètres régionaux, entrez un identifiant de compte à 12 chiffres auquel vous souhaitez distribuer votre AMI dans les comptes Target de la région sélectionnée, puis appuyez sur Entrée. Cela vérifie le bon formatage, puis affiche l'ID de compte que vous avez saisi sous la case. Répétez le processus pour ajouter d'autres comptes.

   Pour supprimer un compte que vous avez saisi, choisissez le X affiché à droite de l'identifiant du compte.

   Entrez le nom de l'AMI de sortie pour chaque région.
7. Continuez à spécifier les paramètres supplémentaires dont vous avez besoin, puis choisissez Créer des paramètres pour créer votre nouvelle ressource de paramètres de distribution.

Configurer la distribution entre comptes pour une AMI Image Builder ()AWS CLI

Cette section décrit comment configurer un fichier de paramètres de distribution et utiliser la create-image commande contenue dans le AWS CLI pour créer et distribuer une AMI Image Builder sur plusieurs comptes.

La configuration de la distribution entre comptes nécessite des autorisations IAM spécifiques. Vous devez remplir la section Prérequis correspondant à cette section avant d'exécuter la create-image commande.

1. Configuration d'un fichier de paramètres de distribution

   Avant d'utiliser la create-image commande AWS CLI pour créer une AMI Image Builder distribuée à un autre compte, vous devez créer une structure `DistributionConfiguration` JSON qui spécifie les identifiants des comptes cibles dans les `AmiDistributionConfiguration` paramètres. Vous devez en spécifier au moins un `AmiDistributionConfiguration` dans la région source.

   Le fichier d'exemple suivant, nommé`create-distribution-configuration.json`, montre la configuration pour la distribution d'images entre comptes dans la région source.

   ```
   {
    "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. Création des paramètres de distribution

   Pour créer une ressource de paramètres de distribution Image Builder à l'aide de la create-distribution-configurationcommande AWS CLI, entrez les paramètres suivants dans la commande :

- Entrez le nom de la distribution dans le `--name` paramètre.
- Joignez le fichier JSON de configuration de distribution que vous avez créé dans le `--cli-input-json` paramètre.

```
aws imagebuilder create-distribution-configuration --name my distribution name --
cli-input-json file://create-distribution-configuration.json
```

> **Note**
>
> - Vous devez inclure l'option `file://` au début du chemin du fichier JSON.
> - Le chemin d'accès du fichier JSON doit suivre la convention appropriée pour le système d'exploitation de base sur lequel vous exécutez la commande. En effet, Windows utilise la barre oblique inverse (\) pour faire référence au chemin du répertoire, et Linux utilise la barre oblique directe (/).

Vous pouvez également fournir du JSON directement dans la commande, à l'aide du `--distributions` paramètre.

Configurer les paramètres de distribution de l'AMI pour utiliser un modèle de lancement Amazon EC2

Pour garantir une expérience de lancement cohérente pour votre AMI Image Builder dans les comptes cibles et les régions, vous pouvez spécifier un modèle de lancement Amazon EC2 dans vos paramètres de distribution, en utilisant. `launchTemplateConfigurations` Lorsqu'ils `launchTemplateConfigurations` sont présents pendant le processus de distribution, Image Builder crée une nouvelle version du modèle de lancement qui inclut tous les paramètres d'origine du modèle et le nouvel ID d'AMI issu de la compilation. Pour plus d'informations sur le lancement d'une instance EC2 à l'aide d'un modèle de lancement, consultez l'un des liens suivants, en fonction de votre système d'exploitation cible.

- Lancer une instance Linux à partir d'un modèle de lancement
- Lancer une instance Windows à partir d'un modèle de lancement

> **Note**
>
> Lorsque vous incluez un modèle de lancement pour activer Windows Fast Launch dans votre image, le modèle de lancement doit inclure la balise suivante afin qu'Image Builder puisse activer Windows Fast Launch en votre nom.
> `CreatedBy: EC2 Image Builder`

Ajouter un modèle de lancement Amazon EC2 aux paramètres de distribution de votre AMI (console)

Pour fournir un modèle de lancement avec votre AMI de sortie, procédez comme suit dans la console :

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Choisissez Paramètres de distribution dans le volet de navigation. Cela affiche la liste des paramètres de distribution créés sous votre compte.
3. En haut de la page des paramètres de distribution, choisissez Créer des paramètres de distribution. Cela ouvre la page Créer des paramètres de distribution.
4. Dans la section Type d'image, choisissez le type de sortie Amazon Machine Image (AMI). Il s'agit du paramètre par défaut.
5. Dans la section Général, entrez le nom de la ressource de paramètres de distribution que vous souhaitez créer (obligatoire).
6. Dans la section Paramètres régionaux, sélectionnez le nom d'un modèle de lancement EC2 dans la liste. Si votre compte ne contient aucun modèle de lancement, choisissez Créer un nouveau modèle de lancement, qui ouvre les modèles de lancement dans le tableau de bord EC2.

   Cochez la case Définir la version par défaut pour mettre à jour la version par défaut du modèle de lancement vers la nouvelle version créée par Image Builder avec votre AMI de sortie.

   Pour ajouter un autre modèle de lancement à la région sélectionnée, choisissez Ajouter une configuration de modèle de lancement.

   Pour supprimer un modèle de lancement, choisissez Supprimer.
7. Continuez à spécifier les paramètres supplémentaires dont vous avez besoin, puis choisissez Créer des paramètres pour créer votre nouvelle ressource de paramètres de distribution.

Ajoutez un modèle de lancement Amazon EC2 aux paramètres de distribution de votre AMI ()AWS CLI

Cette section décrit comment configurer un fichier de paramètres de distribution avec un modèle de lancement et utiliser la create-image commande contenue dans le AWS CLI pour créer et distribuer une AMI Image Builder et une nouvelle version du modèle de lancement qui l'utilise.

1. Configuration d'un fichier de paramètres de distribution

   Avant de créer une AMI Image Builder avec un modèle de lancement AWS CLI, vous devez créer une structure JSON de configuration de distribution qui spécifie les `launchTemplateConfigurations` paramètres. Vous devez spécifier au moins une `launchTemplateConfigurations` entrée dans la région source.

   Le fichier d'exemple suivant, nommé`create-distribution-config-launch-template.json`, montre quelques scénarios possibles pour la configuration du modèle de lancement dans la région source.

```
{
    "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. Création des paramètres de distribution

   Pour créer une ressource de paramètres de distribution Image Builder à l'aide de la create-distribution-configurationcommande AWS CLI, entrez les paramètres suivants dans la commande :

   - Entrez le nom de la distribution dans le `--name` paramètre.
   - Joignez le fichier JSON de configuration de distribution que vous avez créé dans le `--cli-input-json` paramètre.

```
aws imagebuilder create-distribution-configuration --name my distribution name--
cli-input-json file://create-distribution-config-launch-template.json
```

> **Note**
>
> - Vous devez inclure l'option `file://` au début du chemin du fichier JSON.
> - Le chemin d'accès du fichier JSON doit suivre la convention appropriée pour le système d'exploitation de base sur lequel vous exécutez la commande. En effet, Windows utilise la barre oblique inverse (\) pour faire référence au chemin du répertoire, et Linux utilise la barre oblique directe (/).

Vous pouvez également fournir du JSON directement dans la commande, à l'aide du `--distributions` paramètre.

Gérez les politiques de cycle de vie des images EC2 Image Builder

Lorsque vous créez des images personnalisées, il est important de prévoir de retirer ces images avant qu'elles ne deviennent obsolètes. Les pipelines Image Builder peuvent appliquer automatiquement des mises à jour et des correctifs de sécurité. Cependant, chaque build crée une nouvelle version de l'image et de toutes les ressources associées qu'elle distribue. Les versions

antérieures restent dans votre compte jusqu'à ce que vous les supprimiez manuellement ou que vous créiez un script pour effectuer cette tâche.

Grâce aux politiques de gestion du cycle de vie d'Image Builder, vous pouvez automatiser le processus de dépréciation, de désactivation et de suppression des images obsolètes et des ressources associées. Les ressources associées peuvent inclure des images de sortie que vous avez distribuées à d'autres Comptes AWS organisations et unités organisationnelles (UO) Régions AWS. Vous définissez les règles indiquant comment et quand effectuer chaque étape du processus de cycle de vie, ainsi que les étapes à inclure dans votre politique.

Avantages de la gestion automatisée du cycle de vie

Les avantages globaux de la gestion automatisée du cycle de vie sont les suivants :

- Simplifie la gestion du cycle de vie de vos images personnalisées grâce à une méthode automatisée de retrait des images et des ressources associées.
- Aide à prévenir les risques de conformité liés à l'utilisation d'images obsolètes pour lancer de nouvelles instances.
- Maintient les inventaires d'images à jour en supprimant les images périmées.
- Peut réduire les coûts de stockage et de transfert de données en supprimant éventuellement les ressources associées aux images supprimées.

Réalisez des économies

L'utilisation d'EC2 Image Builder pour créer des images d'AMI ou de conteneur personnalisées est gratuite. Cependant, la tarification standard s'applique aux autres services utilisés dans le processus. Lorsque vous supprimez des images inutilisées ou périmées ainsi que les ressources associées Compte AWS, vous pouvez réaliser des économies de temps et d'argent des manières suivantes :

- Réduisez le temps nécessaire pour appliquer des correctifs aux images existantes lorsque vous ne corrigez pas également les images inutilisées ou obsolètes.
- Pour les ressources d'image AMI que vous supprimez, vous pouvez également choisir de supprimer les AMI distribuées et leurs instantanés associés. Cette approche permet d'économiser sur le coût de stockage des instantanés.
- Pour les ressources d'image de conteneur que vous supprimez, vous pouvez choisir de supprimer les ressources sous-jacentes. Cette approche permet de réduire les coûts de stockage Amazon ECR et les taux de transfert de données pour vos Docker images stockées dans des référentiels ECR.

> **Note**
>
> Image Builder ne peut pas évaluer l'impact potentiel de toutes les dépendances possibles en aval, telles que les groupes Auto Scaling ou les modèles de lancement. Vous devez tenir compte des dépendances en aval pour vos images lorsque vous configurez des actions de stratégie.

Table des matières

Conditions préalables à la gestion du cycle de vie pour les images EC2 Image Builder

Avant de pouvoir définir les politiques et règles de gestion du cycle de vie d'EC2 Image Builder pour vos ressources d'image, vous devez remplir les conditions préalables suivantes.

- Créez un rôle IAM qui autorise Image Builder à exécuter des politiques de cycle de vie. Pour créer ce rôle, consultez Création d'un rôle IAM pour la gestion du cycle de vie d'Image Builder.
- Créez un rôle IAM dans le compte de destination pour les ressources associées qui ont été distribuées entre les comptes. Le rôle autorise Image Builder à effectuer des actions de cycle de vie dans le compte de destination pour les ressources associées. Pour créer ce rôle, consultez Création d'un rôle IAM pour la gestion du cycle de vie entre comptes Image Builder.

> **Note**
>
> Cette condition préalable ne s'applique pas si vous avez accordé des autorisations de lancement pour une AMI de sortie. Avec les autorisations de lancement, le compte avec lequel vous avez partagé possède les instances lancées à partir de l'AMI partagée, mais toutes les ressources de l'AMI restent dans votre compte.

- Pour les images de conteneur, vous devez ajouter la balise suivante à vos référentiels ECR pour autoriser Image Builder à exécuter des actions de cycle de vie sur les images de conteneur stockées dans le référentiel :. `LifecycleExecutionAccess: EC2 Image Builder`

Création d'un rôle IAM pour la gestion du cycle de vie d'Image Builder

Pour autoriser Image Builder à exécuter des politiques de cycle de vie, vous devez d'abord créer le rôle IAM qu'il utilise pour effectuer des actions relatives au cycle de vie. Procédez comme suit pour créer le rôle de service qui accorde l'autorisation.

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Choisissez Rôles dans le panneau de navigation.
3. Sélectionnez Créer un rôle. Cela ouvre la première étape du processus : sélectionnez une entité de confiance pour créer votre rôle.
4. Sélectionnez l'option Politique de confiance personnalisée pour le type d'entité fiable.
5. Copiez la politique de confiance JSON suivante et collez-la dans la zone de texte de la politique de confiance personnalisée, en remplaçant le texte d'exemple. Cette politique de confiance permet à Image Builder d'assumer le rôle que vous avez créé pour exécuter des actions du cycle de vie.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sts:AssumeRole"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "imagebuilder.amazonaws.com"
                ]
            }
        }
    ]
}
```

6. Sélectionnez la politique gérée suivante dans la liste : EC2ImageBuilderLifecycleExecutionPolicy, puis choisissez Next. Cela ouvre la page Nom, révision et création.

> **Tip**
>
> Filtrez `image` pour rationaliser les résultats.

7. Entrez un nom de rôle.
8. Après avoir vérifié vos paramètres, choisissez Créer un rôle.

Création d'un rôle IAM pour la gestion du cycle de vie entre comptes Image Builder

Pour autoriser Image Builder à effectuer des actions de cycle de vie dans les comptes de destination pour les ressources associées, vous devez d'abord créer le rôle IAM qu'il utilise pour effectuer des actions de cycle de vie sur ces comptes. Vous devez créer le rôle dans le compte de destination.

Procédez comme suit pour créer le rôle de service qui accorde l'autorisation dans le compte de destination.

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Choisissez Rôles dans le panneau de navigation.
3. Sélectionnez Créer un rôle. Cela ouvre la première étape du processus : sélectionnez une entité de confiance pour créer votre rôle.
4. Sélectionnez l'option Politique de confiance personnalisée pour le type d'entité fiable.
5. Copiez la politique de confiance JSON suivante et collez-la dans la zone de texte de la politique de confiance personnalisée, en remplaçant le texte d'exemple. Cette politique de confiance permet à Image Builder d'assumer le rôle que vous avez créé pour exécuter des actions du cycle de vie.

> **Note**
>
> Lorsque Image Builder utilise ce rôle dans le compte de destination pour agir sur les ressources associées qui ont été distribuées entre les comptes, il agit pour le compte du propriétaire du compte de destination. Le compte Compte AWS que vous configurez

`aws:SourceAccount` dans la politique de confiance est le compte sur lequel Image Builder a distribué ces ressources.

```
{
    "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. Sélectionnez la politique gérée suivante dans la liste : EC2ImageBuilderLifecycleExecutionPolicy, puis choisissez Next. Cela ouvre la page Nom, révision et création.

> **Tip**
>
> Filtrez `image` pour rationaliser les résultats.

7. Entrez `Ec2ImageBuilderCrossAccountLifecycleAccess` comme nom du rôle.

> **Important**
>
> `Ec2ImageBuilderCrossAccountLifecycleAccess`doit être le nom de ce rôle.

8. Après avoir vérifié vos paramètres, choisissez Créer un rôle.

Politiques de gestion du cycle de vie pour les ressources d'image EC2 Image Builder

Grâce aux politiques de cycle de vie des images, vous pouvez définir votre stratégie de gestion des ressources afin de retirer les images obsolètes et les ressources associées grâce à un processus de dépréciation, de désactivation et de suppression des images obsolètes et des ressources associées. Cette section explique comment répertorier les politiques, afficher les détails des politiques et créer de nouvelles politiques pour les AMI et les images de conteneur.

Table des matières

Répertorier les politiques de gestion du cycle de vie pour les ressources d'image Image Builder

Vous pouvez obtenir une liste des politiques de gestion du cycle de vie de vos images qui inclut des colonnes détaillées clés sur la page de liste des politiques de cycle de vie du AWS Management Console ou avec des commandes ou des actions dans l'API Image Builder, les SDK ou AWS CLI.

Vous pouvez utiliser l'une des méthodes suivantes pour répertorier les ressources relatives à la politique de cycle de vie des images Image Builder dans votre Compte AWS. Pour l'action de l'API, voir le ListLifecyclePoliciesmanuel de référence de l'API EC2 Image Builder. Pour la demande de SDK associée, reportez-vous au lien Voir aussi sur la même page.

AWS Management Console

Les informations suivantes sont affichées dans la console pour vos politiques existantes. Vous pouvez sélectionner n'importe quelle colonne pour modifier l'ordre de tri de vos résultats. La liste des politiques est initialement triée par nom de stratégie. Le nom de colonne correspondant à l'ordre de tri actuel est en gras.

Si vous avez plusieurs pages de résultats, les flèches de pagination situées dans le coin supérieur droit du panneau deviennent actives. Vous pouvez filtrer les résultats par nom de politique, statut

de la politique, type d'image de sortie et ARN de la ressource d'image à l'aide de la barre de recherche.

- Nom de la stratégie : nom de la stratégie.
- État de la politique : indique si la stratégie est active ou inactive.
- Type : type d'image de sortie distribuée par Image Builder lorsque vous créez une nouvelle version d'image (AMI ou image de conteneur).
- Date de dernière exécution : dernière exécution de la politique de cycle de vie.
- Date de création : date de création de la politique de cycle de vie.
- ARN — Le nom de ressource Amazon (ARN) de la ressource de politique de cycle de vie.

Pour répertorier les politiques de cycle de vie dans le AWS Management Console, procédez comme suit :

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Sélectionnez Lifecycle policies dans le volet de navigation. Cela affiche une liste des politiques de cycle de vie des images de votre compte.

Actions disponibles

Vous pouvez également effectuer les actions suivantes pour votre politique de cycle de vie à partir de la page de liste des politiques de cycle de vie.

Pour créer une nouvelle politique de cycle de vie des images, choisissez Créer une politique de cycle de vie. Pour plus d'informations sur la création d'une politique, consultezCréez des politiques de cycle de vie.

Pour toutes les actions suivantes, vous devez d'abord sélectionner la politique. Pour sélectionner une politique, vous pouvez cocher la case à côté du nom de la politique.

- Pour activer ou désactiver la politique, sélectionnez Désactiver la politique ou Activer la politique dans le menu Actions.
- Pour modifier la politique, sélectionnez Modifier la politique dans le menu Actions.
- Pour supprimer une politique, sélectionnez Supprimer la politique dans le menu Actions.
- Pour créer une nouvelle politique qui utilise la stratégie que vous avez sélectionnée pour les paramètres de base, sélectionnez Cloner la politique dans le menu Actions.

AWS CLI

L'exemple de commande suivant montre comment utiliser le pour AWS CLI répertorier les politiques de cycle de vie des images pour une image spécifique Région AWS. Pour plus d'informations sur les paramètres et les options que vous pouvez utiliser avec cette commande, consultez la list-lifecycle-policiescommande dans le manuel de référence des AWS CLI commandes.

Exemple :

```
aws imagebuilder list-lifecycle-policies \
--region us-west-1
```

Sortie :

```
{
    "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**
>
> Pour utiliser votre valeur par défaut Région AWS, exécutez cette commande sans le `--region` paramètre.

Afficher les détails des politiques de cycle de vie

La page détaillée de la politique de cycle de vie de la console Image Builder inclut une section récapitulative, avec des informations supplémentaires regroupées sous forme d'onglets. L'en-tête de page est le nom de la politique.

Sur la page des détails de la politique de cycle de vie de la console Image Builder, vous pouvez consulter les détails d'une politique de cycle de vie spécifique. Vous pouvez également utiliser des commandes ou des actions avec l'API Image Builder, les SDK ou AWS CLI pour obtenir des informations sur les politiques.

Table des matières

Afficher les détails de la politique de cycle de vie dans la console Image Builder

La page détaillée de l'image dans la console Image Builder inclut une section récapitulative, avec des informations supplémentaires regroupées sous forme d'onglets. L'en-tête de page correspond au nom et à la version de compilation de la recette qui a créé l'image.

Sections et onglets détaillés de la console

- Section récapitulative
- Onglet Règles
- Onglet Étendue

- RunLog onglet

Section récapitulative

La section récapitulative couvre toute la largeur de la page et inclut les détails suivants. Ces informations sont toujours affichées.

État de la politique

Si la politique est active ou inactive.

Type

Type d'image de sortie distribuée par Image Builder lorsque vous créez une nouvelle version d'image (AMI ou image de conteneur).

Date de création

Horodatage depuis la création de la politique de cycle de vie.

Date de modification

Dernière mise à jour de la politique de cycle de vie.

Date de la dernière diffusion

La dernière fois que la politique de cycle de vie a été exécutée.

Rôle IAM

Rôle IAM utilisé par Image Builder pour effectuer des actions relatives au cycle de vie.

ARN

Le nom de ressource Amazon (ARN) de la ressource de politique de cycle de vie.

Description

Description de la politique de cycle de vie, si elle a été saisie.

Onglet Règles

L'onglet Règles affiche les règles de cycle de vie que vous avez configurées pour la politique que vous consultez. L'onglet inclut les informations suivantes :

- Nom : nom de la règle. Ces noms sont statiques, en fonction des actions de politique que vous pouvez configurer.
  - `Deprecation rule`
  - `Disable rule`
  - `Deletion rule`
- Règle : brève description de l'action configurée pour la règle.
- Conditions de règle : répertorie la configuration pour la gestion des ressources associées, les exceptions à la règle et les paramètres de rétention, le cas échéant.

Pour plus d'informations sur la configuration des règles, consultezComment fonctionnent les règles du cycle de vie.

Onglet Étendue

L'onglet Étendue affiche les critères de sélection des ressources configurés pour la politique que vous consultez. L'onglet inclut les informations suivantes :

- Filtre : ***`type de filtre`*** : type de filtre que vous avez utilisé pour définir le champ d'application. Le type de filtre peut être l'un des suivants :
  - `recipes`— Les recettes utilisées pour créer les images auxquelles s'applique la politique de cycle de vie.
  - `tags`— Ensemble de balises qu'Image Builder utilise pour sélectionner les ressources d'image auxquelles s'applique la politique de cycle de vie.
- Une barre de recherche : vous pouvez filtrer la liste par nom pour rationaliser les résultats qui s'affichent dans l'onglet.
- Nom : chaque ligne contient un nom ou une étiquette que vous avez configuré pour les critères de filtre.
- Version : si vous avez configuré un filtre de recette, Image Builder affiche la version de la recette.

RunLog onglet

Chaque fois que vous exécutez la politique pour vos ressources configurées, Image Builder enregistre les détails de l'exécution. Chaque ligne du tableau représente une instance d'exécution unique. L'onglet inclut les informations suivantes :

- ID d'exécution — Identifie l'instance d'exécution de la politique de cycle de vie.

- État d'exécution : état d'exécution qui indique si l'action politique est en cours d'exécution, s'est exécutée avec succès, a échoué ou a été annulée.
- Ressource affectée : indique si l'instance d'exécution a identifié des ressources d'image pour les actions du cycle de vie.
- Date de début : horodatage du démarrage de l'instance d'exécution.
- Date de fin : horodatage de la fin de l'instance d'exécution.

Créez des politiques de cycle de vie

Lorsque vous créez une nouvelle politique de cycle de vie EC2 Image Builder, la configuration dépend du type d'image auquel la politique est destinée. L'action de l'API pour créer une politique de cycle de vie pour les ressources d'image AMI et les ressources d'image de conteneur est la même (CreateLifecyclePolicy). Cependant, la configuration des ressources d'image et des ressources associées est différente. Cette section explique comment créer des politiques de gestion du cycle de vie pour les deux.

> **Note**
>
> Avant de créer une politique de cycle de vie, assurez-vous que vous les avez toutes respectéesPrérequis.

Créez des politiques de gestion du cycle de vie pour les ressources d'image AMI Image Builder

Vous pouvez utiliser l'une des méthodes suivantes pour créer une politique de cycle de vie des images AMI avec le AWS Management Console ou AWS CLI. Vous pouvez également utiliser l'action CreateLifecyclePolicyAPI. Pour la demande de SDK associée, vous pouvez vous référer au lien Voir aussi correspondant à cette commande dans le manuel EC2 Image Builder API Reference.

AWS Management Console

Pour créer une politique de cycle de vie pour les ressources d'images AMI dans le AWS Management Console, procédez comme suit :

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Choisissez Lifecycle policies dans le volet de navigation.
3. Choisissez Créer une politique de cycle de vie.

4. Configurez les paramètres de stratégie décrits dans les procédures suivantes.
5. Pour créer la politique de cycle de vie après avoir configuré les paramètres, choisissez Create policy.

Configurez les paramètres généraux de votre politique.

1. Sélectionnez l'option AMI dans Type de stratégie.
2. Entrez le nom de la politique.
3. Entrez éventuellement une description pour votre politique de cycle de vie.
4. Par défaut, l'option Activer est activée. Le paramètre par défaut active la politique de cycle de vie et l'ajoute immédiatement au calendrier. Pour créer une politique initialement désactivée, vous pouvez désactiver l'option Activer.
5. Sélectionnez le rôle IAM que vous avez créé pour les autorisations relatives aux politiques de cycle de vie. Si vous n'avez pas encore créé ce rôle, consultez Prérequis pour plus d'informations.

Configurez le champ d'application de la règle pour votre politique.

Cette section configure la sélection des ressources pour votre politique de cycle de vie, en fonction du type de filtre que vous utilisez.

1. Type de filtre : recettes — Pour appliquer des règles de cycle de vie aux ressources d'images en fonction de la recette qui les a créées, sélectionnez jusqu'à 50 versions de recette pour la politique.
2. Type de filtre : balises — Pour appliquer des règles de cycle de vie aux ressources d'images en fonction des balises de ressources, entrez une liste de 50 paires clé-valeur au maximum auxquelles la politique doit correspondre.

Activez une ou plusieurs des règles de cycle de vie suivantes pour les appliquer aux ressources sélectionnées par la politique de cycle de vie. Si une ressource correspond à plusieurs règles de cycle de vie lors de l'exécution de la politique, Image Builder exécute les actions de règle dans l'ordre suivant : 1) Déprécier, 2) Désactiver, 3) Supprimer.

Règle obsolète

Définit le statut de la ressource d'image Image Builder sur`Deprecated`. Les pipelines Image Builder fonctionnent toujours pour les images obsolètes. Vous pouvez éventuellement définir le délai d'obsolescence des AMI associées sans affecter votre capacité à lancer de nouvelles instances.

- Nombre d'unités — Spécifiez la valeur entière correspondant à la période qui doit s'écouler après la création d'une ressource d'image avant qu'elle ne soit marquée comme`Deprecated`.
- Unité — Sélectionnez la plage de temps à utiliser. La plage peut être `DaysWeeks`,`Months`, ou`Years`.
- Déprécier les AMI : cochez la case pour attribuer une date d'obsolescence aux AMI Amazon EC2 associées. Les AMI restent disponibles et vous pouvez toujours lancer de nouvelles instances à partir de celles-ci.

Désactiver la règle

Définit le statut de la ressource d'image Image Builder sur`Disabled`. Cela empêche les pipelines Image Builder de s'exécuter pour cette image. Vous pouvez éventuellement désactiver l'AMI associée pour empêcher le lancement de nouvelles instances.

- Nombre d'unités — Spécifiez la valeur entière correspondant à la période qui doit s'écouler après la création d'une ressource d'image avant qu'elle ne soit marquée comme`Disabled`.
- Unité — Sélectionnez la plage de temps à utiliser. La plage peut être `DaysWeeks`,`Months`, ou`Years`.
- Désactiver les AMI : cochez la case pour désactiver les AMI Amazon EC2 associées. Vous ne pouvez plus utiliser les AMI ni lancer de nouvelles instances à partir de celles-ci.

Supprimer la règle

Supprime les ressources d'image par âge ou par nombre. Vous définissez le seuil qui répond à vos besoins. Lorsqu'une ressource d'image Image Builder dépasse le seuil, elle est supprimée. Vous pouvez éventuellement désenregistrer les AMI associées ou supprimer les instantanés de ces AMI. Vous pouvez également spécifier des balises pour les ressources que vous souhaitez conserver au-delà du seuil.

Lorsque vous configurez la règle de suppression par âge, Image Builder supprime la ressource d'image après une période définie. Supprimez, par exemple, les ressources d'images au bout de 6 mois. Lorsque vous configurez par décompte, Image Builder conserve le nombre d'images le

plus récent que vous spécifiez, ou le plus proche possible de ce nombre, et supprime les versions antérieures.

- Par âge
  - Nombre d'unités — Spécifiez la valeur entière correspondant à la période qui doit s'écouler après la création d'une ressource d'image avant sa suppression.
  - Unité — Sélectionnez la plage de temps à utiliser. La plage peut être `Days`,`Weeks`,`Months`, ou`Years`.
  - Conserver au moins une image par recette : cochez la case pour conserver la dernière ressource d'image disponible pour chaque version de recette concernée par cette règle.

  Par décompte
  - Nombre d'images — Spécifiez la valeur entière du nombre de ressources d'images récentes à conserver pour chaque version de recette.
- Désenregistrer les AMI : cochez la case pour désenregistrer les AMI Amazon EC2 associées. Vous ne pouvez plus utiliser les AMI ni lancer de nouvelles instances à partir de celles-ci.
- Conserver les images, les AMI et les instantanés avec les balises associées : cochez la case pour saisir une liste de balises pour les ressources d'images que vous souhaitez conserver. Les balises s'appliquent aux ressources d'image et aux AMI Amazon EC2. Vous pouvez saisir jusqu'à 50 paires clé-valeur.

Les balises (facultatif)

Ajoutez des balises à votre politique de cycle de vie.

AWS CLI

Pour créer une nouvelle politique de cycle de vie d'Image Builder, vous pouvez utiliser la create-lifecycle-policy commande du. AWS CLI

Créez des politiques de gestion du cycle de vie pour les ressources d'images du conteneur Image Builder

Vous pouvez utiliser l'une des méthodes suivantes pour créer une politique de cycle de vie des images de conteneur avec le AWS Management Console ou AWS CLI. Vous pouvez également utiliser l'action CreateLifecyclePolicyAPI. Pour la demande de SDK associée, vous pouvez vous

référer au lien Voir aussi correspondant à cette commande dans le manuel EC2 Image Builder API Reference.

AWS Management Console

Pour créer une politique de cycle de vie pour les ressources d'images de conteneurs dans le AWS Management Console, procédez comme suit :

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Choisissez Lifecycle policies dans le volet de navigation.
3. Choisissez Créer une politique de cycle de vie.
4. Configurez les paramètres de stratégie décrits dans les procédures suivantes.
5. Pour créer la politique de cycle de vie après avoir configuré les paramètres, choisissez Create policy.

Configuration des politiques : paramètres généraux

Configurez les paramètres généraux de votre politique.

1. Sélectionnez l'option AMI dans Type de stratégie.
2. Entrez le nom de la politique.
3. Entrez éventuellement une description pour votre politique de cycle de vie.
4. Par défaut, l'option Activer est activée. Le paramètre par défaut active la politique de cycle de vie et l'ajoute immédiatement au calendrier. Pour créer une politique initialement désactivée, vous pouvez désactiver l'option Activer.
5. Sélectionnez le rôle IAM que vous avez créé pour les autorisations relatives aux politiques de cycle de vie. Si vous n'avez pas encore créé ce rôle, consultez Prérequis pour plus d'informations.

Configurez le champ d'application de la règle pour votre politique.

Cette section configure la sélection des ressources pour votre politique de cycle de vie, en fonction du type de filtre que vous utilisez.

1. Type de filtre : recettes — Pour appliquer des règles de cycle de vie aux ressources d'images en fonction de la recette qui les a créées, sélectionnez jusqu'à 50 versions de recette pour la politique.

2. Type de filtre : balises — Pour appliquer des règles de cycle de vie aux ressources d'images en fonction des balises de ressources, entrez une liste de 50 paires clé-valeur au maximum auxquelles la politique doit correspondre.

Supprimer la règle

Pour les images de conteneur, cette règle supprime la ressource d'image de conteneur Image Builder. Vous pouvez éventuellement supprimer les images Docker qui ont été distribuées dans les référentiels ECR afin d'éviter qu'elles ne soient utilisées pour exécuter de nouveaux conteneurs.

Lorsque vous configurez la règle de suppression par âge, Image Builder supprime la ressource d'image après une période définie. Supprimez, par exemple, les ressources d'images au bout de 6 mois. Lorsque vous configurez par décompte, Image Builder conserve le nombre d'images le plus récent que vous spécifiez, ou le plus proche possible de ce nombre, et supprime les versions antérieures.

- Par âge
  - Nombre d'unités — Spécifiez la valeur entière correspondant à la période qui doit s'écouler après la création d'une ressource d'image avant sa suppression.
  - Unité — Sélectionnez la plage de temps à utiliser. La plage peut être `Days`,`Weeks`,`Months`, ou`Years`.
  - Conserver au moins une image : cochez cette case pour ne conserver que la dernière ressource d'image disponible pour chaque version de recette affectée par cette règle.

  Par décompte

  - Nombre d'images — Spécifiez la valeur entière du nombre de ressources d'images récentes à conserver pour chaque version de recette.
- Supprimer les images de conteneur ECR : cochez la case pour supprimer les images de conteneur associées stockées dans un référentiel ECR. Vous ne pouvez plus utiliser l'image du conteneur comme base pour créer de nouvelles images ou pour exécuter de nouveaux conteneurs.
- Conserver les images avec les balises associées : cochez la case pour saisir une liste de balises pour les ressources d'images que vous souhaitez conserver.

Les balises (facultatif)

Ajoutez des balises à votre politique de cycle de vie.

AWS CLI

Pour créer une nouvelle politique de cycle de vie d'Image Builder, vous pouvez utiliser la create-lifecycle-policy commande du. AWS CLI

Comment fonctionnent les règles de gestion du cycle de vie pour les ressources d'image EC2 Image Builder

Les politiques de cycle de vie des images utilisent les règles de cycle de vie que vous définissez pour mettre en œuvre votre stratégie globale de gestion des ressources. Les règles que vous définissez permettent de garantir la fraîcheur de vos images disponibles et de minimiser les coûts pour l'infrastructure sous-jacente, telle que le stockage des instantanés pour les AMI de sortie, ou le stockage dans le référentiel ECR et les taux de transfert de données pour les images de conteneur.

Vous pouvez configurer les types de règles suivants pour vos politiques.

Règle de dépréciation

Définit le statut de la ressource d'image Image Builder sur`Deprecated`. Les pipelines Image Builder fonctionnent toujours pour les images obsolètes. Vous pouvez éventuellement définir le délai d'obsolescence des AMI associées sans affecter votre capacité à lancer de nouvelles instances.

Lorsqu'une AMI est obsolète, elle est ignorée par les recherches générales. Par exemple, si vous exécutez la describe-images commande Amazon EC2 dans le AWS CLI, les AMI obsolètes ne seront pas renvoyées dans le jeu de résultats. Cependant, vous pouvez toujours trouver des AMI obsolètes avec leur ID d'AMI.

Cette règle n'est pas disponible pour les images de conteneurs.

Désactiver la règle

Définit le statut de la ressource d'image Image Builder sur`Disabled`. Cela empêche les pipelines Image Builder de s'exécuter pour cette image. Vous pouvez éventuellement désactiver l'AMI associée pour empêcher le lancement de nouvelles instances.

Lorsqu'une AMI est désactivée, elle devient privée et ne peut pas être utilisée pour lancer de nouvelles instances. Si vous avez partagé l'AMI avec des comptes, des organisations ou des unités organisationnelles, ils perdent l'accès à votre AMI lorsqu'elle devient privée.

Cette règle n'est pas disponible pour les images de conteneurs.

Supprimer la règle

Supprime les ressources d'image par âge ou par nombre. Vous définissez le seuil qui répond à vos besoins. Lorsqu'une ressource d'image Image Builder dépasse le seuil, elle est supprimée. Vous pouvez éventuellement désenregistrer les AMI associées ou supprimer les instantanés de ces AMI. Vous pouvez également spécifier des balises pour les ressources que vous souhaitez conserver au-delà du seuil.

Pour les images de conteneur, cette règle supprime la ressource d'image de conteneur Image Builder. Vous pouvez éventuellement supprimer les images de conteneur qui ont été distribuées aux référentiels ECR afin d'éviter qu'elles ne soient utilisées pour exécuter de nouveaux conteneurs.

Table des matières

Règles d'exclusion (API/SDK/CLI)

Les règles d'exclusion suivantes définissent les exceptions aux règles de cycle de vie des AMI. Les AMI qui répondent aux critères spécifiés par les règles d'exclusion sont exclues des actions du cycle de vie. Les règles d'exclusion ne sont pas disponibles dans le AWS Management Console.

Les termes suivants utilisent la notation API issue du type de `LifecyclePolicyDetailExclusionRules` données.

Règles d'exclusion

amis

Contient les paramètres `LifecyclePolicyDetailExclusionRulesAmis` indiqués dans la liste qui suit.

Carte des tags

Vous pouvez fournir une liste contenant jusqu'à 50 balises qui ignorent les actions du cycle de vie pour tout type de ressource.

Les termes suivants utilisent la notation API issue du type de `LifecyclePolicyDetailExclusionRulesAmis` données.

Règles d'exclusion de l'AMI

isPublic

Configure si les AMI publiques sont exclues de l'action du cycle de vie.

Lancé pour la dernière

Spécifie les détails de configuration pour Image Builder afin d'exclure les ressources les plus récentes des actions du cycle de vie.

régions

Configurations Régions AWS exclues de l'action du cycle de vie.

Comptes partagés

Spécifie Comptes AWS les ressources exclues de l'action du cycle de vie.

Carte des tags

Répertorie les balises qui doivent être exclues des actions du cycle de vie des AMI qui en sont dotées.

Afficher le détail des règles de gestion du cycle de vie d'une politique

Les règles sont définies dans les politiques de gestion du cycle de vie que vous créez pour vos ressources d'image Image Builder. Dans la console, la page des détails de la politique de cycle de vie contient un Onglet Règles qui indique les détails des règles que vous avez configurées pour la politique.

Pour obtenir les détails de la politique dans le AWS CLI, vous pouvez exécuter la get-lifecycle-policycommande. Les détails de la politique figurant dans la réponse contiennent une liste des actions (règles) que vous avez définies pour la politique, qui inclut tous les paramètres que vous avez configurés.

Gérez les flux de travail de création et de test pour les images EC2 Image Builder

Un flux de travail d'image définit la séquence d'étapes qu'EC2 Image Builder exécute pendant les étapes de création et de test du processus de création d'image. Cela fait partie de l'infrastructure globale du flux de travail Image Builder.

Avantages du flux de production d'images

- Grâce aux flux de production d'images, vous bénéficiez d'une flexibilité, d'une visibilité et d'un contrôle accrus sur le processus de création d'images.
- Vous pouvez ajouter des étapes de flux de travail personnalisées lorsque vous définissez votre document de flux de travail, ou vous pouvez choisir d'utiliser le flux de travail par défaut d'Image Builder.
- Vous pouvez exclure les étapes du flux de travail incluses dans les flux de travail d'image par défaut.
- Vous pouvez créer des flux de travail de test uniquement qui ignorent complètement le processus de génération. Vous pouvez faire de même pour créer des flux de travail de génération uniquement.

> **Note**
>
> Vous ne pouvez pas modifier un flux de travail existant, mais vous pouvez le cloner ou créer une nouvelle version.

Structure du flux de travail : étapes

Pour personnaliser les flux de travail d'images, il est important de comprendre les étapes du flux de travail qui constituent le cadre du flux de travail de création d'images.

La structure du flux de travail de création d'images comprend les deux étapes distinctes suivantes.

1. Étape de création (pré-instantané) — Au cours de la phase de génération, vous apportez des modifications à l'instance de génération Amazon EC2 qui exécute votre image de base, afin de créer la base de référence pour votre nouvelle image. Par exemple, votre recette peut inclure des

composants qui installent une application ou modifient les paramètres du pare-feu du système d'exploitation.

Une fois cette étape terminée avec succès, Image Builder crée un instantané ou une image conteneur qu'il utilise pour la phase de test et au-delà.

2. Phase de test (post-capture instantanée) — Au cours de la phase de test, il existe certaines différences entre les images qui créent des AMI et les images de conteneur. Pour les flux de travail AMI, Image Builder lance une instance EC2 à partir de l'instantané qu'il a créé comme dernière étape de la phase de création. Des tests sont exécutés sur la nouvelle instance pour valider les paramètres et s'assurer que l'instance fonctionne comme prévu. Pour les flux de travail de conteneurs, les tests s'exécutent sur la même instance que celle utilisée pour la création.

La structure du flux de travail inclut également une étape de distribution. Image Builder gère toutefois les flux de travail pour cette étape.

Accès à un service

Pour exécuter des flux de travail d'images, Image Builder doit être autorisé à effectuer des actions de flux de travail. Vous pouvez spécifier le rôle AWSServiceRoleForImageBuilderlié au service, ou vous pouvez définir votre propre rôle personnalisé pour l'accès au service, comme suit.

- Console — Dans l'assistant de pipeline, étape 3 Définissez le processus de création d'image, sélectionnez le rôle lié au service ou votre propre rôle personnalisé dans la liste des rôles IAM du panneau d'accès aux services.
- API Image Builder : dans la demande CreateImaged'action, spécifiez le rôle lié au service ou votre propre rôle personnalisé comme valeur du paramètre. `executionRole`

Pour en savoir plus sur la création d'un rôle de service, consultez la section Création d'un rôle pour déléguer des autorisations à un AWS service dans le Guide de AWS Identity and Access Management l'utilisateur.

Table des matières

Lister les flux de travail liés

Sur la page de liste des flux de travail d'images de la console Image Builder, vous pouvez obtenir une liste des ressources de flux de travail d'images que vous possédez ou auxquelles vous avez accès, ainsi que des informations clés sur ces ressources. Vous pouvez également utiliser des commandes ou des actions avec l'API Image Builder, les SDK ou AWS CLI pour répertorier les flux de travail liés aux images dans votre compte.

Vous pouvez utiliser l'une des méthodes suivantes pour répertorier les ressources de flux de production d'images dont vous êtes propriétaire ou auxquelles vous avez accès. Pour l'action de l'API, voir le ListWorkflowsmanuel de référence de l'API EC2 Image Builder. Pour la demande de SDK associée, reportez-vous au lien Voir aussi sur la même page.

Console

Détails du flux de travail

La page de liste des flux de travail Image de la console Image Builder contient les informations suivantes :

- Flux de travail : nom de la version la plus récente de la ressource de flux de travail d'images. Dans la console Image Builder, la colonne Workflow renvoie à la page détaillée du flux de travail.
- Version : version la plus récente de la ressource de flux de travail d'images.
- Type — Type de flux de travail : `BUILD` ou`TEST`.
- Propriétaire : propriétaire de la ressource de flux de travail.
- Heure de création : date et heure auxquelles Image Builder a créé la version la plus récente de la ressource de flux de travail d'images.
- ARN — Le nom de ressource Amazon (ARN) de la version actuelle de la ressource de flux de travail d'images.

Lister les flux de travail liés

Pour répertorier les ressources du flux de travail lié aux images dans la console Image Builder, effectuez les opérations suivantes :

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.

2. Choisissez Image workflows dans le volet de navigation.

Filtrer les résultats

Sur la page de liste des flux de travail d'images, vous pouvez rechercher des flux de travail d'images spécifiques pour filtrer vos résultats. Les filtres suivants sont disponibles pour les flux de production d'images :

`Workflow`

Vous pouvez saisir tout ou partie du nom d'un flux de travail pour rationaliser les résultats. Par défaut, tous les flux de travail sont affichés dans la liste.

`Version`

Vous pouvez saisir tout ou partie d'un numéro de version pour rationaliser les résultats. Par défaut, toutes les versions sont affichées dans la liste.

`Type`

Vous pouvez filtrer par type de flux de travail ou afficher tous les types. Par défaut, tous les types de flux de travail sont affichés dans la liste.

- BUILD
- TESTER

`Owner`

Lorsque vous sélectionnez le filtre propriétaire dans la barre de recherche, Image Builder affiche la liste des propriétaires des flux de production d'images de votre compte. Vous pouvez sélectionner un propriétaire dans la liste pour rationaliser les résultats. Par défaut, tous les propriétaires sont affichés dans la liste.

- Compte AWS— Le compte propriétaire de la ressource de flux de travail.
- Amazon — Ressources de flux de travail détenues et gérées par Amazon.

AWS CLI

Lorsque vous exécutez la list-workflows commande dans le AWS CLI, vous pouvez obtenir une liste des flux de travail d'images que vous possédez ou auxquels vous avez accès.

L'exemple de commande suivant montre comment utiliser la list-workflows commande sans filtre pour répertorier toutes les ressources du flux de travail d'image Image Builder que vous possédez ou auxquelles vous avez accès.

Exemple : liste de tous les flux de production d'images

```
aws imagebuilder list-workflows
```

Sortie :

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

Lorsque vous exécutez la list-workflows commande, vous pouvez appliquer des filtres pour rationaliser les résultats, comme le montre l'exemple suivant. Pour plus d'informations sur le filtrage des résultats, consultez la commande list-workflows dans la référence des AWS CLI commandes.

Exemple : filtre pour les flux de travail de création

```
aws imagebuilder list-workflows --filters name="type",values="BUILD"
```

Sortie :

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

Création d'un flux de production d'images

Lorsque vous créez un flux de travail d'images, vous avez un meilleur contrôle sur le processus de création d'images. Vous pouvez spécifier quel flux de travail s'exécute lorsque Image Builder crée votre image et quels flux de travail s'exécutent lorsqu'il teste l'image. Vous pouvez également spécifier une clé gérée par le client pour chiffrer les ressources de votre flux de travail. Pour en savoir plus sur le chiffrement des ressources de votre flux de travail, consultezChiffrement et gestion des clés dans EC2 Image Builder.

Pour la création d'images, vous pouvez spécifier un flux de travail en phase de création et un ou plusieurs flux de travail en phase de test. Vous pouvez même sauter complètement la phase de construction ou de test, selon vos besoins. Vous configurez les actions effectuées par votre flux de travail dans le document de définition YAML qu'il utilise. Pour plus d'informations sur la syntaxe de votre document YAML, consultezCréation d'un document de flux de travail YAML.

Pour connaître les étapes de création d'un nouveau flux de travail de génération ou de test, sélectionnez l'onglet correspondant à l'environnement que vous utiliserez.

AWS Management Console

Vous pouvez utiliser le processus suivant pour créer un nouveau flux de travail dans la console Image Builder.

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.

2. Choisissez Image workflows dans le volet de navigation. Cela affiche une liste des flux de production d'images que votre compte possède ou auxquels il a accès.

   > **Note**
   >
   > Vous verrez toujours dans votre liste les ressources de flux de travail gérées par Amazon qu'Image Builder utilise pour ses flux de travail par défaut. Pour afficher les détails de ces flux de travail, vous pouvez sélectionner le lien Workflow.

3. Pour créer un nouveau flux de travail, choisissez Créer un flux de travail d'images. Cela affiche la page Créer un flux de travail d'image.
4. Configurez les détails de votre nouveau flux de travail. Pour créer un flux de travail de génération, sélectionnez l'option Créer en haut du formulaire. Pour créer un flux de travail de test, sélectionnez l'option Test en haut du formulaire. Image Builder remplit la liste des modèles en fonction de cette option. Toutes les autres étapes sont les mêmes pour les flux de travail de création et de test.

   Général

   La section générale inclut les paramètres qui s'appliquent à votre ressource de flux de travail, tels que le nom et la description. Les paramètres généraux sont les suivants :

   - Nom du flux de travail d'images (obligatoire) : nom de votre flux de travail d'images. Le nom doit être unique dans votre compte. Le nom peut comporter jusqu'à 128 caractères. Les caractères valides incluent les lettres, les chiffres-, les espaces et_.
   - Version (obligatoire) : version sémantique de la ressource de flux de travail à créer (major.minor.patch).
   - Description (facultatif) — Ajoutez éventuellement une description de votre flux de travail.
   - Clé KMS (facultatif) — Vous pouvez chiffrer les ressources de votre flux de travail à l'aide d'une clé gérée par le client. Pour plus d'informations, consultez Chiffrez les flux de production d'images à l'aide d'une clé gérée par le client.

   Document de définition

   Le document de flux de travail YAML contient toute la configuration de votre flux de travail.

Mise en route

- Pour commencer avec un modèle par défaut d'Image Builder comme référence pour votre flux de travail, sélectionnez l'option Commencer à partir des modèles. Cette option est sélectionnée par défaut. Après avoir choisi le modèle à utiliser dans la liste des modèles, la configuration par défaut du modèle que vous avez sélectionné est copiée dans le contenu de votre nouveau document de flux de travail, où vous pouvez apporter des modifications.
- Pour définir votre document de flux de travail à partir de zéro, sélectionnez l'option Commencer à zéro. Cela permet de remplir le contenu avec un bref aperçu de certaines parties importantes du format du document pour vous aider à démarrer.

Le panneau Contenu inclut une barre d'état en bas qui affiche les avertissements ou les erreurs relatifs à votre document YAML. Pour plus d'informations sur la création d'un document de flux de travail YAML, consultezCréation d'un document de flux de travail YAML.

5. Lorsque vous avez terminé votre flux de travail, ou si vous souhaitez enregistrer la progression et y revenir ultérieurement, choisissez Créer un flux de travail.

AWS CLI

Avant d'exécuter la create-workflow commande dans le AWS CLI, vous devez créer le document YAML contenant l'ensemble de la configuration de votre flux de travail. Pour plus d'informations, consultez Création d'un document de flux de travail YAML.

L'exemple suivant montre comment créer un flux de travail de génération à l'aide de la commande create-workflow AWS CLI . Le `--data` paramètre fait référence à un document YAML contenant la configuration de compilation du flux de travail que vous créez.

Exemple : créer un flux de travail

```
aws imagebuilder create-workflow --name example-build-workflow --semantic-
version 1.0.0 --type BUILD --data file://example-build-workflow.yml
```

Sortie :

```
{
"workflowBuildVersionArn": "arn:aws:imagebuilder:us-west-2:111122223333:workflow/
build/example-build-workflow/1.0.0/1",
```

```
"clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

L'exemple suivant montre comment créer un flux de travail de test à l'aide de la commande create-workflow AWS CLI . Le `--data` paramètre fait référence à un document YAML contenant la configuration de compilation du flux de travail que vous créez.

Exemple : création d'un flux de travail de test

```
aws imagebuilder create-workflow --name example-test-workflow --semantic-
version 1.0.0 --type TEST --data file://example-test-workflow.yml
```

Sortie :

```
{
"workflowBuildVersionArn": "arn:aws:imagebuilder:us-west-2:111122223333:workflow/
test/example-test-workflow/1.0.0/1",
"clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

Création d'un document de flux de travail YAML

Le document de définition du format YAML configure les étapes d'entrée, de sortie et de flux de travail pour les étapes de génération et de test du processus de création d'image. Vous pouvez partir de modèles qui incluent des étapes standardisées, ou vous pouvez partir de zéro pour définir votre propre flux de travail. Que vous utilisiez un modèle ou que vous partiez de zéro, vous pouvez personnaliser le flux de travail en fonction de vos besoins.

Structure d'un document de flux de travail YAML

Le document de flux de travail YAML utilisé par Image Builder pour effectuer des actions de création et de test d'images est structuré comme suit.

- Identification
- Paramètres d'entrée
- Étapes
- Outputs

Identification

Identifie le flux de travail de manière unique. Cette section peut inclure les attributs suivants.

Champ	Description	Type	Obligatoire
name	Nom du document de flux de travail.	Chaîne	Non
description	Description du document.	Chaîne	Non
schemaVersion	La version du schéma du document, actuellement 1.0.	Chaîne	Oui

Exemple

```
---
name: sample-test-image
description: Workflow for a sample image, with extra configuration options exposed
 through workflow parameters.
schemaVersion: 1.0
```

Paramètres d'entrée

Cette partie du document de flux de travail définit les paramètres d'entrée que l'appelant peut spécifier. Si vous n'avez aucun paramètre, vous pouvez omettre cette section. Si vous spécifiez des paramètres, chaque paramètre peut inclure les attributs suivants.

Champ	Description	Type	Obligatoire	Constraints
name	Nom du paramètre.	Chaîne	Oui	

Champ	Description	Type	Obligatoire	Constraints
description	Description du paramètre.	Chaîne	Non	
default	La valeur par défaut du paramètre, si aucune valeur n'est fournie. Si vous n'incluez pas de valeur par défaut dans la définition du paramètre , la valeur du paramètre est requise lors de l'exécution.	Correspond au type de données du paramètre.	Non	
type	Type de données du paramètre. Si vous n'incluez pas le type de données dans la définition du paramètre , le type de paramètre est par défaut une valeur de chaîne requise lors de l'exécution.	Chaîne	Oui	Le type de données du paramètre doit être l'un des suivants : • `string` • `integer` • `boolean` • `stringList`

Exemple

Spécifiez le paramètre dans le document de flux de travail.

```
parameters:
  - name: waitForActionAtEnd
    type: boolean
    default: true
    description: "Wait for an external action at the end of the workflow"
```

Utilisez la valeur du paramètre dans le document de flux de travail.

```
$.parameters.waitForActionAtEnd
```

Étapes

Spécifie jusqu'à 15 étapes d'actions pour le flux de travail. Les étapes s'exécutent dans l'ordre dans lequel elles sont définies dans le document de flux de travail. En cas d'échec, une annulation s'exécute dans l'ordre inverse, en commençant par l'étape qui a échoué et en remontant jusqu'aux étapes précédentes.

Chaque étape peut faire référence au résultat de toute action d'étape précédente. C'est ce que l'on appelle le chaînage ou le référencement. Pour faire référence au résultat d'une action d'une étape précédente, vous pouvez utiliser un sélecteur JSONPath. Par exemple :

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

Pour plus d'informations, consultez Utiliser des variables dynamiques dans votre document de flux de travail.

> **Note**
>
> Même si l'étape elle-même ne possède pas d'attribut de sortie, toute sortie d'une action d'étape est incluse dans `stepOutput` l'étape.

Chaque étape peut inclure les attributs suivants.

Champ	Description	Type	Obligatoire	Valeur par défaut	Constraints
action	Action de flux de travail exécutée par cette étape.	Chaîne	Oui		Doit être une action d'étape prise en charge pour les documents de flux de travail Image Builder.
`if`, suivi d'un ensemble d'instructions condition nelles qui modifient l'`if`opérateur.	Les instructi ons condition nelles ajoutent un flux de points de décision de contrôle au corps des étapes de votre flux de travail.	Dict	Non		Image Builder prend en charge les instructions condition nelles suivantes en tant que modificat eurs de l'`if`opérateur : • Condition s de brancheme nt et modificat eurs :`if`,,`and`,`or`. `not` Les condition s de

Champ	Description	Type	Obligatoire	Valeur par défaut	Constraints
					brancheme nt sont spécifiée s sur une ligne par elles-mêm es. • Opérateur s de comparais on : `booleanEq uals numberEqu als ,numberGre aterThan ,numbe aterThanE quals ,,numberLe sThan ,numberLes sThanEqua ls ,stringEqu als .`

Champ	Description	Type	Obligatoire	Valeur par défaut	Constraints
description	Description de l'étape.	Chaîne	Non		Les chaînes vides ne sont pas autorisées. Si elle est incluse, la longueur doit être comprise entre 1 et 1024 caractères.
inputs	Contient les paramètres nécessaires à l'exécution de l'action de l'étape. Vous pouvez spécifier des valeurs clés sous forme de valeurs statiques ou à l'aide d'une variable JSONPath qui correspond au type de données approprié.	Dict	Oui		

Champ	Description	Type	Obligatoire	Valeur par défaut	Constraints
name	Nom de l'étape. Ce nom doit être unique dans le document de flux de travail.	Chaîne	Oui		La longueur doit être comprise entre 3 et 128 caractères. Peut inclure des caractères alphanumériques et. _ Pas d'espaces.

Champ	Description	Type	Obligatoire	Valeur par défaut	Constraints
onFailure	Configure l'action à effectuer en cas d'échec de l'étape, comme suit. Attitude • `Abort`— Échoue l'étape, échoue le flux de travail et n'exécute aucune étape restante après l'étape qui a échoué. Si la restauration est activée, elle commence par l'étape qui a échoué et se poursuit	Chaîne	Non	`Abort`	`Abort` | `Continue`

Champ	Description	Type	Obligatoire	Valeur par défaut	Constraints
	jusqu'à ce que toutes les étapes qui l'autorisent soient annulées. • `Continue`— Échoue l'étape, mais continue d'exécuter les étapes restantes après l'étape qui a échoué. Dans ce cas, il n'y a pas de retour en arrière.				

Champ	Description	Type	Obligatoire	Valeur par défaut	Constraints
Rollback activé	Configure si l'étape sera annulée en cas d'échec. Vous pouvez utiliser une valeur booléenne statique ou une variable JSONPath dynamique qui se résout en une valeur booléenne.	Booléen	Non	`true`	`true` | `false` | ou une variable JSONPath qui prend la valeur true ou false.
timeoutSeconds	Durée maximale, en secondes, pendant laquelle l'étape s'exécute avant d'échouer et de réessayer, si de nouvelles tentatives s'appliquent.	Entier	Non	Dépend de la valeur par défaut définie pour l'action de l'étape, le cas échéant.	Entre 1 et 86 400 secondes (24 heures maximum)

Exemple

```
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

Définit les sorties pour le flux de travail. Chaque sortie est une paire clé-valeur qui spécifie le nom de la sortie et la valeur. Vous pouvez utiliser les sorties pour exporter des données au moment de l'exécution que les flux de travail suivants pourront utiliser. Cette section est facultative.

Chaque sortie que vous définissez inclut les attributs suivants.

Champ	Description	Type	Obligatoire
name	Nom de la sortie. Le nom doit être unique pour tous les flux	Chaîne	Oui

Champ	Description	Type	Obligatoire
	de travail que vous incluez dans votre pipeline.		
value	La valeur de la sortie. La valeur de la chaîne peut être une variable dynamique , telle qu'un fichier de sortie issu d'une action d'étape. Pour plus d'informations, consultez Utiliser des variables dynamiques dans votre document de flux de travail.	Chaîne	Oui

Exemple

Créez un ID d'image de sortie pour le document de flux de travail avec le résultat de l'`createProdImage`étape.

```
outputs:
  - name: 'outputImageId'
    value: '$.stepOutputs.createProdImage.imageId'
```

Reportez-vous au résultat du flux de travail dans le flux de travail suivant.

```
$.workflowOutputs.outputImageId
```

Actions d'étapes prises en charge pour votre document de flux de travail

Cette section contient des informations détaillées sur les actions d'étape prises en charge par Image Builder.

Termes utilisés dans cette section

AMI

Amazon Machine Image

ARN

Nom de la ressource Amazon

Actions soutenues

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

BootstrapInstanceForContainer

Cette action d'étape exécute un script de service pour démarrer l'instance avec les exigences minimales requises pour exécuter des flux de travail de conteneurs. Image Builder utilise sendCommand l'API de Systems Manager pour exécuter ce script. Pour plus d'informations, consultez la section AWS Systems Manager Exécuter la commande.

> **Note**
>
> Le script bootstrap installe les packages Docker AWS CLI et Docker qui sont indispensables pour qu'Image Builder puisse créer des conteneurs Docker avec succès. Si vous n'incluez pas cette action, la création de l'image risque d'échouer.

Délai d'expiration par défaut : 60 minutes

Annulation : il n'y a aucune annulation pour cette action d'étape.

Entrées : Le tableau suivant inclut les entrées prises en charge pour cette action d'étape.

Nom d'entrée	Description	Type	Obligatoire	Par défaut	Constraints
instanceId	L'ID de l'instance à démarrer.	Chaîne	Oui		Il doit s'agir de l'ID d'instance de sortie de l'étape de flux de travail qui a lancé l'instance pour ce flux de travail.

Sorties : Le tableau suivant inclut les sorties pour cette action d'étape.

Nom de sortie	Description	Type
runCommandId	ID du Systems Manager sendCommand qui a exécuté le script bootstrap sur l'instance.	Chaîne
status	Le statut renvoyé par le Systems ManagersendCommand.	Chaîne
output	Sortie renvoyée par le Systems ManagersendCommand.	Chaîne

Exemple

Spécifiez l'action de l'étape dans le document de flux de travail.

```
- name: ContainerBootstrapStep
  action: BootstrapInstanceForContainer
  onFailure: Abort
  inputs:
      instanceId.$: $.stepOutputs.LaunchStep.instanceId
```

Utilisez le résultat de la valeur de l'action de l'étape dans le document de flux de travail.

```
$.stepOutputs.ContainerBootstrapStep.status
```

CollectImageMetadata

Cette action d'étape n'est valide que pour les flux de travail de génération.

EC2 Image Builder AWS Systems Manager exécute l'agent (Systems Manager) sur les instances EC2 qu'il lance pour créer et tester votre image. Image Builder collecte des informations supplémentaires sur l'instance utilisée pendant la phase de construction avec Systems Manager Inventory. Ces informations incluent le nom et la version du système d'exploitation (OS), ainsi que la liste des packages et de leurs versions respectives tels qu'indiqués par votre système d'exploitation.

> **Note**
>
> Cette action de cette étape ne fonctionne que pour les images qui créent des AMI.

Délai d'expiration par défaut : 30 minutes

Annulation : Image Builder annule toutes les ressources Systems Manager créées au cours de cette étape.

Entrées : Le tableau suivant inclut les entrées prises en charge pour cette action d'étape.

Nom d'entrée	Description	Type	Obligatoire	Par défaut	Constraints
instanceId	L'instanc e de build à laquelle	Chaîne	Oui		Il doit s'agir de l'ID d'instance

Nom d'entrée	Description	Type	Obligatoire	Par défaut	Constraints
	appliquer les paramètres de métadonnées.				de sortie de l'étape de flux de travail qui a lancé l'instance de génération pour ce flux de travail.

Sorties : Le tableau suivant inclut les sorties pour cette action d'étape.

Nom de sortie	Description	Type
osVersion	Le nom et la version du système d'exploitation collectés à partir de l'instance de build.	Chaîne
Identifiant de l'association	L'ID d'association Systems Manager utilisé pour la collecte de l'inventaire.	Chaîne

Exemple

Spécifiez l'action de l'étape dans le document de flux de travail.

```
- name: CollectMetadataStep
  action: CollectImageMetadata
  onFailure: Abort
  inputs:
      instanceId: $.stepOutputs.LaunchStep.instanceId
```

Utilisez le résultat de l'action de l'étape dans le document de flux de travail.

```
$.stepOutputs.CollectMetadataStep.osVersion
```

CollectImageScanFindings

Si Amazon Inspector est activé pour votre compte et que la numérisation d'images est activée pour votre pipeline, cette étape collecte les résultats de numérisation d'images signalés par Amazon Inspector pour votre instance de test. Cette action d'étape n'est pas disponible pour les flux de travail de génération.

Délai d'expiration par défaut : 120 minutes

Annulation : il n'y a aucune annulation pour cette action d'étape.

Entrées : Le tableau suivant inclut les entrées prises en charge pour cette action d'étape.

Nom d'entrée	Description	Type	Obligatoire	Par défaut	Constraints
instanceId	ID de l'instance sur laquelle le scan a été exécuté.	Chaîne	Oui		Il doit s'agir de l'ID d'instance de sortie de l'étape de flux de travail qui a lancé l'instance pour ce flux de travail.

Sorties : Le tableau suivant inclut les sorties pour cette action d'étape.

Nom de sortie	Description	Type
runCommandId	ID du Systems Manager sendCommand qui a exécuté le script pour collecter les résultats.	Chaîne
status	Le statut renvoyé par le Systems ManagersendCommand.	Chaîne

Nom de sortie	Description	Type
output	Sortie renvoyée par le Systems ManagersendCommand.	Chaîne

Exemple

Spécifiez l'action de l'étape dans le document de flux de travail.

```
- name: CollectFindingsStep
  action: CollectImageScanFindings
  onFailure: Abort
  inputs:
      instanceId.$: $.stepOutputs.LaunchStep.instanceId
```

Utilisez le résultat de la valeur de l'action de l'étape dans le document de flux de travail.

```
$.stepOutputs.CollectFindingsStep.status
```

CreateImage

Cette étape crée une image à partir d'une instance en cours d'exécution avec l'API Amazon EC2. `CreateImage` Pendant le processus de création, l'action de l'étape attend autant que nécessaire pour vérifier que les ressources ont atteint le bon état avant de continuer.

Délai d'expiration par défaut : 720 minutes

Annulation : il n'y a aucune annulation pour cette action d'étape.

Entrées : Le tableau suivant inclut les entrées prises en charge pour cette action d'étape.

Nom d'entrée	Description	Type	Obligatoire	Par défaut	Constraints
instanceId	L'instance à partir de laquelle créer la nouvelle image.	Chaîne	Oui		L'instance correspondant à l'ID d'instance fourni doit

Nom d'entrée	Description	Type	Obligatoire	Par défaut	Constraints
					être dans un `running` état au début de cette étape.

Sorties : Le tableau suivant inclut les sorties pour cette action d'étape.

Nom de sortie	Description	Type
imageId	ID AMI de l'image créée.	Chaîne

Exemple

Spécifiez l'action de l'étape dans le document de flux de travail.

```
- name: CreateImageFromInstance
  action: CreateImage
  onFailure: Abort
  inputs:
      instanceId.$: "i-1234567890abcdef0"
```

Utilisez le résultat de la valeur de l'action de l'étape dans le document de flux de travail.

```
$.stepOutputs.CreateImageFromInstance.imageId
```

ExecuteComponents

Cette étape exécute les composants spécifiés dans la recette de l'image en cours de création. Les flux de travail de génération exécutent les composants de génération sur l'instance de génération. Les flux de travail de test exécutent uniquement les composants de test sur l'instance de test.

Image Builder utilise sendCommand l'API de Systems Manager pour exécuter les composants. Pour plus d'informations, consultez la section AWS Systems Manager Exécuter la commande.

Délai d'expiration par défaut : 720 minutes

Annulation : il n'y a aucune annulation pour cette action d'étape.

Entrées : Le tableau suivant inclut les entrées prises en charge pour cette action d'étape.

Nom d'entrée	Description	Type	Obligatoire	Par défaut	Constraints
instanceId	L'ID de l'instance sur laquelle les composant s doivent s'exécuter.	Chaîne	Oui		Il doit s'agir de l'ID d'instance de sortie de l'étape de flux de travail qui a lancé l'instance pour ce flux de travail.

Sorties : Le tableau suivant inclut les sorties pour cette action d'étape.

Nom de sortie	Description	Type
runCommandId	ID du Systems Manager sendCommand qui a exécuté les composants sur l'instance.	Chaîne
status	Le statut renvoyé par le Systems ManagersendCommand.	Chaîne
output	Sortie renvoyée par le Systems ManagersendCommand.	Chaîne

Exemple

Spécifiez l'action de l'étape dans le document de flux de travail.

```
- name: ExecComponentsStep
  action: ExecuteComponents
```

```
  onFailure: Abort
  inputs:
      instanceId: $.stepOutputs.LaunchStep.instanceId
```

Utilisez le résultat de l'action de l'étape dans le document de flux de travail.

```
$.stepOutputs.ExecComponentsStep.status
```

LaunchInstance

Cette action lance une instance dans votre instance Compte AWS et attend que l'agent Systems Manager soit exécuté sur l'instance avant de passer à l'étape suivante. L'action de lancement utilise les paramètres de votre recette et les ressources de configuration de l'infrastructure associées à votre image. Par exemple, le type d'instance à lancer provient de la configuration de l'infrastructure. Le résultat est l'ID d'instance de l'instance qu'il a lancée.

L'`waitFor`entrée configure la condition qui satisfait à l'exigence d'achèvement de l'étape.

Délai d'expiration par défaut : 60 minutes

Annulation : pour les instances de build, la restauration exécute l'action que vous avez configurée dans votre ressource de configuration d'infrastructure. Par défaut, les instances de build sont arrêtées en cas d'échec de la création de l'image. Cependant, la configuration de l'infrastructure comporte un paramètre permettant de conserver l'instance de build à des fins de dépannage.

Entrées : Le tableau suivant inclut les entrées prises en charge pour cette action d'étape.

Nom d'entrée	Description	Type	Obligatoire	Par défaut	Constraints
Attendre	Condition à attendre avant de terminer l'étape du flux de travail et de passer à l'étape suivante.	Chaîne	Oui		Image Builder prend actuellement en charge `ssmAgent`.

Sorties : Le tableau suivant inclut les sorties pour cette action d'étape.

Nom de sortie	Description	Type
instanceId	ID d'instance de l'instance qui a été lancée.	Chaîne

Exemple

Spécifiez l'action de l'étape dans le document de flux de travail.

```
- name: LaunchStep
  action: LaunchInstance
  onFailure: Abort
  inputs:
    waitFor: ssmAgent
```

Utilisez le résultat de l'action de l'étape dans le document de flux de travail.

```
$.stepOutputs.LaunchStep.instanceId
```

RunCommand

Cette étape exécute un document de commande pour votre flux de travail. Image Builder utilise sendCommand l'API de Systems Manager pour l'exécuter pour vous. Pour plus d'informations, consultez la section AWS Systems Manager Exécuter la commande.

Délai d'expiration par défaut : 12 heures

Annulation : il n'y a aucune annulation pour cette action d'étape.

Entrées : Le tableau suivant inclut les entrées prises en charge pour cette action d'étape.

Nom d'entrée	Description	Type	Obligatoire	Par défaut	Constraints
instanceId	ID de l'instance sur laquelle exécuter le	Chaîne	Oui		Il doit s'agir de l'ID d'instance de sortie de l'étape de

Nom d'entrée	Description	Type	Obligatoire	Par défaut	Constraints
	document de commande.				flux de travail qui a lancé l'instance pour ce flux de travail.
documentName	Nom du document de commande Systems Manager à exécuter.	Chaîne	Oui		
parameters	Une liste de paires clé-valeur pour tous les paramètres requis par le document de commande.	<string>dictionnaire<chaîne, liste >	Conditionnel		
Version du document	Version du document de commande à exécuter.	Chaîne	Non	$DEFAULT	

Sorties : Le tableau suivant inclut les sorties pour cette action d'étape.

Nom de sortie	Description	Type
runCommandId	ID du Systems Manager sendCommand qui a exécuté le document de commande sur l'instance.	Chaîne

Nom de sortie	Description	Type
status	Le statut renvoyé par le Systems ManagersendCommand.	Chaîne
output	Sortie renvoyée par le Systems ManagersendCommand.	Liste de chaînes

Exemple

Spécifiez l'action de l'étape dans le document de flux de travail.

```
- name: RunCommandDoc
  action: RunCommand
  onFailure: Abort
  inputs:
    documentName: SampleDocument
    parameters:
        osPlatform:
          - "linux"
    instanceId.$: $.stepOutputs.LaunchStep.instanceId
```

Utilisez le résultat de la valeur de l'action de l'étape dans le document de flux de travail.

```
$.stepOutputs.RunCommandDoc.status
```

RunSysPrep

Cette action utilise l'API sendCommand in the Systems Manager pour exécuter le `AWSEC2-RunSysprep` document pour les instances Windows avant que l'instance de build ne s'arrête pour le snapshot. Ces actions suivent les AWS meilleures pratiques en matière de renforcement et de nettoyage de l'image.

Délai d'expiration par défaut : 60 minutes

Annulation : il n'y a aucune annulation pour cette action d'étape.

Entrées : Le tableau suivant inclut les entrées prises en charge pour cette action d'étape.

Nom d'entrée	Description	Type	Obligatoire	Par défaut	Constraints
instanceId	ID de l'instance sur laquelle exécuter le `AWSEC2-RunSysprep` document.	Chaîne	Oui		Il doit s'agir de l'ID d'instance de sortie de l'étape de flux de travail qui a lancé l'instance pour ce flux de travail.

Sorties : Le tableau suivant inclut les sorties pour cette action d'étape.

Nom de sortie	Description	Type
runCommandId	ID du Systems Manager sendCommand qui a exécuté le `AWSEC2-RunSysprep` document sur l'instance.	Chaîne
status	Le statut renvoyé par le Systems ManagersendCommand.	Chaîne
output	Sortie renvoyée par le Systems ManagersendCommand.	Chaîne

Exemple

Spécifiez l'action de l'étape dans le document de flux de travail.

```
- name: RunSysprep
  action: RunSysPrep
```

```
  onFailure: Abort
  inputs:
      instanceId.$: $.stepOutputs.LaunchStep.instanceId
```

Utilisez le résultat de la valeur de l'action de l'étape dans le document de flux de travail.

```
$.stepOutputs.RunSysprep.status
```

SanitizeInstance

Cette étape exécute le script de nettoyage recommandé pour les instances Linux avant que l'instance de build ne s'arrête pour le snapshot. Le script de désinfection permet de garantir que l'image finale respecte les meilleures pratiques de sécurité et que les artefacts ou paramètres de build qui ne devraient pas être répercutés sur votre instantané sont supprimés. Pour plus d'informations sur le script, consultezNettoyage requis après la construction. Cette action de cette étape ne s'applique pas aux images du conteneur.

Image Builder utilise sendCommand l'API de Systems Manager pour exécuter ce script. Pour plus d'informations, consultez la section AWS Systems Manager Exécuter la commande.

Délai d'expiration par défaut : 60 minutes

Annulation : il n'y a aucune annulation pour cette action d'étape.

Entrées : Le tableau suivant inclut les entrées prises en charge pour cette action d'étape.

Nom d'entrée	Description	Type	Obligatoire	Par défaut	Constraints
instanceId	ID de l'instance à désinfecter.	Chaîne	Oui		Il doit s'agir de l'ID d'instance de sortie de l'étape de flux de travail qui a lancé l'instance pour ce flux de travail.

Sorties : Le tableau suivant inclut les sorties pour cette action d'étape.

Nom de sortie	Description	Type
runCommandId	L'ID du Systems Manager sendCommand qui a exécuté le script de nettoyage sur l'instance.	Chaîne
status	Le statut renvoyé par le Systems ManagersendCommand.	Chaîne
output	Sortie renvoyée par le Systems ManagersendCommand.	Chaîne

Exemple

Spécifiez l'action de l'étape dans le document de flux de travail.

```
- name: SanitizeStep
  action: SanitizeInstance
  onFailure: Abort
  inputs:
      instanceId: $.stepOutputs.LaunchStep.instanceId
```

Utilisez le résultat de la valeur de l'action de l'étape dans le document de flux de travail.

```
$.stepOutputs.SanitizeStep.status
```

TerminateInstance

Cette étape met fin à l'instance avec l'identifiant d'instance transmis en entrée.

Délai d'expiration par défaut : 30 minutes

Annulation : il n'y a aucune annulation pour cette action d'étape.

Entrées : Le tableau suivant inclut les entrées prises en charge pour cette action d'étape.

Nom d'entrée	Description	Type	Obligatoire	Par défaut	Constraints
instanceId	L'ID de l'instance à terminer.	Chaîne	Oui		

Sorties : Il n'y a aucune sortie pour cette action d'étape.

Exemple

Spécifiez l'action de l'étape dans le document de flux de travail.

```
- name: TerminateInstance
  action: TerminateInstance
  onFailure: Continue
  inputs:
      instanceId.$: i-1234567890abcdef0
```

WaitForAction

Cette étape interrompt le flux de travail en cours d'exécution et attend de recevoir une action externe de la part de l'action d'SendWorkflowStepActionAPI Image Builder. Cette étape publie un EventBridge événement sur votre bus d' EventBridge événements par défaut avec un type de détail`EC2 Image Builder Workflow Step Waiting`. L'étape peut également envoyer une notification SNS si vous fournissez un ARN de rubrique SNS.

Délai d'expiration par défaut : 3 jours

Annulation : il n'y a aucune annulation pour cette action d'étape.

Entrées : Le tableau suivant inclut les entrées prises en charge pour cette action d'étape.

Nom d'entrée	Description	Type	Obligatoire	Par défaut	Constraints
snsTopicArn	Un ARN de rubrique SNS facultati	Chaîne	Non		

Nom d'entrée	Description	Type	Obligatoire	Par défaut	Constraints
	f auquel envoyer une notificat ion lorsque l'étape du flux de travail est en attente.				

Sorties : Le tableau suivant inclut les sorties pour cette action d'étape.

Nom de sortie	Description	Type
action	L'action renvoyée par l'action d'SendWorkflowStepAc tionAPI.	Chaîne (`RESUMEouSTOP`)
raison	Motif de l'action renvoyée.	Chaîne

Exemple

Spécifiez l'action de l'étape dans le document de flux de travail.

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

Utilisez le résultat de la valeur de l'action de l'étape dans le document de flux de travail.

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

Utiliser des variables dynamiques dans votre document de flux de travail

Vous pouvez utiliser des variables dynamiques dans vos documents de flux de travail pour représenter des valeurs qui varient au moment de l'exécution de votre processus de création d'image.

Les valeurs des variables dynamiques sont représentées sous forme de sélecteurs JSONPath dotés de nœuds structurels qui identifient de manière unique la variable cible.

Structure variable du flux de travail dynamique JSONPath

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

Le premier nœud après la racine ($) fait référence à la structure du document du flux de travail, telle que`stepOutputs`, ou dans le cas des variables système Image Builder,`imageBuilder`. La liste suivante contient les nœuds de structure de document de flux de travail JSONPath pris en charge.

Nœuds de structure de document

- paramètres - Les paramètres du flux de travail
- StepOutputs - Sorties d'une étape du même document de flux de travail
- WorkflowOutputs - Sorties d'un document de flux de travail déjà exécuté
- imagebuilder - Variables système Image Builder

Les nœuds `parameters` et structure du `stepOutputs` document incluent un nœud facultatif pour le nom de l'étape. Cela permet de garantir des noms de variables uniques pour toutes les étapes.

Le dernier nœud du JSONPath est le nom de la variable cible, par exemple. `instanceId`

Chaque étape peut faire référence à la sortie de toute action d'étape précédente avec ces variables dynamiques JSONPath. Ceci est également connu sous le nom de chaînage ou de référencement. Pour faire référence au résultat d'une action d'une étape précédente, vous pouvez utiliser la variable dynamique suivante.

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

Exemple

```
  - name: ApplyTestComponents
    action: ExecuteComponents
    onFailure: Abort
    inputs:
      instanceId.$: "$.stepOutputs.LaunchTestInstance.instanceId"
```

Utiliser les variables système Image Builder

Image Builder fournit les variables système suivantes que vous pouvez utiliser dans votre document de flux de travail :

Nom de variable	Description	Type	Exemple de valeur
cloudWatchLogGroup e	Nom du groupe CloudWatch Logs pour les logs de sortie. Format : `/aws/ imagebuilder/ <recipe-name>`	Chaîne	`/aws/imag ebuilder/ sampleIma geRecipe`
cloudWatchLogStrea m	Nom du flux CloudWatch Logs pour les logs de sortie.	Chaîne	`1,0/1`
collectImageMetadata	Le paramètre qui indique à Image Builder s'il doit collecter les métadonnées de l'instance.	Booléen	`true` | `false`
collectImageScanCo nclusions	La valeur actuelle du paramètre qui permet à Image Builder de collecter les résultats de numérisation d'images.	Booléen	`true` | `false`

Nom de variable	Description	Type	Exemple de valeur
imageBuildNumber	Numéro de version de compilation de l'image.	Entier	*1*
imageId	ID AMI de l'image de base.	Chaîne	*ami-12345 67890abcdef1*
Nom de l'image	Le nom de l'image.	Chaîne	*Exemple d'image*
imageType	Type de sortie d'image.	Chaîne	`AMI` | `Docker`
imageVersionNumber	Numéro de version de l'image.	Chaîne	*1.0.0*
instanceProfileName	Nom du rôle de profil d'instance utilisé par Image Builder pour lancer les instances de génération et de test.	Chaîne	*SampleIma geBuilder InstanceP rofileRole*
platform	La plate-forme du système d'exploitation de l'image créée.	Chaîne	`Linux` | `Windows` | `MacOS`

Nom de variable	Description	Type	Exemple de valeur
Journaux S3	Objet JSON contenant la configuration des journaux S3 écrits par Image Builder.	Objet JSON	`{'s3logs' : {'s3':BucketName' sample-bucket', 's3KeyPrefix' : 'ib-logs'}}`
securityGroups	Les identifiants des groupes de sécurité qui s'appliquent à la création et au test des instances.	Liste [Chaîne]	`[sg-1234567890abcdef1, sg-11112222333344445]`
Source ImageLearn	Le nom de ressource Amazon (ARN) de la ressource d'image Image Builder que le flux de travail utilise pour les étapes de création et de test.	Chaîne	`arn:aws:imagebuilder : us-east-1 : 111122223333:image/ SampleImage/1.0.0/1`
subnetId	ID du sous-réseau dans lequel lancer les instances de génération et de test.	Chaîne	`sous-net-1234567890abcdef1`

Nom de variable	Description	Type	Exemple de valeur
terminateInstanceOnDéfaillance	La valeur actuelle du paramètre qui indique à Image Builder de mettre fin à l'instance en cas de défaillance ou de la conserver à des fins de dépannage.	Booléen	`true` | `false`
Phase du flux de travail	L'étape en cours d'exécution pour l'exécution du flux de travail.	Chaîne	`Build` | `Test`
workingDirectory	Le chemin d'accès au répertoire de travail.	Chaîne	`/tmp`

Utilisez des instructions conditionnelles dans les étapes de votre flux de travail

Les instructions conditionnelles commencent par l'attribut du document de `if` déclaration. Le but ultime de l'`if`instruction est de déterminer s'il faut exécuter l'action de l'étape ou l'ignorer. Si l'`if`instruction est résolue par`true`, l'action de l'étape est exécutée. Si le problème est résolu`false`, Image Builder ignore l'action de l'étape et enregistre le statut de l'étape `SKIPPED` dans le journal.

L'`if`instruction prend en charge les instructions de branchement (`and`,`or`) et les modificateurs conditionnels (`not`). Il prend également en charge les instructions conditionnelles suivantes qui effectuent des comparaisons de valeurs (égal, inférieur à, supérieur à) en fonction des types de données qu'il compare (chaîne ou nombre).

Déclarations conditionnelles prises en charge

- `booleanEquals`
- `numberEquals`

- `numberGreaterThan`
- `numberGreaterThanEquals`
- `numberLessThan`
- `numberLessThanEquals`
- `stringEquals`

Règles relatives aux instructions de branchement et aux modificateurs conditionnels

Les règles suivantes s'appliquent aux instructions de branchement (`and`,`or`) et aux modificateurs conditionnels (`not`).

- Les instructions de branchement et les modificateurs conditionnels doivent apparaître seuls sur une ligne.
- Les instructions de branchement et les modificateurs conditionnels doivent respecter les règles de niveau.
  - Il ne peut y avoir qu'une seule déclaration au niveau du parent.
  - Chaque branche ou modificateur enfant commence un nouveau niveau.

  Pour plus d'informations sur les niveaux, consultezNiveaux imbriqués.
- Chaque instruction secondaire doit comporter au moins une instruction conditionnelle secondaire, mais pas plus de dix.
- Les modificateurs conditionnels ne fonctionnent que sur une seule instruction conditionnelle enfant.

Niveaux imbriqués

Les déclarations conditionnelles fonctionnent à plusieurs niveaux dans une section qui leur est propre. Par exemple, l'attribut `if` statement apparaît au même niveau dans votre document de flux de travail que le nom et l'action de l'étape. Il s'agit de la base de l'énoncé conditionnel.

Vous pouvez spécifier jusqu'à quatre niveaux d'instructions conditionnelles, mais une seule instruction peut apparaître au niveau parent. Toutes les autres instructions de branchement, modificateurs conditionnels ou opérateurs conditionnels sont indentés à partir de là, un retrait par niveau.

Le schéma suivant indique le nombre maximal de niveaux imbriqués pour une instruction conditionnelle.

```
base:
  parent:
    - child (level 2)
        - child (level 3)
            child (level 4)
```

Attribut `if`

L'`if`attribut spécifie l'instruction conditionnelle en tant qu'attribut de document. C'est le niveau zéro.

Niveau parental

Il s'agit du premier niveau d'imbrication pour les instructions conditionnelles. Il ne peut y avoir qu'une seule déclaration à ce niveau. Si vous n'avez pas besoin de branchement ou de modificateurs, il peut s'agir d'un opérateur conditionnel sans instructions enfant. Ce niveau n'utilise pas la notation en tirets, sauf pour les opérateurs conditionnels.

Niveaux pour enfants

Les niveaux 2 à 4 sont considérés comme des niveaux pour enfants. Les instructions enfant peuvent inclure des instructions de branchement, des modificateurs conditionnels ou des opérateurs conditionnels.

Exemple : niveaux imbriqués

L'exemple suivant montre le nombre maximal de niveaux dans une instruction conditionnelle.

```
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"
```

Règles de nidification

- Chaque branche ou modificateur au niveau de l'enfant ouvre un nouveau niveau.

- Chaque niveau est indenté.
- Il peut y avoir un maximum de quatre niveaux, dont une instruction, un modificateur ou un opérateur au niveau parent, et jusqu'à trois niveaux supplémentaires.

Exemples

Ce groupe d'exemples montre différents aspects des déclarations conditionnelles.

Branchement : et

L'instruction de `and` branchement fonctionne sur une liste d'expressions qui sont des enfants de la branche, qui doivent toutes être évaluées à`true`. Image Builder évalue les expressions dans l'ordre dans lequel elles apparaissent dans la liste. Si une expression est évaluée à`false`, le traitement s'arrête et la branche est prise en compte`false`.

L'exemple suivant évalue à`true`, car les deux expressions sont évaluées à`true`.

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

Branchement : ou

L'instruction de `or` branchement fonctionne sur une liste d'expressions qui sont des enfants de la branche, dont au moins l'une doit être évaluée à`true`. Image Builder évalue les expressions dans l'ordre dans lequel elles apparaissent dans la liste. Si une expression est évaluée à`true`, le traitement s'arrête et la branche est prise en compte`true`.

L'exemple suivant évalue à`true`, même si la première expression est`false`.

```
if:
  or:
    - stringEquals: 'test_string'
      value: 'test_string_not_equal'
    - numberEquals: 1
      value: 1
```

Modificateur conditionnel : non

Le modificateur `not` conditionnel annule les instructions conditionnelles qui sont les enfants de la branche.

L'exemple suivant indique `true` quand le `not` modificateur annule l'instruction `stringEquals` conditionnelle.

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

Déclaration conditionnelle : BooleanEquals

L'instruction `booleanEquals` conditionnelle compare les valeurs booléennes et renvoie la valeur true si les valeurs booléennes correspondent exactement.

L'exemple suivant détermine si cette option `collectImageScanFindings` est activée.

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

Déclaration conditionnelle : StringEquals

L'instruction `stringEquals` conditionnelle compare deux chaînes et renvoie la valeur true si les chaînes correspondent exactement. Si l'une des valeurs n'est pas une chaîne, Image Builder la convertit en chaîne avant de procéder à la comparaison.

L'exemple suivant compare la variable système de plate-forme pour déterminer si le flux de travail s'exécute sur une plate-forme Linux.

```
if:
  - stringEquals: 'Linux'
    value: '$.imagebuilder.Platform'
```

Déclaration conditionnelle : NumberEquals

L'instruction `numberEquals` conditionnelle compare deux nombres et renvoie la valeur true s'ils sont égaux. Les nombres à comparer doivent être dans l'un des formats suivants.

- Entier

- Float
- Une chaîne qui correspond au modèle d'expression régulière suivant :`^-?[0-9]+(\.)?[0-9]+$`.

Les exemples de comparaison suivants sont tous évalués à. `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
```

Importez et exportez des images de machines virtuelles (VM) avec EC2 Image Builder

Lorsque vous exportez votre machine virtuelle depuis son environnement de virtualisation, ce processus crée un ensemble d'un ou plusieurs fichiers de conteneur de disque qui agissent comme des instantanés de l'environnement, des paramètres et des données de votre machine virtuelle. Vous pouvez utiliser ces fichiers pour importer votre machine virtuelle et l'utiliser comme image de base pour vos recettes d'images.

Image Builder prend en charge les formats de fichiers suivants pour les conteneurs de disques de votre machine virtuelle :

- Archive de virtualisation ouverte (OVA)

- Disque de machine virtuelle (VMDK)
- Disque dur virtuel (VHD/VHDX)
- Raw

L'importation utilise les disques pour créer une image Amazon Machine Image (AMI) et une ressource d'image Image Builder, l'une ou l'autre pouvant servir d'image de base pour votre recette d'image personnalisée. Les disques de machine virtuelle doivent être stockés dans des compartiments S3 pour l'importation. Vous pouvez également effectuer une importation à partir d'un instantané EBS existant.

Dans la console Image Builder, vous pouvez importer l'image directement, puis utiliser l'image de sortie ou l'AMI dans vos recettes, ou vous pouvez spécifier des paramètres d'importation lorsque vous créez votre recette ou version de recette. Pour plus d'informations sur l'importation directe, consultezImporter une machine virtuelle (console). Pour plus d'informations sur l'importation dans le cadre de votre recette d'image, consultezConfiguration de l'importation de machines virtuelles.

Importer une machine virtuelle dans Image Builder (AWS CLI)

Pour importer une machine virtuelle depuis des disques vers une AMI et créer une ressource d'image Image Builder à laquelle vous pouvez immédiatement faire référence, procédez comme suit à partir du AWS CLI :

1. Lancez une importation de machine virtuelle à l'aide de la commande Amazon EC2 VM Import/Export import-image dans le. AWS CLI Notez l'ID de tâche renvoyé dans la réponse à la commande. Vous en aurez besoin pour la prochaine étape. Pour plus d'informations, consultez la section Importation d'une machine virtuelle sous forme d'image à l'aide de VM Import/Export dans le guide de l'utilisateur de VM Import/Export.
2. Créer un fichier JSON d'entrée CLI

   Pour rationaliser la import-vm-image commande Image Builder utilisée dans le AWS CLI, nous créons un fichier JSON contenant toutes les configurations d'importation que nous voulons transmettre à la commande.

   > **Note**
   >
   > La convention de dénomination des valeurs de données dans le fichier JSON suit le modèle spécifié pour les paramètres de demande d'action de l'API Image Builder.

Pour consulter les paramètres de demande de commande d'API, consultez la ImportVmImagecommande dans le manuel de référence de l'API EC2 Image Builder. Pour fournir les valeurs de données sous forme de paramètres de ligne de commande, reportez-vous aux noms de paramètres spécifiés dans la AWS CLI Command Reference. À la import-vm-image commande Image Builder en tant qu'options.

Voici un résumé des paramètres que nous indiquons dans cet exemple :

- name (chaîne, obligatoire) : nom de la ressource d'image Image Builder à créer en sortie de l'importation.
- SemanticVersion <major>(chaîne, obligatoire) — Version sémantique de l'image de sortie qui spécifie la version au format suivant, avec des valeurs numériques à chaque position pour indiquer une version spécifique :. <minor>. <patch>. Par exemple, `1.0.0`. Pour en savoir plus sur le versionnement sémantique des ressources Image Builder, consultez. Gestion des versions sémantique
- description (chaîne) — Description de la recette d'image.
- platform (string, obligatoire) : plate-forme du système d'exploitation de la machine virtuelle importée.
- vmImportTaskId (chaîne, obligatoire) — Le `ImportTaskId` (AWS CLI) issu du processus d'importation de la machine virtuelle Amazon EC2. Image Builder surveille le processus d'importation pour intégrer l'AMI qu'il crée et créer une ressource d'image Image Builder qui peut être utilisée immédiatement dans des recettes.
- ClientToken (chaîne, obligatoire) — Identifiant unique distinguant majuscules et minuscules que vous fournissez pour garantir l'idempuissance de la demande. Pour plus d'informations, consultez la section Garantir l'idempotencie dans le manuel Amazon EC2 API Reference.
- tags (chaîne de caractères) — Les tags sont des paires clé-valeur associées aux ressources d'importation. Jusqu'à 50 paires clé-valeur sont autorisées.

Enregistrez le fichier sous`import-vm-image.json`, pour l'utiliser dans la import-vm-image commande 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. Importer l'image

   Exécutez la import-vm-image commande en utilisant le fichier que vous avez créé en entrée :

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

> **Note**
>
> - Vous devez inclure l'option `file://` au début du chemin du fichier JSON.
> - Le chemin d'accès du fichier JSON doit suivre la convention appropriée pour le système d'exploitation de base sur lequel vous exécutez la commande. En effet, Windows utilise la barre oblique inverse (\) pour faire référence au chemin du répertoire, et Linux utilise la barre oblique directe (/).

Distribuez des disques de machine virtuelle à partir de votre version d'image (AWS CLI)

Vous pouvez configurer la distribution des fichiers au format de disque de machine virtuelle pris en charge dans les compartiments S3 des régions cibles dans le cadre de votre processus de création d'image habituel, à l'aide des configurations de distribution d'Image Builder dans le AWS CLI. Pour plus d'informations, voir Création de paramètres de distribution pour les disques de machine virtuelle de sortie (AWS CLI).

Partagez les ressources d'EC2 Image Builder

EC2 Image Builder s' AWS Resource Access Manager intègre àAWS RAM() pour vous permettre de partager certaines ressources avec Compte AWS n'importe qui ou AWS Organizations via. Les ressources EC2 Image Builder qui peuvent être partagées sont les suivantes :

- Composants
- Images
- Recettes

Cette section fournit des informations pour vous aider à partager ces ressources EC2 Image Builder.

Contenu de la section

Utilisation de composants, d'images et de recettes partagés dans EC2 Image Builder

Le partage de composants, d'images et de recettes permet aux propriétaires de ressources de partager des configurations logicielles avec d'autres personnes Comptes AWS ou au sein d'une AWS organisation. Vous pouvez gérer le partage des ressources de manière centralisée et définir un ensemble de comptes avec lesquels la configuration peut être partagée.

Dans ce modèle, le Compte AWS propriétaire du composant, de l'image ou de la recette (propriétaires) le partage avec d'autres Comptes AWS (consommateurs). Les utilisateurs peuvent associer un composant partagé à leurs pipelines d'images afin de consommer automatiquement les mises à jour du composant, de l'image ou de la recette partagé.

Le propriétaire d'un composant, d'une image ou d'une recette peut partager ces ressources avec :

- Spécifique Comptes AWS à l'intérieur ou à l'extérieur de son organisation dans AWS Organizations.

- Une unité organisationnelle (UO) au sein de son organisation dans AWS Organizations.
- L'ensemble de son organisation dans AWS Organizations.
- AWS Organizations ou des unités d'organisation extérieures à son organisation.

Conditions préalables au partage de composants, d'images et de recettes

Pour partager un composant, une image ou une recette d'Image Builder :

- Vous devez être propriétaire du composant, de l'image ou de la recette de votre Compte AWS. Vous ne pouvez pas partager les ressources qui ont été partagées avec vous.
- La clé AWS Key Management Service (AWS KMS) associée aux ressources chiffrées doit être explicitement partagée avec les comptes, organisations ou unités d'organisation cibles.
- Pour partager vos ressources Image Builder avec des unités d'organisation à AWS Organizations l'aide de AWS RAM celles-ci, vous devez activer le partage. Pour plus d'informations, consultez Activation du partage avec AWS Organizations dans le Guide de l'utilisateur AWS RAM .
- Si vous distribuez une image chiffrée AWS KMS sur plusieurs comptes dans différentes régions, vous devez créer une clé KMS et un alias dans chaque région cible. En outre, les personnes qui lanceront des instances dans ces régions devront avoir accès à la clé KMS spécifiée dans la politique clé.

Les ressources suivantes créées par Image Builder à partir de votre pipeline ne sont pas considérées comme des ressources Image Builder. Il s'agit plutôt de ressources externes qu'Image Builder distribue dans votre compte, ainsi qu'aux comptes et aux organisations ou unités organisationnelles (UO) que vous spécifiez dans votre configuration de distribution. Régions AWS

- Amazon Machine Images (AMI)
- Images de conteneurs résidant dans Amazon ECR

Pour plus d'informations sur les paramètres de distribution de votre AMI, consultezCréation et mise à jour des configurations de distribution d'AMI. Pour plus d'informations sur les paramètres de distribution de votre image de conteneur dans Amazon ECR, consultezCréation et mise à jour des paramètres de distribution pour les images de conteneurs.

Pour plus d'informations sur le partage de votre AMI avec AWS Organizations et des unités d'organisation, voir Partager une AMI avec des organisations ou des unités d'organisation.

Services connexes

AWS Resource Access Manager

Le partage de composants, d'images et de recettes s'intègre à AWS Resource Access Manager (AWS RAM). AWS RAM est un service qui vous permet de partager vos AWS ressources avec n'importe quel AWS compte ou via AWS Organizations. Avec AWS RAM, vous partagez les ressources que vous possédez en créant un partage de ressources. Un partage de ressources indique les ressources à partager et les consommateurs avec qui les partager. Les consommateurs peuvent être des individus Comptes AWS, des unités organisationnelles ou une organisation entière AWS Organizations.

Pour plus d'informations AWS RAM, consultez le guide de AWS RAM l'utilisateur.

Partage entre les régions

Les composants, images et recettes partagés ne peuvent être partagés que dans une AWS région spécifiée. Lorsque vous partagez ces ressources, elles ne seront pas répliquées d'une région à l'autre.

Partage d'un composant, d'une image ou d'une recette

Pour partager un composant, une image ou une recette Image Builder, vous devez l'ajouter à un partage de ressources. Un partage de ressources est une AWS RAM ressource qui vous permet de partager vos ressources entre différents AWS comptes. Un partage de ressources indique les ressources à partager et les consommateurs avec lesquels elles sont partagées. Pour ajouter le composant, l'image ou la recette à un nouveau partage de ressources, vous devez d'abord créer le partage de ressources à l'aide de la AWS RAM console.

Si vous faites partie d'une organisation AWS Organizations et que le partage au sein de votre organisation est activé, les consommateurs de votre organisation ont automatiquement accès au composant, à l'image ou à la recette partagés. Dans le cas contraire, les consommateurs reçoivent une invitation à rejoindre le partage de ressources et ont accès à la ressource partagée après avoir accepté l'invitation.

Les options suivantes sont disponibles pour partager vos ressources :.

Option 1 : créer un partage de ressources RAM

Lorsque vous créez un partage de ressources RAM, vous pouvez partager un composant, une image ou une recette que vous possédez en une seule étape. Utilisez l'une des méthodes suivantes pour créer votre partage de ressources :

- Console

  Pour créer votre partage de ressources à l'aide de la AWS RAM console, voir Partager AWS les ressources qui vous appartiennent dans le guide de AWS RAM l'utilisateur.

- AWS CLI

  Pour créer votre partage de ressources à l'aide de l'interface de ligne de AWS RAM commande, exécutez la create-resource-share commande dans le AWS CLI.

Option 2 : appliquer une politique de ressources et passer à un partage de ressources RAM

La deuxième option pour partager vos ressources implique deux étapes, en exécutant des commandes dans AWS CLI les deux cas. La première étape utilise les commandes Image Builder AWS CLI pour appliquer des politiques basées sur les ressources à la ressource partagée. La deuxième étape fait passer la ressource à un partage de ressources RAM à l'aide de la promote-resource-share-created-from-policy AWS RAM commande contenue dans le AWS CLI afin de garantir que la ressource est visible par tous les principaux avec lesquels vous l'avez partagée.

1. Appliquer la politique en matière de ressources

   Pour appliquer correctement la politique de ressources, vous devez vous assurer que le compte avec lequel vous partagez est autorisé à accéder à toutes les ressources sous-jacentes.

   Choisissez l'onglet correspondant à votre type de ressource pour la commande applicable.

   Image

   Vous pouvez appliquer une politique de ressources à une image afin de permettre à d'autres utilisateurs de l'utiliser comme image de base dans leurs recettes.

   Exécutez la commande put-image-policy Image Builder dans le AWS CLI, pour identifier les AWS principaux acteurs avec lesquels partager l'image.

```
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

Vous pouvez appliquer une politique de ressources à un composant de génération ou de test pour permettre le partage entre comptes. Cette commande autorise les autres comptes à utiliser votre composant dans leurs recettes. Pour appliquer correctement la politique de ressources, vous devez vous assurer que le compte avec lequel vous partagez est autorisé à accéder à toutes les ressources référencées par le composant partagé, telles que les fichiers hébergés sur des référentiels privés.

Exécutez la commande put-component-policy Image Builder dans le AWS CLI, pour identifier les AWS principaux acteurs avec lesquels partager le composant.

```
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

Vous pouvez appliquer une politique de ressources à une recette d'image pour permettre le partage entre comptes. Cette commande autorise les autres comptes à utiliser votre recette pour créer des images dans leurs comptes. Pour appliquer correctement la politique de ressources, vous devez vous assurer que le compte avec lequel vous partagez est autorisé à accéder à toutes les ressources auxquelles la recette fait référence, telles que l'image de base ou les composants sélectionnés.

Exécutez la commande put-image-recipe-policy Image Builder dans le AWS CLI, pour identifier les AWS principaux acteurs avec lesquels partager l'image.

```
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

Vous pouvez appliquer une politique de ressources à une recette de conteneur pour permettre le partage entre comptes. Cette commande autorise les autres comptes à utiliser votre recette pour créer des images dans leurs comptes. Pour appliquer correctement la politique de ressources, vous devez vous assurer que le compte avec lequel vous partagez est autorisé à accéder à toutes les ressources auxquelles la recette fait référence, telles que l'image de base ou les composants sélectionnés.

Exécutez la commande put-container-recipe-policy Image Builder dans le AWS CLI, pour identifier les AWS principaux acteurs avec lesquels partager l'image.

```
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**
>
> Pour définir les bonnes politiques de partage et de non-partage d'une ressource, le propriétaire de la ressource doit disposer des `imagebuilder:put*` autorisations nécessaires.

2. Promouvez en tant que partage de ressources RAM

   Pour que la ressource soit visible par tous les principaux avec lesquels vous l'avez partagée, exécutez la promote-resource-share-created-from-policy AWS RAM commande dans le AWS CLI.

Annulation du partage d'un composant, d'une image ou d'une recette

Pour annuler le partage d'un composant, d'une image ou d'une recette que vous possédez, vous devez le supprimer du partage de ressources. Vous pouvez le faire à l'aide de la AWS Resource Access Manager console ou du AWS CLI.

> **Note**
>
> Pour annuler le partage d'un composant, d'une image ou d'une recette, le consommateur ne peut avoir aucune dépendance à leur égard. Le consommateur doit supprimer toute dépendance vis-à-vis des ressources partagées avant que le propriétaire ne puisse annuler leur partage.

Pour annuler le partage d'un composant, d'une image ou d'une recette partagés dont vous êtes propriétaire à l'aide de la console AWS Resource Access Manager

Consultez Mise à jour d'un partage de ressources dans le Guide de l'utilisateur AWS RAM .

Pour annuler le partage d'un composant, d'une image ou d'une recette partagés dont vous êtes propriétaire à l'aide du AWS CLI

Utilisez la disassociate-resource-share commande pour arrêter de partager la ressource.

Identification d'un composant, d'une image ou d'une recette partagés

Les propriétaires et les consommateurs peuvent identifier les composants, les images et les recettes d'images partagés à l'aide des commandes Image Builder dans le AWS CLI.

Identifier un composant partagé

Exécutez la commande list-components pour obtenir une liste des composants que vous possédez et des composants partagés avec vous. La commande get-component indique l' Compte AWS ID du propriétaire du composant.

Identifier une image partagée

Exécutez la commande list-images pour obtenir une liste des images que vous possédez et des images partagées avec vous. La commande get-image affiche l' Compte AWS ID du propriétaire de l'image.

Identifier une image de conteneur partagée

Exécutez la commande list-images pour obtenir une liste des images que vous possédez et des images partagées avec vous. La commande get-image affiche l' Compte AWS ID du propriétaire de l'image.

Identifier une recette d'image partagée

Exécutez la list-image-recipescommande pour obtenir une liste des recettes d'images que vous possédez et des recettes d'images partagées avec vous. La get-image-recipecommande indique l' Compte AWS ID du propriétaire de la recette d'image.

Identifier une recette de conteneur partagée

Exécutez la list-container-recipescommande pour obtenir une liste des recettes de conteneurs que vous possédez et des recettes de conteneurs partagées avec vous. La get-container-recipecommande indique l' Compte AWS ID du propriétaire de la recette du conteneur.

Autorisations partagées pour les composants, les images et les recettes

Autorisations accordées aux propriétaires

Les propriétaires ne peuvent pas supprimer un composant, une image ou une recette d'image partagés tant qu'ils ne sont plus partagés. Un propriétaire ne peut pas annuler le partage de ces ressources tant qu'aucun consommateur n'en dépend.

Autorisations accordées aux consommateurs

Les consommateurs peuvent lire un composant, une image ou une recette d'image, mais ne peuvent en aucun cas les modifier. Ils ne peuvent pas consulter ou modifier ces ressources si elles appartiennent à d'autres consommateurs ou au propriétaire de la ressource. Les consommateurs peuvent utiliser des composants et des images partagés dans des recettes d'images pour créer des images personnalisées. Les consommateurs peuvent utiliser des recettes d'images partagées pour créer leurs propres images personnalisées.

Facturation et mesures

L'utilisation d'EC2 Image Builder est gratuite.

Limites des ressources

Les composants, images et recettes d'images partagés ne sont pris en compte que dans les limites de ressources correspondantes du propriétaire. Les limites de ressources des consommateurs ne sont pas affectées par les ressources qui ont été partagées avec eux.

Étiqueter les ressources d'EC2 Image Builder

Le balisage de vos ressources peut être utile pour filtrer et suivre les coûts des ressources, ou pour d'autres catégories. Vous pouvez également contrôler l'accès en fonction des balises. Pour plus d'informations sur l'autorisation basée sur des balises, voir Autorisation basée sur les balises Image Builder

Image Builder prend en charge les balises dynamiques suivantes :

- `- {{imagebuilder:buildDate}}`

  Corrige à la date/heure de construction au moment de la construction.
- `- {{imagebuilder:buildVersion}}`

  Résout à une version de build, qui est un numéro situé à la fin du nom de ressource Amazon (ARN) d'Image Builder. Par exemple, `"arn:aws:imagebuilder:us-west-2:123456789012:component/myexample-component/2019.12.02/1"` affiche la version de compilation sous la forme`1`.

Pour vous aider à suivre les Amazon Machine Images (AMI) que vous avez distribuées, Image Builder ajoute automatiquement les balises suivantes à vos AMI de sortie.

- `"CreatedBy":"EC2 Image Builder"`
- `"Ec2ImageBuilderArn":"arn:aws:imagebuilder:`*`us-west-2`*`:`*`123456789012`*`:image/`*`simple-recipe-linux`*`/`*`1.0.0/10`*`"`. Cette balise contient l'ARN de la ressource d'image Image Builder qui a été utilisée pour créer l'AMI.

Table des matières

Marquer une ressource (AWS CLI)

L'exemple suivant montre comment utiliser une commande imagebuilder CLI pour ajouter et étiqueter une ressource dans EC2 Image Builder. Vous devez fournir les balises `resourceArn` et à appliquer.

Le `tag-resource.json` contenu de l'exemple est le suivant :

```
{
    "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/my-example-pipeline",
    "tags": {
        "KeyName": "KeyValue"
    }
}
```

Exécutez la commande suivante, qui fait référence au `tag-resource.json` fichier précédent.

```
aws imagebuilder tag-resource --cli-input-json file://tag-resource.json
```

Supprimer le tag d'une ressource ()AWS CLI

L'exemple suivant montre comment utiliser une commande imagebuilder CLI pour supprimer une balise d'une ressource. Vous devez fournir les clés `resourceArn` et pour retirer le tag.

Le `untag-resource.json` contenu de l'exemple est le suivant :

```
{
    "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/my-example-pipeline",
    "tagKeys": [
        "KeyName"
    ]
}
```

Exécutez la commande suivante, qui fait référence au `untag-resource.json` fichier précédent.

```
aws imagebuilder untag-resource --cli-input-json file://untag-resource.json
```

Répertorier tous les tags d'une ressource spécifique (AWS CLI)

L'exemple suivant montre comment utiliser une commande imagebuilder CLI pour répertorier toutes les balises d'une ressource spécifique.

```
aws imagebuilder list-tags-for-resource --resource-arn arn:aws:imagebuilder:us-
west-2:123456789012:image-pipeline/my-example-pipeline
```

Supprimer les ressources EC2 Image Builder

Votre environnement Image Builder, tout comme votre maison, a besoin d'un entretien régulier pour vous aider à trouver ce dont vous avez besoin et à accomplir vos tâches sans vous encombrer. Assurez-vous de nettoyer régulièrement les ressources temporaires que vous avez créées pour les tests. Sinon, vous pourriez oublier ces ressources et, plus tard, ne plus vous souvenir de leur utilisation. D'ici là, il se peut que vous ne sachiez pas si vous pouvez vous en débarrasser en toute sécurité.

La suppression de ressources ne supprime pas les AMI Amazon EC2 ou les images de conteneur Amazon ECR créées pendant le processus de création de l'image. Vous devez les nettoyer séparément, à l'aide des actions de console Amazon EC2 ou Amazon ECR appropriées, de l'API ou des commandes. AWS CLI

> **Tip**
>
> Pour éviter les erreurs de dépendance lorsque vous supprimez des ressources, veillez à supprimer vos ressources dans l'ordre suivant :
>
> 1. Pipeline d'images
> 2. Recette d'images
> 3. Toutes les ressources restantes

Supprimer des ressources à l'aide de la console AWS de gestion

Pour supprimer un pipeline d'images et ses ressources, procédez comme suit :

Supprimer le pipeline

1. Pour voir la liste des pipelines de génération créés sous votre compte, choisissez Image pipelines dans le volet de navigation.
2. Cochez la case à côté du nom du pipeline pour sélectionner le pipeline que vous souhaitez supprimer.
3. En haut du panneau Pipelines d'images, dans le menu Actions, choisissez Supprimer.
4. Pour confirmer la suppression, entrez `Delete` dans le champ et choisissez Supprimer.

Supprimer la recette

1. Pour voir la liste des recettes créées sous votre compte, choisissez Image recipes dans le volet de navigation.
2. Cochez la case à côté du nom de la recette pour sélectionner la recette que vous souhaitez supprimer.
3. En haut du panneau Recettes d'images, dans le menu Actions, choisissez Supprimer la recette.
4. Pour confirmer la suppression, entrez `Delete` dans le champ et choisissez Supprimer.

Supprimer la configuration de l'infrastructure

1. Pour consulter la liste des configurations d'infrastructure créées sous votre compte, choisissez Configuration de l'infrastructure dans le volet de navigation.
2. Cochez la case à côté du nom de la configuration pour sélectionner la configuration d'infrastructure que vous souhaitez supprimer.
3. En haut du panneau de configuration de l'infrastructure, choisissez Supprimer.
4. Pour confirmer la suppression, entrez `Delete` dans le champ et choisissez Supprimer.

Supprimer les paramètres de distribution

1. Pour voir la liste des paramètres de distribution créés sous votre compte, choisissez Paramètres de distribution dans le volet de navigation.
2. Cochez la case à côté du nom de la configuration pour sélectionner les paramètres de distribution que vous avez créés pour ce didacticiel.
3. En haut du panneau des paramètres de distribution, choisissez Supprimer.

4. Pour confirmer la suppression, entrez `Delete` dans le champ et choisissez Supprimer.

Supprime une image

1. Pour voir la liste des images créées sous votre compte, choisissez Images dans le volet de navigation.
2. Choisissez la version de l'image que vous souhaitez supprimer. Cela ouvre la page des versions de génération d'images.
3. Cochez la case à côté de la version pour toute image que vous souhaitez supprimer. Vous pouvez sélectionner plusieurs versions d'image à la fois.
4. En haut du panneau des versions de génération d'images, choisissez Supprimer la version.
5. Pour confirmer la suppression, entrez `Delete` dans le champ et choisissez Supprimer.

Supprimez un pipeline d'images à l'aide du AWS CLI

Les exemples suivants montrent comment supprimer des ressources Image Builder à l'aide du AWS CLI. Comme indiqué précédemment, les ressources doivent être supprimées dans l'ordre suivant pour éviter les erreurs de dépendance :

1. Pipeline d'images
2. Recette d'images
3. Toutes les ressources restantes

Supprimer le pipeline d'images (AWS CLI)

L'exemple suivant montre comment supprimer un pipeline d'images en spécifiant son ARN.

```
aws imagebuilder delete-image-pipeline --image-pipeline-arn arn:aws:imagebuilder:us-
west-2:123456789012:image-pipeline/my-example-pipeline
```

Supprimer la recette d'image (AWS CLI)

L'exemple suivant montre comment supprimer une recette d'image en spécifiant son ARN.

```
aws imagebuilder delete-image-recipe --image-recipe-arn arn:aws:imagebuilder:us-
west-2:123456789012:image-recipe/my-example-recipe/2019.12.03
```

Supprime une configuration d'infrastructure

L'exemple suivant montre comment supprimer une ressource de configuration d'infrastructure en spécifiant son ARN.

```
aws imagebuilder delete-infrastructure-configuration --infrastructure-configuration-arn
 arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/my-example-
infrastructure-configuration
```

Supprimer les paramètres de distribution

L'exemple suivant montre comment supprimer une ressource de paramètres de distribution en spécifiant son ARN.

```
aws imagebuilder delete-distribution-configuration --distribution-configuration-arn
 arn:aws:imagebuilder:us-west-2:123456789012:distribution-configuration/my-example-
distribution-configuration
```

Supprime une image

L'exemple suivant montre comment supprimer une version de génération d'image en spécifiant son ARN.

```
aws imagebuilder delete-image --image-build-version-arn arn:aws:imagebuilder:us-
west-2:123456789012:image/my-example-image/2019.12.02/1
```

Supprime un composant

L'exemple suivant montre comment utiliser une commande imagebuilder CLI pour supprimer une version de construction d'un composant en spécifiant son 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**
>
> Assurez-vous qu'aucune recette ne fait référence à la version de construction du composant de quelque manière que ce soit avant de la supprimer. Ne pas le faire pourrait entraîner des défaillances du pipeline.

Gérez les pipelines EC2 Image Builder à l'aide de la console

Les pipelines d'images Image Builder fournissent un cadre d'automatisation permettant de créer et de gérer des AMI et des images de conteneur personnalisées. Les pipelines fournissent les fonctionnalités suivantes :

- Assemblez l'image de base, les composants à créer et à tester, la configuration de l'infrastructure et les paramètres de distribution.
- Facilitez la planification des processus de maintenance automatisés `Schedule builder` à l'aide de l'assistant intégré à la console ou en saisissant des expressions cron pour les mises à jour récurrentes de vos images.
- Activez la détection des modifications pour l'image de base et les composants, afin d'ignorer automatiquement les builds planifiés en l'absence de modifications.
- Activez l'automatisation basée sur des règles via Amazon. EventBridge

> **Note**
>
> Pour plus d'informations sur l'utilisation de l' EventBridge API pour consulter ou modifier les règles, consultez le Amazon EventBridge API Reference. Pour plus d'informations sur l'utilisation EventBridge events des commandes dans les règles AWS CLI d'affichage ou de modification, voir les événements dans le manuel de référence des AWS CLI commandes.

Table des matières

Répertorier et afficher les détails du pipeline

Cette section décrit les différentes manières de trouver des informations et d'afficher les détails de vos pipelines d'images EC2 Image Builder.

Détails du pipeline

- Répertorier les pipelines d'images (AWS CLI)
- Obtenir les détails du pipeline d'images (AWS CLI)

Répertorier les pipelines d'images (AWS CLI)

L'exemple suivant montre comment utiliser la list-image-pipelines commande figurant dans la liste AWS CLI de tous vos pipelines d'images.

```
aws imagebuilder list-image-pipelines
```

Obtenir les détails du pipeline d'images (AWS CLI)

L'exemple suivant montre comment utiliser la get-image-pipeline commande du AWS CLI pour obtenir des informations sur un pipeline d'images via son ARN.

```
aws imagebuilder get-image-pipeline --image-pipeline-arn arn:aws:imagebuilder:us-
west-2:123456789012:image-pipeline/my-example-pipeline
```

Création et mise à jour de pipelines d'images AMI

Vous pouvez configurer et gérer les pipelines d'images AMI depuis la console Image Builder, via l'API Image Builder ou à l'aide des imagebuilder commandes du AWS CLI. Vous pouvez utiliser l'assistant de création de pipeline d'images pour vous guider dans les étapes suivantes :

- Spécifiez les détails du pipeline tels que le nom, la description et les balises de ressources.
- Sélectionnez une recette d'image AMI qui inclut une image de base provenant d'images gérées au démarrage rapide ou d'images que vous avez créées ou partagées avec vous. La recette inclut également des composants qui exécutent les tâches suivantes sur les instances EC2 utilisées par Image Builder pour créer votre image :
  - Ajouter et supprimer des logiciels

  - Personnaliser les paramètres et les scripts
  - Exécuter les tests sélectionnés
- Spécifiez des flux de travail pour configurer la création d'images et les étapes de test exécutées par votre pipeline.
- Définissez la configuration de l'infrastructure de votre pipeline à l'aide de paramètres par défaut ou de paramètres que vous configurez vous-même. La configuration inclut le type d'instance et la paire de clés à utiliser pour votre image, les paramètres de sécurité et de réseau, les paramètres de stockage des journaux et de résolution des problèmes, ainsi que les notifications SNS.

  Il s'agit d'une étape facultative. Image Builder utilise les paramètres par défaut pour la configuration de votre infrastructure si vous ne définissez pas la configuration vous-même.
- Définissez les paramètres de distribution pour envoyer vos images vers les AWS régions et les comptes de destination. Vous pouvez spécifier une clé KMS pour le chiffrement, configurer le partage d'AMI ou la configuration des licences, ou configurer un modèle de lancement pour les AMI que vous distribuez.

  Il s'agit d'une étape facultative. Si vous ne définissez pas vous-même la configuration, Image Builder utilise le nom par défaut pour votre AMI de sortie et distribue l'AMI dans la région source. La région source est la région dans laquelle vous exécutez le pipeline.

Pour plus d'informations et un step-by-step didacticiel sur l'utilisation de l'assistant de création de pipeline d'images avec les valeurs par défaut fournies, voirCréez un pipeline d'images à l'aide de l'assistant de console EC2 Image Builder.

Table des matières

Création d'un pipeline d'images AMI (AWS CLI)

Vous pouvez créer un pipeline d'images AMI avec un fichier JSON contenant les détails de configuration en entrée de la create-image-pipeline commande dans le AWS CLI.

La fréquence à laquelle votre pipeline crée une nouvelle image pour intégrer les mises à jour en attente à partir de votre image de base et de vos composants dépend de `schedule` celle que vous avez configurée. A `schedule` possède les attributs suivants :

- `scheduleExpression`— Définit le calendrier d'exécution de votre pipeline afin d'évaluer `pipelineExecutionStartCondition` et de déterminer s'il doit démarrer une construction. Le planning est configuré avec des expressions cron. Pour plus d'informations sur le formatage d'une expression cron dans Image Builder, consultezUtiliser des expressions cron dans EC2 Image Builder.
- `pipelineExecutionStartCondition`— Détermine si votre pipeline doit démarrer la construction. Les valeurs valides sont les suivantes :
  - `EXPRESSION_MATCH_ONLY`— votre pipeline créera une nouvelle image chaque fois que l'expression cron correspond à l'heure actuelle.
  - `EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE`— votre pipeline ne démarrera pas une nouvelle génération d'image à moins que des modifications ne soient en attente de modification de votre image de base ou de vos composants.

Lorsque vous exécutez la create-image-pipeline commande dans le AWS CLI, de nombreuses ressources de configuration sont facultatives. Cependant, certaines ressources ont des exigences conditionnelles, en fonction du type d'image créé par le pipeline. Les ressources suivantes sont requises pour les pipelines d'images AMI :

- Recette d'image (ARN)
- ARN de configuration de l'infrastructure

1. Créer un fichier JSON d'entrée CLI

   Utilisez votre outil d'édition de fichiers préféré pour créer un fichier JSON avec les clés suivantes, ainsi que des valeurs valides pour votre environnement. Cet exemple utilise un fichier nommé`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**
>
> - Vous devez inclure l'option `file://` au début du chemin du fichier JSON.
> - Le chemin d'accès du fichier JSON doit suivre la convention appropriée pour le système d'exploitation de base sur lequel vous exécutez la commande. En effet, Windows utilise la barre oblique inverse (\) pour faire référence au chemin du répertoire, et Linux utilise la barre oblique directe (/).

2. Exécutez la commande suivante en utilisant le fichier que vous avez créé en entrée.

```
aws imagebuilder create-image-pipeline --cli-input-json file://create-image-
pipeline.json
```

Mettre à jour les pipelines d'images AMI (console)

Après avoir créé un pipeline d'images Image Builder pour votre image AMI, vous pouvez modifier la configuration de l'infrastructure et les paramètres de distribution depuis la console Image Builder.

Pour mettre à jour un pipeline d'images avec une nouvelle recette d'image, vous devez utiliser le AWS CLI. Pour plus d'informations, consultez Mettre à jour les pipelines d'images AMI (AWS CLI) dans ce guide.

Choisissez un pipeline Image Builder existant

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Pour voir la liste des pipelines d'images créés sous votre compte, choisissez Pipelines d'images dans le volet de navigation.

   > **Note**
   >
   > La liste des pipelines d'images inclut un indicateur du type d'image de sortie créée par le pipeline : AMI ou Docker.

3. Pour afficher les détails ou modifier un pipeline, cliquez sur le lien Nom du pipeline. Cela ouvre la vue détaillée du pipeline.

   > **Note**
   >
   > Vous pouvez également cocher la case à côté du nom du pipeline, puis sélectionner Afficher les détails.

Détails du pipeline

La page de détails du pipeline comprend les sections suivantes :

Récapitulatif

La section en haut de la page résume les principaux détails du pipeline qui sont visibles lorsque l'un des onglets de détails est ouvert. Les informations affichées dans cette section ne sont modifiables que dans leurs onglets de détails respectifs.

Onglets détaillés

- Images de sortie : affiche les images de sortie produites par le pipeline.
- Recette illustrée — Affiche les détails de la recette. Une fois que vous avez créé une recette, vous ne pouvez pas la modifier. Vous devez créer une nouvelle version de la recette à partir de la page Image recipes de la console Image Builder ou en utilisant les commandes Image Builder dans le AWS CLI. Pour plus d'informations, consultez Gérez les recettes.
- Configuration de l'infrastructure : affiche des informations modifiables pour configurer votre infrastructure de pipeline de construction.

- Paramètres de distribution — Affiche les informations modifiables pour la distribution des AMI.
- EventBridge règles — Pour le bus d'événements sélectionné, affiche EventBridge les règles qui ciblent le pipeline actuel. Inclut les actions Créer un bus d'événements et Créer des règles liées à la EventBridge console. Pour plus d'informations sur cet onglet, consultez EventBridge Règles d'utilisation.

Modifier la configuration de l'infrastructure de votre pipeline

La configuration de l'infrastructure inclut les détails suivants que vous pouvez modifier après avoir créé le pipeline :

- Description de la configuration de votre infrastructure.
- Le rôle IAM à associer au profil d'instance.
- AWS infrastructure, y compris le type d'instance et une rubrique SNS pour les notifications.
- VPC, sous-réseau et groupes de sécurité.
- Paramètres de résolution des problèmes, notamment l'arrêt de l'instance en cas de défaillance, la paire de clés pour la connexion et un emplacement du compartiment S3 facultatif pour les journaux d'instance.

Pour modifier la configuration de l'infrastructure depuis la page des détails du pipeline, procédez comme suit :

1. Choisissez l'onglet Configuration de l'infrastructure.
2. Choisissez Modifier dans le coin supérieur droit du panneau des détails de configuration.
3. Lorsque vous êtes prêt à enregistrer les mises à jour que vous avez apportées à la configuration de votre infrastructure, choisissez Enregistrer les modifications.

Modifier les paramètres de distribution de votre pipeline

Les paramètres de distribution incluent les détails suivants que vous pouvez modifier après avoir créé le pipeline :

- Description de votre configuration de distribution.
- Paramètres régionaux pour les régions dans lesquelles vous distribuez votre image. La région 1 correspond par défaut à la région dans laquelle vous avez créé le pipeline. Vous pouvez ajouter

des régions à distribuer à l'aide du bouton Ajouter une région, et vous pouvez supprimer toutes les régions à l'exception de la région 1.

Les paramètres régionaux incluent :

- Région cible
- Le nom de l'AMI de sortie
- Autorisations de lancement et comptes avec lesquels les partager
- Licences associées (configurations de licences associées)

> **Note**
>
> Les paramètres du License Manager ne seront pas répliqués dans AWS les régions qui doivent être activées dans votre compte, par exemple entre les régions `ap-east-1` (Hong Kong) et `me-south-1` (Bahreïn).

Pour modifier vos paramètres de distribution depuis la page des détails du pipeline, procédez comme suit :

1. Choisissez l'onglet Paramètres de distribution.
2. Choisissez Modifier dans le coin supérieur droit du panneau des détails de distribution.
3. Lorsque vous êtes prêt à enregistrer vos mises à jour, choisissez Enregistrer les modifications.

Modifier le calendrier de construction de votre pipeline

La page Modifier le pipeline contient les informations suivantes que vous pouvez modifier après avoir créé le pipeline :

- Description de votre pipeline.
- Collecte de métadonnées améliorée. Cette option est activée par défaut. Pour le désactiver, décochez la case Activer la collecte améliorée de métadonnées.
- Le calendrier de construction de votre pipeline. Vous pouvez modifier vos options de planification et tous les paramètres ici.

Pour modifier votre pipeline depuis la page des détails du pipeline, procédez comme suit :

1. Dans le coin supérieur droit de la page des détails du pipeline, choisissez Actions, puis Modifier le pipeline.
2. Lorsque vous êtes prêt à enregistrer vos mises à jour, choisissez Enregistrer les modifications.

> **Note**
>
> Pour plus d'informations sur la planification de votre build à l'aide d'expressions cron, consultezUtiliser des expressions cron dans EC2 Image Builder.

Mettre à jour les pipelines d'images AMI (AWS CLI)

Vous pouvez mettre à jour un pipeline d'images AMI en utilisant un fichier JSON comme entrée de la update-image-pipeline commande dans le AWS CLI. Pour configurer le fichier JSON, vous devez disposer des Amazon Resource Names (ARN) pour référencer les ressources existantes suivantes :

- Pipeline d'images à mettre à jour
- Recette d'images
- Configuration de l'infrastructure
- Paramètres de distribution

Vous pouvez mettre à jour un pipeline d'images AMI à l'aide de la update-image-pipeline commande AWS CLI suivante :

> **Note**
>
> UpdateImagePipeline ne prend pas en charge les mises à jour sélectives pour le pipeline. Vous devez spécifier toutes les propriétés requises dans la demande de mise à jour, et pas uniquement les propriétés modifiées.

1. Créer un fichier JSON d'entrée CLI

   Utilisez votre outil d'édition de fichiers préféré pour créer un fichier JSON avec les clés suivantes, ainsi que des valeurs valides pour votre environnement. Cet exemple utilise un fichier nommé`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**
>
> - Vous devez inclure l'option `file://` au début du chemin du fichier JSON.
> - Le chemin d'accès du fichier JSON doit suivre la convention appropriée pour le système d'exploitation de base sur lequel vous exécutez la commande. En effet, Windows utilise la barre oblique inverse (\) pour faire référence au chemin du répertoire, et Linux utilise la barre oblique directe (/).

2. Exécutez la commande suivante en utilisant le fichier que vous avez créé en entrée.

```
aws imagebuilder update-image-pipeline --cli-input-json file://update-image-
pipeline.json
```

Création et mise à jour de pipelines d'images de conteneurs

Vous pouvez configurer, configurer et gérer des pipelines d'images de conteneurs à l'aide de la console Image Builder, via l'API Image Builder ou à l'aide de imagebuilder commandes dans le AWS CLI. L'assistant de création de la console de pipeline d'images fournit des artefacts de départ et vous guide à travers les étapes à suivre pour :

- Sélectionnez une image de base parmi les images gérées à démarrage rapide, les référentiels Amazon ECR ou Docker Hub
- Ajouter et supprimer des logiciels
- Personnaliser les paramètres et les scripts
- Exécuter les tests sélectionnés
- Créez un Dockerfile à l'aide de variables de génération préconfigurées.
- Diffuser des images dans les AWS régions

Pour plus d'informations et un step-by-step didacticiel sur l'utilisation de l'assistant de console Create Image Pipeline, consultezCréez un pipeline d'images de conteneur à l'aide de l'assistant de console EC2 Image Builder.

Table des matières

Création d'un pipeline d'images de conteneur (AWS CLI)

Vous pouvez créer un pipeline d'images de conteneur en utilisant un fichier JSON comme entrée de la create-image-pipeline commande dans le AWS CLI.

La fréquence à laquelle votre pipeline crée une nouvelle image pour intégrer les mises à jour en attente à partir de votre image de base et de vos composants dépend de `schedule` celle que vous avez configurée. A `schedule` possède les attributs suivants :

- `scheduleExpression`— Définit le calendrier d'exécution de votre pipeline afin d'évaluer `pipelineExecutionStartCondition` et de déterminer s'il doit démarrer une construction.

Le planning est configuré avec des expressions cron. Pour plus d'informations sur le formatage d'une expression cron dans Image Builder, consultezUtiliser des expressions cron dans EC2 Image Builder.

- `pipelineExecutionStartCondition`— Détermine si votre pipeline doit démarrer la construction. Les valeurs valides sont les suivantes :
  - `EXPRESSION_MATCH_ONLY`— votre pipeline créera une nouvelle image chaque fois que l'expression cron correspond à l'heure actuelle.
  - `EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE`— votre pipeline ne démarrera pas une nouvelle génération d'image à moins que des modifications ne soient en attente de modification de votre image de base ou de vos composants.

Lorsque vous exécutez la create-image-pipeline commande dans le AWS CLI, de nombreuses ressources de configuration sont facultatives. Cependant, certaines ressources ont des exigences conditionnelles, en fonction du type d'image créé par le pipeline. Les ressources suivantes sont requises pour les pipelines d'images de conteneurs :

- Recette du contenant ARN
- ARN de configuration de l'infrastructure

Si vous n'incluez aucune ressource de configuration de distribution lorsque vous exécutez la create-image-pipeline commande, l'image de sortie est stockée dans le référentiel ECR que vous spécifiez comme référentiel cible dans votre recette de conteneur dans la région où vous exécutez la commande. Si vous incluez une ressource de configuration de distribution pour votre pipeline, le référentiel cible que vous avez spécifié pour la première région de la distribution est utilisé.

1. Créer un fichier JSON d'entrée CLI

   Utilisez votre outil d'édition de fichiers préféré pour créer un fichier JSON avec les clés suivantes, ainsi que des valeurs valides pour votre environnement. Cet exemple utilise un fichier nommé`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**
>
> - Vous devez inclure l'option `file://` au début du chemin du fichier JSON.
> - Le chemin d'accès du fichier JSON doit suivre la convention appropriée pour le système d'exploitation de base sur lequel vous exécutez la commande. En effet, Windows utilise la barre oblique inverse (\) pour faire référence au chemin du répertoire, et Linux utilise la barre oblique directe (/).

2. Exécutez la commande suivante en utilisant le fichier que vous avez créé en entrée.

```
aws imagebuilder create-image-pipeline --cli-input-json file://create-image-
pipeline.json
```

Mettre à jour un pipeline d'images de conteneur (console)

Après avoir créé un pipeline d'images de conteneur Image Builder pour votre image Docker, vous pouvez modifier la configuration de l'infrastructure et les paramètres de distribution depuis la console Image Builder.

Pour mettre à jour un pipeline d'images de conteneur avec une nouvelle recette de conteneur, vous devez utiliser le AWS CLI. Pour plus d'informations, consultez Mettre à jour les pipelines d'images du conteneur (AWS CLI) dans ce guide.

Choisissez un pipeline d'images Docker Image Builder existant

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Pour voir la liste des pipelines d'images créés sous votre compte, choisissez Pipelines d'images dans le volet de navigation.

   > **Note**
   >
   > La liste des pipelines d'images inclut un indicateur du type d'image de sortie créée par le pipeline : AMI ou Docker.

3. Pour afficher les détails ou modifier un pipeline, cliquez sur le lien Nom du pipeline. Cela ouvre la vue détaillée du pipeline.

   > **Note**
   >
   > Vous pouvez également cocher la case à côté du nom du pipeline, puis sélectionner Afficher les détails.

Détails du pipeline

La page de détails du pipeline EC2 Image Builder inclut les sections suivantes :

Récapitulatif

La section en haut de la page récapitule les principaux détails du pipeline qui sont visibles lorsque l'un des onglets de détails est ouvert. Les informations affichées dans cette section ne sont modifiables que dans leurs onglets de détails respectifs.

Onglets détaillés

- Images de sortie : affiche les images de sortie produites par le pipeline.
- Recette du contenant — Affiche les détails de la recette. Une fois que vous avez créé une recette, vous ne pouvez pas la modifier. Vous devez créer une nouvelle version de la recette depuis la

page des recettes du conteneur. Pour plus d'informations, consultez Création d'une nouvelle version d'une recette de conteneur.

- Configuration de l'infrastructure : affiche des informations modifiables pour configurer votre infrastructure de pipeline de construction.
- Paramètres de distribution : affiche des informations modifiables pour la distribution d'images Docker.
- EventBridge règles — Pour le bus d'événements sélectionné, affiche EventBridge les règles qui ciblent le pipeline actuel. Inclut les actions Créer un bus d'événements et Créer des règles liées à la EventBridge console. Pour plus d'informations sur cet onglet, consultez EventBridge Règles d'utilisation.

Modifier la configuration de l'infrastructure de votre pipeline

La configuration de l'infrastructure inclut les détails suivants que vous pouvez modifier après avoir créé le pipeline :

- Description de la configuration de votre infrastructure.
- Rôle IAM à associer au profil d'instance.
- AWS infrastructure, y compris le type d'instance et une rubrique SNS pour les notifications.
- VPC, sous-réseau et groupes de sécurité.
- Paramètres de résolution des problèmes, notamment l'arrêt de l'instance en cas de défaillance, la paire de clés pour la connexion et un emplacement du compartiment S3 facultatif pour les journaux d'instance.

Pour modifier la configuration de l'infrastructure depuis la page des détails du pipeline, procédez comme suit :

1. Choisissez l'onglet Configuration de l'infrastructure.
2. Choisissez Modifier dans le coin supérieur droit du panneau des détails de configuration.
3. Lorsque vous êtes prêt à enregistrer les mises à jour que vous avez apportées à la configuration de votre infrastructure, choisissez Enregistrer les modifications.

Modifier les paramètres de distribution de votre pipeline

Les paramètres de distribution incluent les détails suivants que vous pouvez modifier après avoir créé le pipeline :

- Description de vos paramètres de distribution.
- Paramètres régionaux pour les régions dans lesquelles vous distribuez votre image. La région 1 correspond par défaut à la région dans laquelle vous avez créé le pipeline. Vous pouvez ajouter des régions à distribuer à l'aide du bouton Ajouter une région, et vous pouvez supprimer toutes les régions à l'exception de la région 1.

  Les paramètres régionaux incluent :

  - Région cible
  - Le service est défini par défaut sur « ECR » et n'est pas modifiable.
  - Nom du référentiel : nom de votre référentiel cible (sans compter l'emplacement Amazon ECR). Par exemple, le nom du référentiel avec son emplacement ressemblerait au modèle suivant :

    ```
    <account-id>.dkr.ecr.<region>.amazonaws.com/<repository-name>
    ```

    > **Note**
    >
    > Si vous modifiez le nom du référentiel, seules les images créées après le changement de nom seront ajoutées sous le nouveau nom. Toutes les images précédemment créées par votre pipeline restent dans leur référentiel d'origine.

Pour modifier vos paramètres de distribution depuis la page des détails du pipeline, procédez comme suit :

1. Choisissez l'onglet Paramètres de distribution.
2. Choisissez Modifier dans le coin supérieur droit du panneau des détails de distribution.
3. Lorsque vous êtes prêt à enregistrer les mises à jour que vous avez apportées à vos paramètres de distribution, choisissez Enregistrer les modifications.

Modifier le calendrier de construction de votre pipeline

La page Modifier le pipeline contient les informations suivantes que vous pouvez modifier après avoir créé le pipeline :

- Description de votre pipeline.
- Collecte de métadonnées améliorée. Cette option est activée par défaut. Pour le désactiver, décochez la case Activer la collecte améliorée de métadonnées.
- Le calendrier de construction de votre pipeline. Vous pouvez modifier vos options de planification et tous les paramètres de cette section.

Pour modifier votre pipeline depuis la page des détails du pipeline, procédez comme suit :

1. Dans le coin supérieur droit de la page des détails du pipeline, choisissez Actions, puis Modifier le pipeline.
2. Lorsque vous êtes prêt à enregistrer vos mises à jour, choisissez Enregistrer les modifications.

> **Note**
>
> Pour plus d'informations sur la planification de votre build à l'aide d'expressions cron, consultezUtiliser des expressions cron dans EC2 Image Builder.

Mettre à jour les pipelines d'images du conteneur (AWS CLI)

Vous pouvez mettre à jour un pipeline d'images de conteneur à l'aide d'un fichier JSON comme entrée de la update-image-pipeline commande dans le AWS CLI. Pour configurer le fichier JSON, vous devez disposer des Amazon Resource Names (ARN) pour référencer les ressources existantes suivantes :

- Pipeline d'images à mettre à jour
- Recette de contenant
- Configuration de l'infrastructure
- Paramètres de distribution (s'ils sont inclus dans le pipeline actuel)

> **Note**
>
> Si la ressource des paramètres de distribution est incluse, le référentiel ECR spécifié comme référentiel cible dans les paramètres de distribution de la région où la commande s'exécute (région 1) a priorité sur le référentiel cible spécifié dans la recette du conteneur.

Procédez comme suit pour mettre à jour un pipeline d'images de conteneur à l'aide de la update-image-pipeline commande suivante AWS CLI :

> **Note**
>
> UpdateImagePipeline ne prend pas en charge les mises à jour sélectives pour le pipeline. Vous devez spécifier toutes les propriétés requises dans la demande de mise à jour, et pas uniquement les propriétés modifiées.

1. Créer un fichier JSON d'entrée CLI

   Utilisez votre outil d'édition de fichiers préféré pour créer un fichier JSON avec les clés suivantes, ainsi que des valeurs valides pour votre environnement. Cet exemple utilise un fichier nommé`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**
>
> - Vous devez inclure l'option `file://` au début du chemin du fichier JSON.
> - Le chemin d'accès du fichier JSON doit suivre la convention appropriée pour le système d'exploitation de base sur lequel vous exécutez la commande. En effet, Windows utilise la barre oblique inverse (\) pour faire référence au chemin du répertoire, et Linux utilise la barre oblique directe (/).

2. Exécutez la commande suivante en utilisant le fichier que vous avez créé en entrée.

```
aws imagebuilder update-image-pipeline --cli-input-json file://update-image-pipeline.json
```

Configurer des flux de production d'images pour votre pipeline EC2 Image Builder

Avec les flux de travail d'images, vous pouvez personnaliser les flux de travail exécutés par votre pipeline afin de créer et de tester des images en fonction de vos besoins. Les flux de travail que vous définissez s'exécutent dans le contexte du framework de flux de travail Image Builder. Pour plus d'informations sur les étapes qui constituent la structure du flux de travail, consultezGérez les flux de travail de création et de test pour les images EC2 Image Builder.

Créer un flux de travail

Créez des flux de travail exécutés au cours de la `Build` phase du framework de flux de travail. Vous ne pouvez spécifier qu'un seul flux de production pour votre pipeline. Vous pouvez également ignorer complètement la génération pour configurer un pipeline de test uniquement.

Flux de travail de test

Les flux de travail de test sont exécutés au cours de la `Test` phase du framework de flux de travail. Vous pouvez définir jusqu'à dix flux de travail de test pour votre pipeline. Vous pouvez

également ignorer complètement les tests si vous souhaitez uniquement que votre pipeline soit créé.

Définissez des groupes de test pour les flux de travail de test

Les flux de travail de test sont définis au sein des groupes de test. Vous pouvez exécuter jusqu'à dix flux de travail de test pour votre pipeline. Vous décidez d'exécuter les flux de travail de test dans un ordre spécifique ou d'en exécuter autant que possible en même temps. Leur mode d'exécution dépend de la façon dont vous définissez vos groupes de test. Les scénarios suivants illustrent plusieurs manières de définir vos flux de travail de test.

> **Note**
>
> Si vous utilisez la console pour créer des flux de travail, nous vous recommandons de prendre le temps de planifier la manière dont vous souhaitez exécuter vos flux de travail de test avant de définir vos groupes de test. Dans la console, vous pouvez ajouter ou supprimer des flux de travail et des groupes de test, mais vous ne pouvez pas les réorganiser.

Scénario 1 : exécuter un flux de travail de test à la fois

Pour exécuter tous vos flux de travail de test un par un, vous pouvez configurer jusqu'à dix groupes de test, chacun contenant un seul flux de travail de test. Les groupes de test s'exécutent un par un, dans l'ordre dans lequel vous les ajoutez à votre pipeline. C'est une façon de garantir que vos flux de travail de test s'exécutent un par un dans un ordre spécifique.

Scénario 2 : exécuter plusieurs flux de travail de test en même temps

Si l'ordre n'a pas d'importance et que vous souhaitez exécuter autant de flux de travail de test que possible en même temps, vous pouvez configurer un seul groupe de test et y placer le nombre maximum de flux de travail de test. Image Builder démarre jusqu'à cinq flux de travail de test en même temps, et lance des flux de travail de test supplémentaires au fur et à mesure que les autres se terminent. Si votre objectif est d'exécuter vos flux de travail de test le plus rapidement possible, c'est une façon de le faire.

Scénario 3 : Mix and match

Si vous avez un scénario mixte, avec certains flux de travail de test pouvant être exécutés en même temps et d'autres devant être exécutés un par un, vous pouvez configurer vos groupes de test pour

atteindre cet objectif. La seule limite à la manière dont vous configurez vos groupes de test est le nombre maximum de flux de travail de test pouvant être exécutés pour votre pipeline.

Définir les paramètres du flux de travail dans un pipeline Image Builder (console)

Les paramètres du flux de travail fonctionnent de la même manière pour les flux de travail de création et les flux de travail de test. Lorsque vous créez ou mettez à jour un pipeline, vous sélectionnez les flux de travail de création et de test que vous souhaitez inclure. Si vous avez défini des paramètres dans le document de flux de travail pour un flux de travail que vous avez sélectionné, Image Builder les affiche dans le panneau Paramètres. Le panneau est masqué pour les flux de travail dont les paramètres ne sont pas définis.

Chaque paramètre affiche les attributs suivants définis par votre document de flux de travail :

- Nom (non modifiable) : nom du paramètre.
- Type (non modifiable) : type de données pour la valeur du paramètre.
- Valeur — La valeur du paramètre. Vous pouvez modifier la valeur du paramètre pour la définir pour votre pipeline.

Spécifiez le rôle de service IAM qu'Image Builder utilise pour exécuter des actions de flux de travail

Accès à un service

Pour exécuter des flux de travail d'images, Image Builder doit être autorisé à effectuer des actions de flux de travail. Vous pouvez spécifier le rôle AWSServiceRoleForImageBuilderlié au service, ou vous pouvez définir votre propre rôle personnalisé pour l'accès au service, comme suit.

- Console — Dans l'assistant de pipeline, étape 3 Définissez le processus de création d'image, sélectionnez le rôle lié au service ou votre propre rôle personnalisé dans la liste des rôles IAM du panneau d'accès aux services.
- API Image Builder : dans la demande CreateImaged'action, spécifiez le rôle lié au service ou votre propre rôle personnalisé comme valeur du paramètre. `executionRole`

Pour en savoir plus sur la création d'un rôle de service, consultez la section Création d'un rôle pour déléguer des autorisations à un AWS service dans le Guide de AWS Identity and Access Management l'utilisateur.

Exécutez votre pipeline d'images

Si vous avez choisi l'option de planification manuelle pour votre pipeline, celui-ci ne s'exécutera que lorsque vous lancerez manuellement la construction. Si vous avez choisi l'une des options de planification automatique, vous pouvez également l'exécuter manuellement, entre les exécutions planifiées régulièrement. Par exemple, si vous avez un pipeline qui s'exécute normalement une fois par mois, mais que vous devez intégrer une mise à jour de l'un de vos composants deux semaines après l'exécution précédente, vous pouvez choisir d'exécuter votre pipeline manuellement.

Console

Pour exécuter votre pipeline depuis la page des détails du pipeline de la console Image Builder, choisissez Run pipeline dans le menu Actions en haut de la page. Un message d'état apparaît en haut de la page pour vous informer que votre pipeline a démarré ou s'il y a une erreur.

1. Dans le coin supérieur gauche de la page des détails du pipeline, choisissez Exécuter le pipeline dans le menu Actions.
2. Vous pouvez voir l'état actuel de votre pipeline dans l'onglet Images de sortie, dans la colonne État.

AWS CLI

L'exemple suivant montre comment utiliser la start-image-pipeline-execution commande du AWS CLI pour démarrer un pipeline d'images manuellement. Lorsque vous exécutez cette commande, le pipeline crée et distribue une nouvelle image.

```
aws imagebuilder start-image-pipeline-execution --image-pipeline-arn
 arn:aws:imagebuilder:us-west-2:111122223333:image-pipeline/my-example-pipeline
```

Pour voir quelles ressources sont créées lors de l'exécution du pipeline de génération, voirRessources créées.

Utiliser des expressions cron dans EC2 Image Builder

Utilisez des expressions cron pour EC2 Image Builder afin de configurer une fenêtre temporelle afin d'actualiser votre image avec des mises à jour qui s'appliquent à l'image de base et aux composants de votre pipeline. La fenêtre temporelle pour l'actualisation de votre pipeline commence avec l'heure que vous avez définie dans l'expression cron. Vous pouvez définir l'heure dans votre expression cron à la minute près. La construction de votre pipeline peut s'exécuter à l'heure de début ou après.

L'exécution de votre build peut parfois prendre quelques secondes, voire une minute.

> **Note**
>
> Les expressions Cron utilisent le fuseau horaire UTC (Universal Coordinated Time) par défaut, ou vous pouvez spécifier le fuseau horaire. Pour plus d'informations sur l'heure UTC et pour connaître le décalage correspondant à votre fuseau horaire, consultez la section Abréviations des fuseaux horaires — Liste mondiale.

Valeurs prises en charge pour les expressions cron dans Image Builder

EC2 Image Builder utilise un format cron composé de six champs obligatoires. Chacune est séparée des autres par un espace entre les deux, sans espaces de début ou de fin :

```
<Minute> <Hour> <Day> <Month> <Day of the week> <Year>
```

Le tableau suivant montre les valeurs prises en charge pour les entrées cron obligatoires.

Valeurs prises en charge pour les expressions cron

Champ	Valeurs	Caractères génériques
Minute	`0-59`	`, - * /`
Heure	`0-23`	`, - * /`
jour	`1-31`	`, - * ? / L W`
Mois	`1-12` ou `jan-dec`	`, - * /`
Jour de la semaine	`1-7` ou `sun-sat`	`, - * ? L #`

Champ	Valeurs	Caractères génériques
Année	1970-2199	, - * /

Caractères génériques

Le tableau suivant décrit comment Image Builder utilise des caractères génériques pour les expressions cron. N'oubliez pas que le démarrage de la compilation peut prendre jusqu'à une minute après l'heure spécifiée.

Caractères génériques pris en charge pour les expressions cron

Caractère générique	Description
,	Le caractère générique , (virgule) inclut des valeurs supplémentaires. Dans le champ Mois, `jan,feb,mar` inclut janvier, février et mars.
-	Le caractère générique - (tiret) spécifie des plages. Dans le champ Jour du mois, `1-15` inclut les jours 1 à 15 du mois spécifié.
*	Le caractère générique * (astérisque) inclut toutes les valeurs valides du champ.
?	Le caractère générique ? (point d'interrogation) indique que la valeur du champ dépend d'un autre paramètre. Dans le cas des ay-of-week champs Day et D, lorsque l'un est spécifié ou inclut toutes les valeurs possibles (`*`), l'autre doit être `?` a. Vous ne pouvez pas spécifier les deux. Par exemple, si vous saisissez un 7 dans le champ Jour (lancez le build le septième jour du mois), la ay-of-week position D doit contenir un`?`.
/	Le caractère générique / (barre oblique) spécifie les incréments. Par exemple, si vous

Caractère générique	Description
	souhaitez que votre build soit exécuté tous les deux jours, entrez `*/2` dans le champ jour.
L	Le caractère générique L dans l'un ou l'autre des champs du jour indique le dernier jour : du 28 au 31 pour le jour du mois, selon le mois, ou le dimanche, pour le jour de la semaine.
W	Le caractère générique W dans le ay-of-month champ D indique un jour de semaine. Dans le ay-of-month champ D, si vous entrez un nombre avant le`W`, cela signifie que vous souhaitez cibler le jour de la semaine le plus proche de ce jour. Par exemple, si vous le spécifiez`3W`, vous souhaitez que votre build soit exécuté le jour de la semaine le plus proche du troisième jour du mois.
#	Le # (hachage) n'est autorisé que pour le champ du jour de la semaine et doit être suivi d'un nombre compris entre 1 et 5. Le numéro indique les semaines d'un mois donné qui s'appliquent à l'exécution de la version. Par exemple, si vous souhaitez que votre build soit exécuté le deuxième vendredi de chaque mois, utilisez `fri#2` le champ du jour de la semaine.

Restrictions

- Vous ne pouvez pas spécifier les ay-of-week champs D ay-of-month et D dans la même expression cron. Si vous spécifiez une valeur ou `*` dans l'un de ces champs, vous devez utiliser un `?` dans l'autre.
- Les expressions cron qui entraînent des fréquences d'une rapidité supérieure à une minute ne sont pas prises en charge.

Exemples d'expressions cron dans EC2 Image Builder

Les expressions Cron sont saisies différemment pour la console Image Builder et pour l'API ou la CLI. Pour voir des exemples, choisissez l'onglet qui s'applique à vous.

Image Builder console

Les exemples suivants montrent les expressions cron que vous pouvez saisir dans la console pour votre calendrier de génération. L'heure UTC est spécifiée à l'aide d'une horloge de 24 heures.

Fonctionne tous les jours à 10 h 00 (UTC)

```
0 10 * * ? *
```

Fonctionne tous les jours à 12 h 15 (UTC)

```
15 12 * * ? *
```

Fonctionne tous les jours à minuit (UTC)

```
0 0 * * ? *
```

Ouvert à 10 h 00 (UTC) tous les matins de la semaine

```
0 10 ? * 2-6 *
```

Ouvert à 18 h (UTC) tous les soirs de semaine

```
0 18 ? * mon-fri *
```

Ouvert à 8 h 00 (UTC) le premier jour de chaque mois

```
0 8 1 * ? *
```

Organisé le deuxième mardi de chaque mois à 22 h 30 (UTC)

```
30 22 ? * tue#2 *
```

> **Tip**
>
> Si vous ne souhaitez pas que votre tâche de pipeline s'étende jusqu'au lendemain pendant son exécution, assurez-vous de prendre en compte le temps de construction lorsque vous spécifiez l'heure de début.

API/CLI

Les exemples suivants montrent les expressions cron que vous pouvez saisir pour votre calendrier de construction à l'aide de commandes CLI ou de requêtes d'API. Seule l'expression cron est affichée.

Fonctionne tous les jours à 10 h 00 (UTC)

```
cron(0 10 * * ? *)
```

Fonctionne tous les jours à 12 h 15 (UTC)

```
cron(15 12 * * ? *)
```

Fonctionne tous les jours à minuit (UTC)

```
cron(0 0 * * ? *)
```

Ouvert à 10 h 00 (UTC) tous les matins de la semaine

```
cron(0 10 ? * 2-6 *)
```

Ouvert à 18 h 00 (UTC) tous les soirs de semaine

```
cron(0 18 ? * mon-fri *)
```

Ouvert à 8 h 00 (UTC) le premier jour de chaque mois

```
cron(0 8 1 * ? *)
```

Organisé le deuxième mardi de chaque mois à 22 h 30 (UTC)

```
cron(30 22 ? * tue#2 *)
```

> **Tip**
>
> Si vous ne souhaitez pas que votre tâche de pipeline s'étende jusqu'au lendemain pendant son exécution, assurez-vous de prendre en compte le temps de construction lorsque vous spécifiez l'heure de début.

Expressions de taux dans EC2 Image Builder

Une expression de fréquence démarre au moment où vous créez la règle d'événement planifié, puis s'exécute selon le calendrier défini.

Les expressions de fréquence comportent deux champs obligatoires. Ces champs sont séparés par un espace.

Syntaxe

```
rate(value unit)
```

value

Nombre positif.

unité

Unité de temps. Des unités différentes sont nécessaires pour les valeurs de 1 (par exemple `minute`) et les valeurs supérieures à 1, (par exemple, `minutes`).

Valeurs valides : minute | minutes | heure | heures | jour | jours

Restrictions

Si la valeur est égale à 1, l'unité doit être au singulier. De même, pour les valeurs supérieures à 1, l'unité doit être au pluriel. Par exemple, `rate(1 hours)` et `rate(5 hour)` ne sont pas valides, mais `rate(1 hour)` et `rate(5 hours)` sont valides.

Utiliser des EventBridge règles avec les pipelines Image Builder

Les événements issus d'un large éventail de services AWS et de services partenaires sont diffusés sur les bus d' EventBridge événements Amazon en temps quasi réel. Vous pouvez également générer des événements personnalisés et envoyer des événements depuis vos propres applications à EventBridge. Les bus d'événements utilisent des règles pour déterminer où acheminer les données d'événements.

Les pipelines Image Builder sont disponibles en tant que cibles de EventBridge règles, ce qui signifie que vous pouvez exécuter un pipeline Image Builder en fonction des règles que vous créez pour répondre aux événements survenus dans le bus ou selon un calendrier.

> **Note**
>
> Les bus dédiés aux événements sont spécifiques à une région. La règle et la cible doivent se trouver dans la même région.

Table des matières

EventBridge termes

Cette section contient un résumé des termes qui vous aideront à comprendre comment EventBridge s'intègre à vos pipelines Image Builder.

Événement

Décrit une modification d'un environnement susceptible d'affecter une ou plusieurs ressources de l'application. L'environnement peut être un AWS environnement, un service ou une application partenaire SaaS, ou l'une de vos applications ou services. Vous pouvez également configurer des événements planifiés sur une chronologie.

Bus d'événement

Un pipeline qui reçoit les données d'événements des applications et des services.

Source

Le service ou l'application qui a envoyé l'événement au bus d'événements.

Cible

Ressource ou point de terminaison EventBridge qui s'appelle lorsqu'il correspond à une règle, fournissant les données de l'événement à la cible.

Règle

Une règle correspond à des événements entrants et les envoie vers des cibles pour être traités. Une seule règle peut envoyer un événement à plusieurs cibles, qui peuvent ensuite s'exécuter en parallèle. Les règles sont basées soit sur un modèle d'événement, soit sur un calendrier.

Modèle

Un modèle d'événement définit la structure de l'événement et les champs auxquels une règle correspond afin de lancer l'action cible.

Planificateur

Les règles de planification exécutent une action selon un calendrier, par exemple en exécutant un pipeline Image Builder pour actualiser une image tous les trimestres. Il existe deux types d'expressions de planification :

- Expressions Cron — Faites correspondre des critères de planification spécifiques à l'aide de la syntaxe cron qui peut définir des critères simples, par exemple, une exécution hebdomadaire un jour précis. Vous pouvez également établir des critères plus complexes, tels que la diffusion trimestrielle le cinquième jour du mois, entre 2 h et 4 h du matin.
- Expressions de taux — Spécifiez un intervalle régulier lorsque la cible est invoquée, par exemple toutes les 12 heures.

Afficher EventBridge les règles de votre pipeline Image Builder

L'onglet EventBridge règles de la page détaillée des pipelines d'images Image Builder affiche les bus d' EventBridge événements auxquels votre compte a accès, ainsi que les règles du bus d'événements sélectionné qui s'appliquent au pipeline actuel. Cet onglet renvoie également directement à la EventBridge console pour créer de nouvelles ressources.

Actions liées à la EventBridge console

- Créer un bus d'événements
- Créer une règle

Pour en savoir plus EventBridge, consultez les rubriques suivantes dans le guide de EventBridge l'utilisateur Amazon.

- Qu'est-ce qu'Amazon EventBridge
- Bus EventBridge événementiels Amazon
- EventBridge Événements Amazon
- EventBridge Règles Amazon

Utiliser EventBridge des règles pour planifier la construction d'un pipeline

Dans cet exemple, nous créons une nouvelle règle de planification pour le bus d'événements par défaut, à l'aide d'une expression de débit. Dans cet exemple, la règle génère un événement sur le bus d'événements tous les 90 jours. L'événement lance la création d'un pipeline pour actualiser l'image.

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Pour voir la liste des pipelines d'images créés sous votre compte, choisissez Pipelines d'images dans le volet de navigation.

   > **Note**
   >
   > La liste des pipelines d'images inclut un indicateur du type d'image de sortie créée par le pipeline : AMI ou Docker.

3. Pour afficher les détails ou modifier un pipeline, cliquez sur le lien Nom du pipeline. Cela ouvre la vue détaillée du pipeline.

   > **Note**
   >
   > Vous pouvez également cocher la case à côté du nom du pipeline, puis sélectionner Afficher les détails.

4. Ouvrez l'onglet EventBridge Règles.
5. Conservez le bus d'événements par défaut présélectionné dans le panneau Event Bus.
6. Choisissez Créer une règle. Cela vous amène à la page Créer une règle dans la EventBridge console Amazon.
7. Saisissez un nom et une description pour la règle. Le nom de la règle doit être unique dans le bus d'événements pour la région sélectionnée.
8. Dans le panneau Définir le modèle, choisissez l'option Planifier. Cela élargit le panel, avec le taux fixe pour chaque option sélectionnée.
9. Entrez `90` dans la première case, puis sélectionnez Jours dans la liste déroulante.
10. Effectuez les actions suivantes dans le panneau Sélectionner des cibles :

    a. Sélectionnez `EC2 Image Builder` dans la liste déroulante Cible.

b. Pour appliquer la règle à un pipeline Image Builder, sélectionnez le pipeline cible dans la liste déroulante Image Pipeline.

c. EventBridge a besoin d'une autorisation pour lancer une génération pour le pipeline sélectionné. Pour cet exemple, conservez l'option par défaut pour créer un nouveau rôle pour cette ressource spécifique.

d. Sélectionnez Ajouter une cible.

11. Sélectionnez Create (Créer).

> **Note**
>
> Pour en savoir plus sur les paramètres des règles d'expression de débit qui ne sont pas abordées dans cet exemple, consultez la section Expressions de débit dans le guide de EventBridge l'utilisateur Amazon.

Intégrer des produits et services dans EC2 Image Builder

EC2 Image Builder s'intègre à AWS Marketplace d' Services AWS autres applications pour vous aider à créer des images de machine personnalisées robustes et sécurisées.

Produits

Les recettes Image Builder peuvent intégrer des produits d'image provenant AWS Marketplace de composants gérés par Image Builder pour fournir des fonctionnalités de création et de test spécialisées, comme suit.

- AWS Marketplace produits d'image — Utilisez un produit d'image AWS Marketplace comme image de base dans votre recette afin de répondre aux normes organisationnelles, telles que CIS Hardening. Lorsque vous créez une recette depuis la console Image Builder, vous pouvez choisir parmi vos abonnements existants ou rechercher un produit spécifique auprès de AWS Marketplace. Lorsque vous créez une recette à partir de l'API, de la CLI ou du SDK Image Builder, vous pouvez spécifier un produit d'imagerie Amazon Resource Name (ARN) à utiliser comme image de base.
- AWSTOE composants : les composants que vous spécifiez dans vos recettes peuvent effectuer des actions de génération et de test, par exemple pour installer des logiciels ou effectuer une validation de conformité. Certains produits illustrés auxquels vous vous abonnez AWS Marketplace peuvent inclure un composant complémentaire que vous pouvez utiliser dans vos recettes. Les images CIS Hardened incluent un AWSTOE composant correspondant que vous pouvez utiliser dans votre recette pour appliquer les directives CIS Benchmarks de niveau 1 à votre configuration.

> **Note**
>
> Pour plus d'informations sur les produits liés à la conformité, consultez. Produits de conformité pour vos images Image Builder

Services

Image Builder s'intègre aux éléments suivants Services AWS pour fournir des mesures détaillées sur les événements, la journalisation et la surveillance. Ces informations vous aident à suivre votre activité, à résoudre les problèmes liés à la création d'images et à créer des automatisations basées sur les notifications d'événements.

- AWS CloudTrail— Surveille les événements Image Builder envoyés à CloudTrail. Pour plus d'informations CloudTrail, voir Qu'est-ce que c'est AWS CloudTrail ? dans le guide de AWS CloudTrail l'utilisateur.
- Amazon CloudWatch Logs — Surveillez, stockez et accédez à vos fichiers journaux Image Builder. Vous pouvez éventuellement enregistrer vos journaux dans un compartiment S3. Pour plus d'informations sur CloudWatch les journaux, consultez Qu'est-ce qu'Amazon CloudWatch Logs ? dans le guide de l'utilisateur d'Amazon CloudWatch Logs.
- Amazon EventBridge — Connectez-vous à un flux de données d'événements en temps réel provenant des activités d'Image Builder sur votre compte. Pour plus d'informations EventBridge, consultez Qu'est-ce qu'Amazon EventBridge ? dans le guide de EventBridge l'utilisateur Amazon.
- Amazon Inspector — Découvrez les vulnérabilités de vos logiciels et de vos paramètres réseau grâce à des scans automatiques pour que l'instance de test EC2 lancée par Image Builder crée une nouvelle image. Image Builder enregistre les résultats pour votre ressource d'image de sortie afin que vous puissiez les étudier et y remédier après la fin de votre instance de test. Pour plus d'informations sur les scans et les tarifs, consultez Qu'est-ce qu'Amazon Inspector ? dans le guide de l'utilisateur d'Amazon Inspector.

  Amazon Inspector peut également analyser vos référentiels ECR si vous configurez une analyse améliorée. Pour plus d'informations, consultez la section Numérisation d'images de conteneurs Amazon ECR dans le guide de l'utilisateur Amazon Inspector.

  > **Note**
  >
  > Amazon Inspector est une fonctionnalité payante.

- AWS Marketplace— Consultez la liste de vos abonnements actuels aux AWS Marketplace produits et recherchez des produits illustrés directement depuis Image Builder. Vous pouvez également utiliser un produit d'image auquel vous êtes abonné comme image de base pour une recette Image Builder. Pour plus d'informations sur la gestion des AWS Marketplace abonnements, consultez le Guide de AWS Marketplace l'acheteur.
- Amazon Simple Notification Service (Amazon SNS) : si cette option est configurée, publiez des messages détaillés concernant le statut de votre image sur une rubrique SNS à laquelle vous êtes abonné. Pour plus d'informations sur Amazon SNS, consultez Qu'est-ce qu'Amazon SNS ? dans le Guide du développeur Amazon Simple Notification Service.

Sujets relatifs à l'intégration des produits et services

AWS CloudTrail intégration dans Image Builder

Ce service prend en charge AWS CloudTrail. CloudTrail est un service qui enregistre vos AWS appels Compte AWS et transmet les fichiers journaux à un compartiment Amazon S3. En utilisant les informations collectées par CloudTrail, vous pouvez déterminer à quelles demandes ont été adressées avec succès Services AWS, qui a fait la demande, quand elle a été faite, etc. Pour plus d'informations sur CloudTrail l'intégration avec Image Builder, consultezJournalisation des appels d'API EC2 Image Builder à l'aide de AWS CloudTrail.

Pour en savoir plus CloudTrail, notamment comment l'activer et trouver vos fichiers journaux, consultez le guide de l'AWS CloudTrail utilisateur.

Intégration d'Amazon CloudWatch Logs dans Image Builder

CloudWatch La prise en charge des journaux est activée par défaut. Les journaux sont conservés sur l'instance pendant le processus de création et transmis à CloudWatch Logs. Les journaux d'instance sont supprimés de l'instance avant la création de l'image.

Les journaux de build sont transmis au groupe et au flux Image Builder CloudWatch Logs suivants :

LogGroup:

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

LogStream (x.x.x/x) :

```
ImageVersion/ImageBuildVersion
```

Vous pouvez désactiver le streaming des CloudWatch journaux en supprimant les autorisations suivantes associées au profil d'instance.

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

Pour un dépannage avancé, vous pouvez exécuter des commandes et des scripts prédéfinis à l'aide de AWS Systems Manager Run Command. Pour plus d'informations, consultez Résoudre les problèmes liés à EC2 Image Builder.

EventBridge Intégration d'Amazon dans Image Builder

Amazon EventBridge est un service de bus d'événements sans serveur que vous pouvez utiliser pour connecter votre application Image Builder à des données connexes provenant d'autres Services AWS applications. Dans EventBridge, une règle fait correspondre les événements entrants et les envoie aux cibles pour traitement. Une seule règle peut envoyer un événement à plusieurs cibles, et ces événements s'exécutent ensuite en parallèle.

Vous pouvez Services AWS ainsi automatiser et répondre automatiquement aux événements du système tels que les problèmes de disponibilité des applications ou les modifications des ressources. EventBridge Les événements de Services AWS sont transmis à EventBridge en temps quasi réel. Vous pouvez configurer des règles qui réagissent aux événements entrants afin de lancer des actions, par exemple en envoyant un événement à une fonction Lambda lorsque le statut d'une instance EC2 passe de « en attente » à « en cours d'exécution ». C'est ce qu'on appelle des modèles. Pour créer une règle basée sur un modèle d'événement, consultez la section Création de EventBridge règles Amazon qui réagissent aux événements dans le guide de EventBridge l'utilisateur Amazon.

Les actions qui peuvent être initiées automatiquement sont les suivantes :

- Invoquer une AWS Lambda fonction

- Appeler Run Command (Exécuter la commande) d'Amazon EC2
- Relayer l'événement à Amazon Kinesis Data Streams
- Activer une machine à AWS Step Functions états
- Notifier une rubrique Amazon SNS ou une file d'attente Amazon SQS

Vous pouvez également configurer des règles de planification pour que le bus d'événements par défaut exécute une action à intervalles réguliers, par exemple exécuter un pipeline Image Builder pour actualiser une image tous les trimestres. Il existe deux types d'expressions de planification :

- expressions cron — L'exemple suivant d'expression cron planifie l'exécution d'une tâche tous les jours à midi UTC+0 :

  ```
  cron(0 12 * * ? *)
  ```

  Pour plus d'informations sur l'utilisation des expressions cron avec EventBridge, consultez les expressions cron dans le guide de EventBridge l'utilisateur Amazon.
- expressions de taux — L'exemple suivant d'expression de taux planifie l'exécution d'une tâche toutes les 12 heures :

  ```
  rate(12 hour)
  ```

  Pour plus d'informations sur l'utilisation des expressions de débit avec EventBridge, consultez la section Expressions de débit dans le guide de EventBridge l'utilisateur Amazon.

Pour plus d'informations sur le mode EventBridge d'intégration aux pipelines d'images Image Builder, consultezUtiliser des EventBridge règles avec les pipelines Image Builder.

Intégration d'Amazon Inspector dans Image Builder

Lorsque vous activez le scan de sécurité avec Amazon Inspector, celui-ci analyse en permanence les images des machines et les instances en cours d'exécution de votre compte pour détecter les vulnérabilités du système d'exploitation et du langage de programmation. Lorsqu'elle est activée, l'analyse de sécurité est automatique et Image Builder peut enregistrer un instantané des résultats de votre instance de test lorsque vous créez une nouvelle image. Amazon Inspector est un service payant.

Lorsqu'Amazon Inspector découvre des vulnérabilités dans votre logiciel ou dans vos paramètres réseau, il prend les mesures suivantes :

- Vous informe qu'une découverte a été faite.
- Évalue la gravité du résultat. L'indice de gravité classe les vulnérabilités afin de vous aider à hiérarchiser vos découvertes, et inclut les valeurs suivantes :
  - Non trié
  - Informationnel
  - Faible
  - Medium
  - Élevée
  - Critique
- Fournit des informations sur la découverte et des liens vers des ressources supplémentaires pour plus de détails.
- Fournit des conseils de correction pour vous aider à résoudre les problèmes à l'origine du résultat.

Configuration des scans de sécurité

Si vous avez activé Amazon Inspector pour votre compte, Amazon Inspector scanne automatiquement les instances EC2 lancées par Image Builder pour créer et tester une nouvelle image. Ces instances ont une courte durée de vie pendant le processus de création et de test, et leurs résultats expirent normalement dès leur fermeture. Pour vous aider à étudier et corriger les résultats relatifs à votre nouvelle image, Image Builder peut éventuellement enregistrer sous forme de capture instantanée tous les résultats identifiés par Amazon Inspector sur votre instance de test pendant le processus de création.

Pour configurer les scans de sécurité pour votre pipeline, consultezConfigurez les scans de sécurité pour les images Image Builder dans AWS Management Console.

Passez en revue les résultats de sécurité

Dans la console Image Builder, vous pouvez consulter les résultats de sécurité pour toutes vos ressources Image Builder en un seul endroit. Vous pouvez consulter tous les résultats sur la page Résultats de sécurité dans la section Présentation de la sécurité, ou vous pouvez regrouper vos résultats par vulnérabilité, par pipeline d'images ou par image. La console affiche par défaut tous les résultats de sécurité. Le panneau récapitulatif de l'option Toutes les constatations de sécurité indique

le nombre de constatations que vous avez pour chaque niveau de gravité. Pour plus d'informations, consultez Gérez les résultats de sécurité relatifs aux images Image Builder dans AWS Management Console.

Pour en savoir plus sur les vulnérabilités détectées par Amazon Inspector, consultez la section Comprendre les découvertes dans Amazon Inspector dans le guide de l'utilisateur d'Amazon Inspector.

AWS Marketplace intégration dans Image Builder

AWS Marketplace est un catalogue numérique organisé dans lequel vous pouvez trouver et vous abonner à des logiciels, des données et des services tiers qui vous aident à créer des solutions adaptées aux besoins de votre entreprise. AWS Marketplace propose aux acheteurs authentifiés et aux vendeurs enregistrés des listes de logiciels appartenant à des catégories populaires telles que la sécurité, les réseaux, le stockage, l'apprentissage automatique, etc.

Un AWS Marketplace vendeur peut être un fournisseur de logiciels indépendant (ISV), un revendeur ou une personne qui a quelque chose à offrir qui fonctionne avec des AWS produits et des services. Lorsque le vendeur soumet un produit AWS Marketplace, il définit le prix du produit et les conditions générales d'utilisation. Les acheteurs acceptent les prix, les termes et conditions définis pour l'offre. Pour en savoir plus AWS Marketplace, voir Qu'est-ce que c'est AWS Marketplace ?

> **Note**
>
> Les fournisseurs de produits de données doivent satisfaire aux critères d'éligibilité de AWS Data Exchange. Pour plus d'informations, consultez la section Providing Data Products on AWS Data Exchange dans le Guide de l'utilisateur de AWS Data Exchange.

AWS Marketplace fonctionnalités d'intégration

Image Builder s'intègre AWS Marketplace à Image Builder pour fournir les fonctionnalités suivantes directement depuis la console Image Builder :

- Recherchez des produits illustrés disponibles dans AWS Marketplace.
- Consultez la liste de vos abonnements actuels aux AWS Marketplace produits.
- Utilisez un produit d' AWS Marketplace image comme image de base pour une recette Image Builder.

Pour les produits qui incluent des composants associés AWS Task Orchestrator and Executor (AWSTOE), vous pouvez filtrer le propriétaire du produit dans la console, ainsi que dans l'API, le SDK et la CLI. Pour plus d'informations, consultez Lister AWSTOE les composants.

Rechercher des produits AWS Marketplace illustrés à partir de la console Image Builder

Image Builder s'intègre AWS Marketplace à Image Builder pour afficher vos abonnements à des produits d'imagerie directement depuis la AWS Marketplacesection de la console Image Builder. Vous pouvez également rechercher des produits AWS Marketplace illustrés sur la page des produits Image sans quitter la console Image Builder.

Pour rechercher un produit AWS Marketplace imagé à partir de la console Image Builder, procédez comme suit :

1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
2. Dans le volet de navigation, choisissez Image products dans la AWS Marketplacesection.
3. La page Produits illustrés affiche un résumé des produits illustrés auxquels vous êtes abonné dans l'onglet Abonnements, ou vous pouvez rechercher des produits illustrés dans cet AWS Marketplaceonglet.

   Image Builder préfiltre les produits AWS Marketplace pour se concentrer sur les images de machine que vous pouvez utiliser dans vos recettes Image Builder. Pour plus d'informations sur AWS Marketplace l'intégration à Image Builder, choisissez l'onglet correspondant à ce que vous souhaitez voir.

   AWS Marketplace

   Cet onglet contient deux panneaux. Sur la gauche, le panneau Affiner les résultats vous permet de filtrer les résultats pour trouver les produits auxquels vous souhaitez vous abonner. Sur la droite, le panneau Rechercher des produits affiche les produits qui répondent à vos critères de filtrage et vous permet également de faire une recherche par nom de produit.

   Affiner les résultats

   La liste suivante présente quelques-uns des filtres que vous pouvez appliquer à votre recherche de produits :

- Sélectionnez une ou plusieurs catégories de produits, telles que les logiciels d'infrastructure ou le machine learning.
- Choisissez les systèmes d'exploitation pour votre produit d'image ou choisissez tous les produits pour une plate-forme de système d'exploitation spécifique, par exemple All Linux/Unix.
- Choisissez un ou plusieurs éditeurs pour afficher leurs produits disponibles. Cliquez sur le lien Afficher tout pour afficher tous les éditeurs dont les produits correspondent aux filtres que vous avez appliqués.

> **Note**
>
> Les noms des éditeurs ne sont pas classés par ordre alphabétique. Si vous recherchez un éditeur spécifique, par exemple`Center for Internet Security`, vous pouvez saisir une partie du nom dans le champ de recherche en haut de la boîte de dialogue Tous les éditeurs. Vous devez épeler le nom, sous forme d'abréviation, `CIS` afin de ne pas produire les résultats que vous recherchez. Vous pouvez également parcourir les noms des éditeurs page par page.

Les choix de filtres sont dynamiques. Chaque choix que vous faites affecte vos options pour toutes les autres catégories. Des milliers de produits sont disponibles AWS Marketplace, donc plus vous pouvez filtrer, plus vous avez de chances de trouver ce que vous recherchez.

Rechercher des produits

Pour trouver un produit spécifique par son nom, vous pouvez saisir une partie du nom dans la barre de recherche en haut de ce panneau. Chaque résultat de produit inclut les informations suivantes :

- Le nom et le logo du produit. Les deux sont liés à la page détaillée du produit dans AWS Marketplace. La page détaillée s'ouvre dans un nouvel onglet de votre navigateur. À partir de là, vous pouvez vous abonner au produit image si vous souhaitez l'utiliser dans une recette Image Builder. Pour plus d'informations, consultez la section Acheter des produits dans le Guide de AWS Marketplace l'acheteur.

  Si vous vous abonnez au produit image dans AWS Marketplace, revenez à l'onglet Image Builder de votre navigateur et actualisez la liste des produits d'image abonnés pour le voir.

> **Note**
>
> Quelques minutes peuvent s'écouler avant que votre nouvel abonnement ne soit disponible.

- Le nom de l'éditeur. Ceci est lié à la page détaillée de l'éditeur dans AWS Marketplace. La page détaillée de l'éditeur s'ouvre dans un nouvel onglet de votre navigateur.
- La version du produit.
- Le nombre d'étoiles du produit et les liens directs vers la section d'évaluation de la page détaillée du produit dans AWS Marketplace. La page détaillée s'ouvre dans un nouvel onglet de votre navigateur.
- Les premières lignes de la description du produit.

Juste en dessous de la barre de recherche, vous pouvez voir combien de résultats votre recherche a produit et quel sous-ensemble de ces résultats est actuellement affiché. Vous pouvez utiliser des commandes supplémentaires sur le côté droit du panneau pour ajuster les paramètres relatifs au nombre de produits à afficher simultanément et à l'ordre de tri à appliquer à vos résultats. Vous pouvez également utiliser le contrôle de pagination pour parcourir vos résultats.

Subscriptions

Cet onglet affiche la liste des produits d'image auxquels vous êtes abonné AWS Marketplace. Chaque produit souscrit présente les informations suivantes :

- Nom du produit. Ceci est lié à la page détaillée du produit dans AWS Marketplace. La page détaillée du produit auquel vous êtes abonné s'ouvre dans un nouvel onglet de votre navigateur.
- Le nom de l'éditeur. Ceci est lié à la page détaillée de l'éditeur dans AWS Marketplace. La page détaillée de l'éditeur s'ouvre dans un nouvel onglet de votre navigateur.
- La version du produit à laquelle vous vous êtes abonné.
- Si un composant associé est inclus dans le produit auquel vous êtes abonné, Image Builder affiche un lien vers le détail du AWSTOE composant.

En haut de la page, vous pouvez rechercher un produit spécifique par son nom, ou vous pouvez parcourir vos résultats à l'aide des commandes de pagination. Pour utiliser un produit

abonné comme image de base pour une nouvelle recette, sélectionnez un produit abonné et choisissez Créer une nouvelle recette. Image Builder présélectionne le premier produit de votre liste par défaut.

> **Note**
>
> Si vous recherchez un produit auquel vous venez de vous abonner et qu'il ne figure pas dans la liste, utilisez le bouton d'actualisation en haut de l'onglet pour actualiser vos résultats. Quelques minutes peuvent être nécessaires pour qu'un nouvel abonnement apparaisse dans la liste.

Utiliser un produit AWS Marketplace imagé dans les recettes d'Image Builder

Dans la console Image Builder, vous pouvez créer une nouvelle recette d'image de deux manières à partir de l'un des produits d'image auxquels vous êtes abonné.

1. Vous pouvez commencer à partir de la page des produits Image comme suit :
    1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
    2. Dans le volet de navigation, choisissez Image products dans la AWS Marketplacesection.
    3. Ouvrez l'onglet Abonnements.
    4. Sélectionnez le produit image auquel vous êtes abonné à utiliser comme image de base dans votre recette.
    5. Choisissez Créer une nouvelle recette. Cela ouvre la page Créer une recette avec l'option AWS Marketplace images et le produit image auquel vous êtes abonné est présélectionné.
    6. Configurez les autres paramètres de votre recette comme vous le feriez normalement. Pour plus d'informations sur les recettes d'images, consultezCréation d'une nouvelle version d'une recette d'image.
2. Vous pouvez également ouvrir la page Créer une recette et sélectionner une AWS Marketplace image de produit à utiliser comme image de base.
    1. Ouvrez la console EC2 Image Builder sur https://console.aws.amazon.com/imagebuilder/.
    2. Dans le volet de navigation, choisissez Image recipes dans la AWS Marketplacesection. Cela vous montre une liste des recettes d'images que vous avez créées.

3. Choisissez Créer une recette d'image. Cela ouvre la page Créer une recette.
4. Entrez le nom et la version de votre recette dans la section Détails de la recette, comme d'habitude.
5. Dans la section Image de base, choisissez l'option AWS Marketplace images. Cela vous montre une liste des produits AWS Marketplace d'image auxquels vous êtes abonné dans l'onglet Abonnements. Vous pouvez choisir votre image de base dans la liste.

   Vous pouvez également rechercher d'autres produits d'imagerie disponibles AWS Marketplace directement depuis l'AWS Marketplaceonglet. Choisissez Ajouter des produits ou ouvrez directement l'AWS Marketplaceonglet. Pour plus d'informations sur la définition des filtres et la recherche dans le AWS Marketplace, voir Rechercher des produits AWS Marketplace illustrés à partir de la console Image Builder.
6. Entrez les informations restantes comme d'habitude, puis choisissez Créer une recette.

> **Note**
>
> Si votre abonnement à un produit d'imagerie inclut un AWSTOE composant de génération, vous pouvez le sélectionner dans la liste des composants de génération. Sélectionnez dans `Third party managed` la liste des types de propriétaire du composant pour le voir. Si votre abonnement au produit inclut un composant de AWSTOE test, suivez la même procédure pour la liste des composants de test.

Intégration d'Amazon SNS dans Image Builder

Amazon Simple Notification Service (Amazon SNS) est un service géré qui fournit des messages asynchrones des éditeurs aux abonnés (également appelés producteurs et consommateurs). Vous pouvez spécifier une rubrique SNS dans la configuration de votre infrastructure. Lorsque vous créez une image ou que vous exécutez un pipeline, Image Builder peut publier des messages détaillés sur le statut de votre image dans cette rubrique. Lorsque l'état de l'image atteint l'un des états suivants, Image Builder publie un message :

- `AVAILABLE`
- `FAILED`

Pour un exemple de message SNS d'Image Builder, consultezFormat de message SNS. Si vous souhaitez créer une nouvelle rubrique SNS, consultez Getting started with Amazon SNS dans le manuel Amazon Simple Notification Service Developer Guide.

Rubriques SNS cryptées

Si votre rubrique SNS est cryptée, vous devez autoriser le rôle de service Image Builder dans la AWS KMS key politique à effectuer les actions suivantes :

- `kms:Decrypt`
- `kms:GenerateDataKey`

> **Note**
>
> Si votre rubrique SNS est chiffrée, la clé qui chiffre cette rubrique doit résider dans le compte sur lequel le service Image Builder est exécuté. Image Builder ne peut pas envoyer de notifications aux rubriques SNS chiffrées à l'aide de clés provenant d'autres comptes.

Exemple d'ajout d'une politique de clé KMS

L'exemple suivant montre la section supplémentaire que vous ajoutez à la politique de clé KMS. Utilisez le nom de ressource Amazon (ARN) pour le rôle lié au service IAM qu'Image Builder a créé sous votre compte lorsque vous avez créé une image Image Builder pour la première fois. Pour en savoir plus sur le rôle lié au service Image Builder, consultez. Utilisation de rôles liés à un service pour 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": "*"
  }]
```

```
}
```

Vous pouvez utiliser l'une des méthodes suivantes pour obtenir l'ARN.

AWS Management Console

Pour obtenir l'ARN du rôle lié à un service créé par Image Builder sous votre compte à partir du AWS Management Console, procédez comme suit :

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation de gauche, choisissez Rôles.
3. Recherchez `ImageBuilder` et choisissez le nom de rôle suivant dans les résultats :`AWSServiceRoleForImageBuilder`. Cela affiche la page détaillée du rôle.
4. Pour copier l'ARN dans votre presse-papiers, cliquez sur l'icône à côté du nom de l'ARN.

AWS CLI

Pour obtenir l'ARN du rôle lié au service créé par Image Builder sous votre compte à partir du AWS CLI, utilisez la commande IAM get-role, comme suit.

```
aws iam get-role --role-name AWSServiceRoleForImageBuilder
```

Exemple de sortie partiel :

```
{
    "Role": {
        "Path": "/aws-service-role/imagebuilder.amazonaws.com/",
        "RoleName": "AWSServiceRoleForImageBuilder",
        ...
        "Arn": "arn:aws:iam::123456789012:role/aws-service-role/
imagebuilder.amazonaws.com/AWSServiceRoleForImageBuilder",
        ...
}
```

Format de message SNS

Une fois qu'Image Builder a publié un message sur votre rubrique Amazon SNS, les autres services abonnés à ce sujet peuvent filtrer le format du message et déterminer s'il répond aux critères d'une

action ultérieure. Par exemple, un message de réussite peut lancer une tâche de mise à jour d'un magasin de AWS Systems Manager paramètres ou de lancement d'un flux de travail de test de conformité externe pour l'AMI de sortie.

L'exemple suivant montre la charge utile JSON d'un message type publié par Image Builder lorsqu'une construction de pipeline est terminée et crée une image Linux.


```
{
  "versionlessArn": "arn:aws:imagebuilder:us-west-1:123456789012:image/example-linux-
image",
  "semver": 1237940039285380274899124227,
  "arn": "arn:aws:imagebuilder:us-west-1:123456789012:image/example-linux-
image/1.0.0/3",
  "name": "example-linux-image",
  "version": "1.0.0",
  "type": "AMI",
  "buildVersion": 3,
  "state": {
    "status": "AVAILABLE"
  },
  "platform": "Linux",
  "imageRecipe": {
    "arn": "arn:aws:imagebuilder:us-west-1:123456789012:image-recipe/example-linux-
image/1.0.0",
    "name": "amjule-barebones-linux",
    "version": "1.0.0",
    "components": [
      {
        "componentArn": "arn:aws:imagebuilder:us-west-1:123456789012:component/update-
linux/1.0.2/1"
      }
    ],
    "platform": "Linux",
    "parentImage": "arn:aws:imagebuilder:us-west-1:987654321098:image/amazon-linux-2-
x86/2022.6.14/1",
    "blockDeviceMappings": [
      {
        "deviceName": "/dev/xvda",
        "ebs": {
          "encrypted": false,
          "deleteOnTermination": true,
          "volumeSize": 8,
          "volumeType": "gp2"
```

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

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

L'exemple suivant montre la charge utile JSON d'un message type publié par Image Builder en cas d'échec de la création d'un pipeline pour une image Linux.

```
{
  "versionlessArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-
image",
  "semver": 12379400392853802748991242 31,
  "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-image/1.0.0/7",
  "name": "My Example Image",
  "version": "1.0.0",
  "type": "AMI",
  "buildVersion": 7,
  "state": {
    "status": "FAILED",
    "reason": "Image Failure reason."
  },
  "platform": "Linux",
  "imageRecipe": {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-example-
image/1.0.0",
    "name": "My Example Image",
    "version": "1.0.0",
    "description": "Testing Image recipe",
    "components": [
      {
        "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/my-
example-image-component/1.0.0/1"
      }
    ],
    "platform": "Linux",
    "parentImage": "ami-0cd12345db678d90f",
    "dateCreated": "Jun 21, 2022 11:36:14 PM",
    "tags": {
      "internalId": "1a234567-8901-2345-bcd6-ef7890123456",
      "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-
example-image/1.0.0"
    },
    "accountId": "123456789012"
  },
  "sourcePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/my-
example-image-pipeline",
  "infrastructureConfiguration": {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/
my-example-infra-config",
    "name": "SNS topic Infra config",
    "description": "An example that will retain instances of failed builds",
```

```
    "instanceTypes": [
      "t2.micro"
    ],
    "instanceProfileName": "EC2InstanceProfileForImageBuilder",
    "tags": {
      "internalId": "234abc56-d789-0123-a4e5-6b789d012c34",
      "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-
configuration/my-example-infra-config"
    },
    "terminateInstanceOnFailure": true,
    "snsTopicArn": "arn:aws:sns:us-west-2:123456789012:example-pipeline-notification-
topic",
    "dateCreated": "Jul 5, 2022 7:31:53 PM",
    "accountId": "123456789012"
  },
  "imageTestsConfigurationDocument": {
    "imageTestsEnabled": true,
    "timeoutMinutes": 720
  },
  "distributionConfiguration": {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:distribution-configuration/my-
example-distribution-config",
    "name": "New distribution config",
    "dateCreated": "Dec 3, 2021 9:24:22 PM",
    "distributions": [
      {
        "region": "us-west-2",
        "amiDistributionConfiguration": {},
        "fastLaunchConfigurations": [
          {
            "enabled": true,
            "snapshotConfiguration": {
              "targetResourceCount": 2
            },
            "maxParallelLaunches": 2,
            "launchTemplate": {
              "launchTemplateId": "lt-01234567890"
            },
            "accountId": "123456789012"
          }
        ]
      }
    ],
    "tags": {
```

```
      "internalId": "1fecd23a-4f56-7f89-01e2-345678abbe90",
      "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:distribution-
configuration/my-example-distribution-config"
    },
    "accountId": "123456789012"
  },
  "dateCreated": "Jul 5, 2022 7:40:15 PM",
  "outputResources": {
    "amis": []
  },
  "accountId": "123456789012",
  "enhancedImageMetadataEnabled": true,
  "buildType": "SCHEDULED",
  "tags": {
    "internalId": "456c78b9-0e12-3f45-afb6-7e89b0f1a23b",
    "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-
image/1.0.0/7"
  }
}
```


Produits de conformité pour vos images Image Builder

Compte tenu de l'évolution constante des normes de sécurité, il peut être difficile de maintenir la conformité et de protéger votre entreprise contre les cybermenaces. Pour garantir la conformité de vos images personnalisées, et le rester grâce à des mises à jour automatiques lorsque les éditeurs publient de nouvelles versions, Image Builder s'intègre AWS Marketplace aux produits et AWSTOE composants conformes.

Image Builder s'intègre aux produits de conformité suivants :

- Renforcement des benchmarks du Center for Internet Security (CIS)

  Vous pouvez utiliser les images renforcées CIS et les composants de renforcement CIS associés pour créer des images personnalisées conformes aux dernières directives CIS Benchmarks de niveau 1. Les images renforcées CIS sont disponibles en AWS Marketplace. Pour en savoir plus sur la configuration et l'utilisation des images renforcées CIS et des composants de renforcement, consultez les guides de démarrage rapide sur le portail d'assistance du site Web du CIS.

> **Note**
>
> Lorsque vous vous abonnez à une image renforcée CIS, vous avez également accès au composant de génération associé qui exécute un script pour appliquer les directives CIS Benchmark Level 1 à votre configuration. Pour plus d'informations, consultez composants de durcissement CIS.

- Guides de mise en œuvre technique de sécurité (STIG)

  Pour garantir la conformité aux normes STIG, vous pouvez utiliser des composants STIG gérés par Amazon AWS Task Orchestrator and Executor (AWSTOE) dans vos recettes Image Builder. Les composants STIG analysent votre instance de build pour détecter les erreurs de configuration et exécutent un script de correction pour corriger les problèmes détectés. Nous ne pouvons pas garantir la conformité aux STIG pour les images que vous créez avec Image Builder. Vous devez travailler avec l'équipe de conformité de votre organisation pour vérifier que votre image finale est conforme. Pour obtenir la liste complète des composants AWSTOE STIG que vous pouvez utiliser dans vos recettes Image Builder, consultezAmazon a géré les composants de renforcement STIG pour EC2 Image Builder.

Surveillez les événements et les journaux dans EC2 Image Builder

Pour maintenir la fiabilité, la disponibilité et les performances de vos pipelines EC2 Image Builder, il est important de surveiller les événements et les journaux. Les événements et les journaux vous aident à avoir une vue d'ensemble et à approfondir les détails en cas d'échec d'un appel d'API. Image Builder s'intègre à des services capables d'envoyer des alertes et de déclencher des réponses automatisées lorsque les événements répondent aux critères que vous avez configurés.

Les rubriques suivantes décrivent les techniques de surveillance que vous pouvez utiliser via les services intégrés à Image Builder.

Surveiller les événements et les journaux

- Journalisation des appels d'API EC2 Image Builder à l'aide de AWS CloudTrail

Journalisation des appels d'API EC2 Image Builder à l'aide de AWS CloudTrail

EC2 Image Builder est intégré AWS CloudTrailà un service qui fournit un enregistrement des actions effectuées par tous les appels d'API par un utilisateur, un rôle ou AWS un service via l'API Image Builder. CloudTrail capture Image Builder sous forme d'événements. Les appels capturés incluent des appels provenant de la console Image Builder et des appels de code vers les opérations de l'API Image Builder.

Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment S3, y compris des événements pour Image Builder. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite à Image Builder, l'adresse IP à partir de laquelle la demande a été faite, qui a fait la demande, quand elle a été faite et des informations supplémentaires.

Pour en savoir plus CloudTrail, consultez le guide de AWS CloudTrail l'utilisateur.

Informations sur Image Builder dans CloudTrail

CloudTrail est activé sur votre compte Compte AWS lorsque vous créez le compte. Lorsqu'une activité se produit dans Image Builder, cette activité est enregistrée dans un CloudTrail événement avec d'autres événements de AWS service dans l'historique des événements. Vous pouvez consulter, rechercher et télécharger les événements récents dans votre Compte AWS. Pour plus d'informations, consultez la section Affichage des événements avec l'historique des CloudTrail événements.

Pour un enregistrement continu des événements de votre site Compte AWS, y compris des événements pour Image Builder, créez un parcours. Un journal permet CloudTrail de fournir des fichiers journaux à un compartiment S3. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à toutes les régions Régions AWS. Le journal enregistre les événements de toutes les régions de la AWS partition et transmet les fichiers journaux au compartiment S3 que vous spécifiez. En outre, vous pouvez en configurer d'autres Services AWS pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour plus d'informations, consultez les ressources suivantes :

- Vue d'ensemble de la création d'un parcours.
- CloudTrail services et intégrations pris en charge.
- Configuration des notifications Amazon SNS pour. CloudTrail
- Réception de fichiers CloudTrail journaux provenant de plusieurs régions.
- Réception de fichiers CloudTrail journaux provenant de plusieurs comptes.

CloudTrail enregistre toutes les actions d'Image Builder documentées dans le manuel EC2 Image Builder API Reference. Par exemple, les appels aux `CreateImagePipeline`, `UpdateInfrastructureConfiguration`, et `StartImagePipelineExecution` les actions génèrent des entrées dans les fichiers CloudTrail journaux.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec des informations d'identification d'utilisateur root ou IAM.
- Si la demande a été effectuée avec des informations d'identification de sécurité temporaires pour un rôle ou un utilisateur fédéré.
- Si la demande a été faite par un autre AWS service.

Pour plus d'informations sur la manière de déterminer qui a demandé un événement, consultez l'élément CloudTrail UserIdentity.

Sécurité dans EC2 Image Builder

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez d'un centre de données et d'une architecture réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le modèle de responsabilité partagée décrit cela comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est responsable de la protection de l'infrastructure qui fonctionne Services AWS dans le AWS cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de AWS conformité Programmes de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à EC2 Image Builder, Services AWS voir Scope by Compliance Program.
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation d'Image Builder. Les rubriques suivantes expliquent comment configurer Image Builder pour répondre à vos objectifs de sécurité et de conformité. Vous apprendrez également à en utiliser d'autres Services AWS qui vous aident à surveiller et à sécuriser vos ressources Image Builder.

Rubriques

Protection des données dans EC2 Image Builder

Le modèle de responsabilité AWS partagée s'applique à la protection des données dans EC2 Image Builder. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour en savoir plus sur la confidentialité des données, consultez Questions fréquentes (FAQ) sur la confidentialité des données. Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog Modèle de responsabilité partagée AWS et RGPD (Règlement général sur la protection des données) sur le Blog de sécuritéAWS .

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- Utilisez le protocole SSL/TLS pour communiquer avec les ressources. AWS Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-2 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour en savoir plus sur les points de terminaison FIPS (Federal Information Processing Standard) disponibles, consultez Federal Information Processing Standard (FIPS) 140-2 (Normes de traitement de l'information fédérale).

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Name (Nom). Cela inclut lorsque vous travaillez avec Image Builder ou un autre outil Services AWS à l'aide de la console, de l'API ou AWS des SDK. AWS CLI Toutes les données

que vous saisissez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

Chiffrement et gestion des clés dans EC2 Image Builder

Image Builder chiffre les données en transit et au repos par défaut à l'aide d'une clé KMS appartenant au service, sauf dans les cas suivants :

- Composants personnalisés : Image Builder chiffre les composants personnalisés à l'aide de votre clé KMS par défaut ou d'une clé KMS appartenant au service.
- Flux de travail d'images — Image Builder peut chiffrer vos flux de travail d'images à l'aide d'une clé gérée par le client si vous spécifiez la clé lors de la création du flux de travail. Image Builder gère le chiffrement et le déchiffrement à l'aide de votre clé afin d'exécuter les flux de travail que vous avez configurés pour vos images.

Vous pouvez gérer vos propres clés via AWS KMS. Toutefois, vous n'êtes pas autorisé à gérer la clé KMS Image Builder détenue par Image Builder. Pour plus d'informations sur la gestion de vos clés KMS avec AWS Key Management Service, consultez Getting Started dans le guide du AWS Key Management Service développeur.

Contexte de chiffrement

Pour fournir un contrôle supplémentaire de l'intégrité et de l'authenticité de vos données chiffrées, vous avez la possibilité d'inclure un contexte de chiffrement lorsque vous chiffrez les données. Lorsqu'une ressource est chiffrée avec un contexte de chiffrement, lie AWS KMS cryptographiquement le contexte au texte chiffré. La ressource ne peut être déchiffrée que si le demandeur fournit une correspondance exacte, distinguant majuscules et minuscules, pour le contexte.

Les exemples de politique présentés dans cette section utilisent un contexte de chiffrement similaire à l'Amazon Resource Name (ARN) d'une ressource de flux de travail Image Builder.

Chiffrez les flux de production d'images à l'aide d'une clé gérée par le client

Pour ajouter une couche de protection, vous pouvez chiffrer les ressources de votre flux de travail Image Builder avec votre propre clé gérée par le client. Si vous utilisez votre clé gérée par le client

pour chiffrer les flux de travail Image Builder que vous créez, vous devez autoriser l'accès dans la politique des clés pour qu'Image Builder utilise votre clé lorsqu'il chiffre et déchiffre les ressources du flux de travail. Vous pouvez supprimer cet accès à tout moment. Cependant, Image Builder n'aura accès à aucun flux de travail déjà chiffré si vous révoquez l'accès à la clé.

Le processus pour autoriser Image Builder à utiliser votre clé gérée par le client comporte deux étapes, comme suit :

Étape 1 : ajouter des autorisations politiques clés pour les flux de travail Image Builder

Pour permettre à Image Builder de chiffrer et de déchiffrer les ressources de flux de travail lorsqu'il crée ou utilise ces flux de travail, vous devez spécifier des autorisations dans la politique de clé KMS.

Cet exemple de politique clé accorde l'accès aux pipelines Image Builder afin de chiffrer les ressources du flux de travail pendant le processus de création et de déchiffrer les ressources du flux de travail pour les utiliser. La politique accorde également l'accès aux principaux administrateurs. Le contexte de chiffrement et la spécification des ressources utilisent un caractère générique pour couvrir toutes les régions dans lesquelles vous disposez de ressources de flux de travail.

Comme condition préalable à l'utilisation de flux de travail par image, vous avez créé un rôle d'exécution de flux de travail IAM qui autorise Image Builder à exécuter des actions de flux de travail. Le principal de la première instruction présenté dans l'exemple de politique clé présenté ici doit spécifier votre rôle d'exécution du flux de travail IAM.

Pour plus d'informations sur les clés gérées par le client, consultez la section Gestion de l'accès aux clés gérées par le client dans le Guide du AWS Key Management Service développeur.

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

Étape 2 : Accorder un accès clé à votre rôle d'exécution du flux de travail

Le rôle IAM assumé par Image Builder pour exécuter vos flux de travail nécessite l'autorisation d'utiliser votre clé gérée par le client. Sans accès à votre clé, Image Builder ne sera pas en mesure de chiffrer ou de déchiffrer les ressources de votre flux de travail à l'aide de celle-ci.

Modifiez la politique de votre rôle d'exécution du flux de travail pour ajouter la déclaration de politique suivante.

```
{
 "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 événements pour les flux de production d'images

Les exemples suivants présentent des AWS CloudTrail entrées typiques pour le chiffrement et le déchiffrement des flux de travail d'images stockés avec une clé gérée par le client.

Exemple : GenerateDataKey

Cet exemple montre à quoi peut ressembler un CloudTrail événement lorsqu'Image Builder appelle l'action d' AWS KMS GenerateDataKeyAPI à partir de l'action d'CreateWorkflowAPI Image Builder. Image Builder doit chiffrer un nouveau flux de travail avant de créer la ressource de flux de travail.

```
{
 "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"
}
```


Exemple : Déchiffrer

Cet exemple montre à quoi peut ressembler un CloudTrail événement lorsqu'Image Builder appelle l'action d' AWS KMS DecryptAPI à partir de l'action d'GetWorkflowAPI Image Builder. Les pipelines Image Builder doivent déchiffrer une ressource de flux de travail avant de pouvoir l'utiliser.


```
{
 "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"
}
```


Stockage des données dans EC2 Image Builder

Image Builder ne stocke aucun de vos journaux dans le service. Tous les journaux sont enregistrés sur votre instance Amazon EC2 qui est utilisée pour créer l'image, ou dans vos journaux d'automatisation de Systems Manager.

Confidentialité du trafic interréseau dans EC2 Image Builder

Les connexions sont sécurisées entre Image Builder et les sites sur site, entre les zones d'accès au sein d'une AWS région et entre les AWS régions via HTTPS. Il n'existe aucun lien direct entre les comptes.

Identity and Access Management pour EC2 Image Builder

Rubriques

Public ciblé

La façon dont vous utilisez AWS Identity and Access Management (IAM) varie en fonction du travail que vous effectuez dans Image Builder.

Utilisateur du service : si vous utilisez le service Image Builder pour effectuer votre travail, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Au fur et à mesure que vous utilisez de plus en plus de fonctionnalités d'Image Builder pour effectuer votre travail, vous aurez peut-être besoin d'autorisations supplémentaires. En comprenant bien la gestion des accès, vous saurez demander les autorisations appropriées à votre administrateur. Si vous ne pouvez pas accéder à une fonctionnalité dans Image Builder, consultezRésolution des problèmes d'identité et d'accès à EC2 Image Builder.

Administrateur du service — Si vous êtes responsable des ressources Image Builder dans votre entreprise, vous avez probablement un accès complet à Image Builder. C'est à vous de déterminer les fonctionnalités et les ressources d'Image Builder auxquelles les utilisateurs de votre service doivent accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d'IAM. Pour en savoir plus sur la manière dont votre entreprise peut utiliser IAM avec Image Builder, consultezFonctionnement d'EC2 Image Builder avec IAM.

Administrateur IAM : si vous êtes administrateur IAM, vous souhaiterez peut-être en savoir plus sur la manière dont vous pouvez rédiger des politiques pour gérer l'accès à Image Builder. Pour consulter des exemples de politiques basées sur l'identité Image Builder que vous pouvez utiliser dans IAM, consultez. Politiques basées sur l'identité d'Image Builder

Authentification par des identités

Pour obtenir des informations détaillées sur la manière de fournir une authentification aux personnes et aux processus de votre entreprise Compte AWS, consultez Identités dans le guide de l'utilisateur IAM.

Fonctionnement d'EC2 Image Builder avec IAM

Avant d'utiliser IAM pour gérer l'accès à Image Builder, découvrez quelles fonctionnalités IAM peuvent être utilisées avec Image Builder.

Pour obtenir une vue d'ensemble de la façon dont Image Builder et les autres AWS services fonctionnent avec la plupart des fonctionnalités IAM, consultez les AWS services compatibles avec IAM dans le guide de l'utilisateur IAM.

Politiques basées sur l'identité pour Image Builder

Prend en charge les politiques basées sur l'identité	Oui

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un Groupes d'utilisateurs IAM ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, veuillez consulter Création de politiques IAM dans le Guide de l'utilisateur IAM.

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Vous ne pouvez pas spécifier le principal dans une politique basée sur une identité car celle-ci s'applique à l'utilisateur ou au rôle auquel elle est attachée. Pour découvrir tous les éléments que vous utilisez dans une politique JSON, consultez Références des éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

Exemples de politiques basées sur l'identité pour Image Builder

Pour consulter des exemples de politiques basées sur l'identité d'Image Builder, consultez. Politiques basées sur l'identité d'Image Builder

Politiques basées sur les ressources dans Image Builder

Prend en charge les politiques basées sur les ressources	Non

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Des politiques basées sur les ressources sont, par exemple, les politiques de confiance de rôle IAM et des politiques de compartiment Amazon S3. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée

sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Pour permettre un accès intercompte, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que principal dans une politique basée sur les ressources. L'ajout d'un principal entre comptes à une politique basée sur les ressources ne représente qu'une partie de l'instauration de la relation d'approbation. Lorsque le principal et la ressource sont différents Comptes AWS, un administrateur IAM du compte sécurisé doit également accorder à l'entité principale (utilisateur ou rôle) l'autorisation d'accéder à la ressource. Pour ce faire, il attache une politique basée sur une identité à l'entité. Toutefois, si une politique basée sur des ressources accorde l'accès à un principal dans le même compte, aucune autre politique basée sur l'identité n'est requise. Pour plus d'informations, consultez Différence entre les rôles IAM et les politiques basées sur une ressource dans le Guide de l'utilisateur IAM.

Actions politiques pour Image Builder

Prend en charge les actions de politique	Oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Les actions de stratégie portent généralement le même nom que l'opération AWS d'API associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une stratégie afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des actions d'Image Builder, consultez la section Actions définies par EC2 Image Builder dans le Service Authorization Reference.

Les actions de stratégie dans Image Builder utilisent le préfixe suivant avant l'action :

```
imagebuilder
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

```
"Action": [
      "imagebuilder:action1",
      "imagebuilder:action2"
         ]
```

Pour consulter des exemples de politiques basées sur l'identité d'Image Builder, consultez. Politiques basées sur l'identité d'Image Builder

Ressources relatives aux politiques pour Image Builder

Prend en charge les ressources de politique	Oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets pour lesquels l'action s'applique. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de définir une ressource à l'aide de son Amazon Resource Name (ARN). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

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

Pour consulter la liste des types de ressources Image Builder et de leurs ARN, consultez la section Ressources définies par EC2 Image Builder dans le Service Authorization Reference. Pour savoir avec quelles actions vous pouvez spécifier l'ARN de chaque ressource, voir Actions définies par EC2 Image Builder.

Pour consulter des exemples de politiques basées sur l'identité d'Image Builder, consultez. Politiques basées sur l'identité d'Image Builder

Clés de conditions de politique pour Image Builder

Prend en charge les clés de condition de politique spécifiques au service	Oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des opérateurs de condition, tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération `AND` logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une `OR` opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez accorder à un utilisateur IAM l'autorisation d'accéder à une ressource uniquement si elle est balisée avec son nom d'utilisateur IAM. Pour plus d'informations, consultez Éléments d'une politique IAM : variables et identifications dans le Guide de l'utilisateur IAM.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. Pour voir toutes les clés de condition AWS globales, voir les clés de contexte de condition AWS globales dans le guide de l'utilisateur IAM.

Pour consulter la liste des clés de condition d'Image Builder, consultez la section Clés de condition pour EC2 Image Builder dans le Service Authorization Reference. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, voir Actions définies par EC2 Image Builder.

Pour consulter des exemples de politiques basées sur l'identité d'Image Builder, consultez. Politiques basées sur l'identité d'Image Builder

ACL dans Image Builder

Prend en charge les listes ACL	Non

Les listes de contrôle d'accès (ACL) vérifient quels principaux (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

ABAC avec Image Builder

Prise en charge d'ABAC (identifications dans les politiques)	Partielle

Le contrôle d'accès basé sur les attributs (ABAC) est une politique d'autorisation qui définit des autorisations en fonction des attributs. Dans AWS, ces attributs sont appelés balises. Vous pouvez associer des balises aux entités IAM (utilisateurs ou rôles) et à de nombreuses AWS ressources. L'étiquetage des entités et des ressources est la première étape d'ABAC. Vous concevez ensuite des politiques ABAC pour autoriser des opérations quand l'identification du principal correspond à celle de la ressource à laquelle il tente d'accéder.

L'ABAC est utile dans les environnements qui connaissent une croissance rapide et pour les cas où la gestion des politiques devient fastidieuse.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'élément de condition d'une politique utilisant les clés de condition `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informations sur l'ABAC, consultez Qu'est-ce que le contrôle d'accès basé sur les attributs (ABAC) ? dans le Guide de l'utilisateur IAM. Pour accéder à un didacticiel décrivant les étapes de configuration de l'ABAC, consultez Utilisation du contrôle d'accès basé sur les attributs (ABAC) dans le Guide de l'utilisateur IAM.

Utilisation d'informations d'identification temporaires avec Image Builder

Prend en charge les informations d'identification temporaires	Oui

Certains Services AWS ne fonctionnent pas lorsque vous vous connectez à l'aide d'informations d'identification temporaires. Pour plus d'informations, y compris celles qui Services AWS fonctionnent avec des informations d'identification temporaires, consultez Services AWS la section relative à l'utilisation d'IAM dans le guide de l'utilisateur d'IAM.

Vous utilisez des informations d'identification temporaires si vous vous connectez à l' AWS Management Console aide d'une méthode autre qu'un nom d'utilisateur et un mot de passe. Par exemple, lorsque vous accédez à AWS l'aide du lien d'authentification unique (SSO) de votre entreprise, ce processus crée automatiquement des informations d'identification temporaires. Vous créez également automatiquement des informations d'identification temporaires lorsque vous vous connectez à la console en tant qu'utilisateur, puis changez de rôle. Pour plus d'informations sur le changement de rôle, consultez Changement de rôle (console) dans le Guide de l'utilisateur IAM.

Vous pouvez créer manuellement des informations d'identification temporaires à l'aide de l' AWS API AWS CLI or. Vous pouvez ensuite utiliser ces informations d'identification temporaires pour y accéder AWS. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez Informations d'identification de sécurité temporaires dans IAM.

Autorisations principales interservices pour Image Builder

Prend en charge les transmissions de sessions d'accès (FAS)	Oui

Lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, l'action que vous effectuez est susceptible de lancer une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux

actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez Transmission des sessions d'accès.

Rôles de service pour Image Builder

Prend en charge les fonctions de service	Oui

Une fonction du service est un rôle IAM qu'un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.

> **Warning**
>
> La modification des autorisations associées à un rôle de service peut perturber les fonctionnalités d'Image Builder. Modifiez les rôles de service uniquement lorsque Image Builder fournit des instructions à cet effet.

Rôles liés à un service pour Image Builder

Prend en charge les rôles liés à un service	Non

Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Pour plus de détails sur le rôle lié au service Image Builder, consultez. Utilisation de rôles liés à un service pour EC2 Image Builder

Politiques basées sur l'identité d'Image Builder

Avec les politiques basées sur l'identité IAM, vous pouvez spécifier des actions et des ressources autorisées ou refusées ainsi que les conditions selon lesquelles les actions sont autorisées ou refusées. Image Builder prend en charge des actions, des ressources et des clés de condition

spécifiques. Pour plus d'informations sur tous les éléments que vous utilisez dans une politique JSON, consultez la section Actions, ressources et clés de condition pour Amazon EC2 Image Builder dans le guide de l'utilisateur IAM.

Actions

Les actions de politique dans Image Builder utilisent le préfixe suivant avant l'action :`imagebuilder:`. Les déclarations de politique doivent inclure un élément `Action` ou `NotAction`. Image Builder définit son propre ensemble d'actions décrivant les tâches que vous pouvez effectuer avec ce service.

Pour spécifier plusieurs actions dans une seule déclaration, séparez-les par des virgules comme suit :

```
"Action": [
      "imagebuilder:action1",
      "imagebuilder:action2"
```

Vous pouvez aussi spécifier plusieurs actions à l'aide de caractères génériques (*). Par exemple, pour spécifier toutes les actions qui commencent par le mot `List`, incluez l'action suivante :

```
"Action": "imagebuilder:List*"
```

Pour consulter la liste des actions d'Image Builder, reportez-vous à la section Actions, ressources et clés de Services AWS condition du guide de l'utilisateur IAM.

Gestion des accès à l'aide de politiques

Pour obtenir des informations détaillées sur la façon de gérer l'accès en AWS créant des politiques et en les associant aux identités ou aux AWS ressources IAM, consultez la section Politiques et autorisations du guide de l'utilisateur IAM.

Le rôle IAM que vous associez à votre profil d'instance doit être autorisé à exécuter les composants de génération et de test inclus dans votre image. Les politiques de rôle IAM suivantes doivent être associées au rôle IAM associé au profil d'instance :

- EC2InstanceProfileForImageBuilder
- EC2InstanceProfileForImageBuilderECRContainerBuilds
- AmazonSSMManagedInstanceCore

Ressources

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets pour lesquels l'action s'applique. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de définir une ressource à l'aide de son Amazon Resource Name (ARN). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

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

La ressource d'instance Image Builder possède le nom de ressource Amazon (ARN) suivant.

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

Pour plus d'informations sur le format des ARN, consultez Amazon Resource Names (ARN) et AWS Service Namespaces.

Par exemple, pour spécifier l'`i-1234567890abcdef0`instance dans votre instruction, utilisez l'ARN suivant.

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

Pour spécifier toutes les instances qui appartiennent à un compte spécifique, utilisez le caractère générique (*).

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

Certaines actions d'Image Builder, telles que celles relatives à la création de ressources, ne peuvent pas être effectuées sur une ressource spécifique. Dans ces cas-là, vous devez utiliser le caractère générique (*).

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

De nombreuses actions de l'API EC2 Image Builder impliquent plusieurs ressources. Pour spécifier plusieurs ressources dans une seule instruction, séparez leurs ARN par des virgules.

```
"Resource": [
      "resource1",
      "resource2"
```

Clés de condition

Image Builder fournit des clés de condition spécifiques au service et prend en charge l'utilisation de certaines clés de condition globales. Pour voir toutes les clés de condition AWS globales, consultez la section Clés contextuelles de condition AWS globale dans le guide de l'utilisateur IAM. Les clés de condition spécifiques au service suivantes sont fournies.

générateur d'images : CreatedResourceTagKeys

Fonctionne avec des opérateurs de chaîne.

Utilisez cette clé pour filtrer l'accès en fonction de la présence de clés de balise dans la demande. Cela vous permet de gérer les ressources créées par Image Builder.

Disponibilité — Cette clé n'est disponible que pour les `UpdateInfrastructureConfiguration` API `CreateInfrastrucutreConfiguration` et.

générateur d'images :/CreatedResourceTag<key>

Fonctionne avec des opérateurs de chaîne.

Utilisez cette clé pour filtrer l'accès en fonction des paires clé-valeur de balise associées à la ressource créée par Image Builder. Cela vous permet de gérer les ressources d'Image Builder via des balises définies.

Disponibilité — Cette clé n'est disponible que pour les `UpdateInfrastructureConfiguration` API `CreateInfrastrucutreConfiguration` et.

Générateur d'images : EC2 MetadataHttpTokens

Fonctionne avec des opérateurs de chaîne.

Utilisez cette clé pour filtrer l'accès en fonction de l'exigence de jeton HTTP de métadonnées d'instance EC2 spécifiée dans la demande.

La valeur de cette clé peut être l'une `optional` ou l'autre`required`.

Disponibilité — Cette clé n'est disponible que pour les `UpdateInfrastructureConfiguration` API `CreateInfrastrucutreConfiguration` et.

générateur d'images : StatusTopicArn

Fonctionne avec des opérateurs de chaîne.

Utilisez cette touche pour filtrer l'accès par l'ARN de la rubrique SNS dans la demande à laquelle les notifications d'état du terminal seront publiées.

Disponibilité — Cette clé n'est disponible que pour les `UpdateInfrastructureConfiguration` API `CreateInfrastrucutreConfiguration` et.

Exemples

Pour consulter des exemples de politiques basées sur l'identité d'Image Builder, consultez. Politiques basées sur l'identité d'EC2 Image Builder

Politiques basées sur les ressources d'Image Builder

Les politiques basées sur les ressources spécifient les actions qu'un principal spécifié peut effectuer sur la ressource Image Builder et dans quelles conditions. Image Builder prend en charge les politiques d'autorisation basées sur les ressources pour les composants, les images et les recettes d'images. Les politiques basées sur les ressources permettent d'accorder une autorisation à d'autres comptes en fonction des ressources. Vous pouvez également utiliser une politique basée sur les ressources pour autoriser un AWS service à accéder à vos composants, images et recettes d'images.

Pour plus d'informations sur la façon d'associer une politique basée sur les ressources à un composant, à une image ou à une recette d'image, consultez. Partagez les ressources d'EC2 Image Builder

> **Note**
>
> Lorsque vous mettez à jour une politique de ressources à l'aide d'Image Builder, la mise à jour apparaît dans la console RAM.

Autorisation basée sur les balises Image Builder

Vous pouvez associer des balises aux ressources d'Image Builder ou transmettre des balises dans une demande à Image Builder. Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'élément de condition d'une politique utilisant les clés de condition `imagebuilder:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`. Pour plus d'informations sur le balisage des ressources Image Builder, consultezMarquer une ressource (AWS CLI).

Rôles IAM dans Image Builder

Un rôle IAM est une entité au sein de vous Compte AWS qui possède des autorisations spécifiques.

Utilisation d'informations d'identification temporaires avec Image Builder

Vous pouvez utiliser des informations d'identification temporaires pour vous connecter à l'aide de la fédération, endosser un rôle IAM ou encore pour endosser un rôle intercompte. Vous obtenez des informations d'identification de sécurité temporaires en appelant des opérations d' AWS STS API telles que AssumeRoleou GetFederationToken.

Rôles liés à un service

Les rôles liés à un service permettent Services AWS d'accéder aux ressources d'autres services pour effectuer une action en votre nom. Les rôles liés à un service s'affichent dans votre compte IAM et sont la propriété du service. Un utilisateur disposant d'un accès administratif peut consulter mais pas modifier les autorisations pour les rôles liés à un service.

Image Builder prend en charge les rôles liés à un service. Pour plus d'informations sur la création ou la gestion des rôles liés au service Image Builder, consultez. Utilisation de rôles liés à un service pour EC2 Image Builder

Rôles de service

Cette fonction permet à un service d'endosser une fonction du service en votre nom. Ce rôle autorise le service à accéder à des ressources d'autres services pour effectuer une action en votre nom. Les fonctions du service s'affichent dans votre compte IAM et sont la propriété du compte. Cela signifie qu'un utilisateur disposant d'un accès administratif peut modifier les autorisations associées à ce rôle. Toutefois, une telle action peut perturber le bon fonctionnement du service.

Politiques basées sur l'identité d'EC2 Image Builder

Rubriques

Bonnes pratiques en matière de politiques basées sur l'identité

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer des ressources Image Builder dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour de plus amples informations, consultez Politiques gérées AWS ou Politiques gérées AWS pour les activités professionnelles dans le Guide de l'utilisateur IAM.
- Accorder les autorisations de moindre privilège - Lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation de IAM pour appliquer des autorisations, consultez Politiques et autorisations dans IAM dans le Guide de l'utilisateur IAM.
- Utiliser des conditions dans les politiques IAM pour restreindre davantage l'accès - Vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que AWS CloudFormation. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.
- Utilisez IAM Access Analyzer pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles - IAM Access Analyzer valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les

bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour de plus amples informations, consultez Validation de politique IAM Access Analyzer dans le Guide de l'utilisateur IAM.

- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue. Compte AWS Pour exiger le MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour de plus amples informations, consultez Configuration de l'accès aux API protégé par MFA dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.

Utilisation de la console Image Builder

Pour accéder à la console EC2 Image Builder, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations vous permettent de répertorier et d'afficher des informations détaillées sur les ressources Image Builder de votre Compte AWS. Si vous créez une politique basée sur l'identité qui est plus restrictive que les autorisations minimales requises, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs et rôles IAM) tributaires de cette politique.

Pour garantir que vos entités IAM peuvent utiliser la console Image Builder, vous devez leur associer l'une des politiques AWS gérées suivantes :

- Stratégie AWSImageBuilderReadOnlyAccess
- Stratégie AWSImageBuilderFullAccess

Pour plus d'informations sur les politiques gérées par Image Builder, consultezUtilisation de politiques gérées pour EC2 Image Builder.

> **Important**
>
> La AWSImageBuilderFullAccesspolitique est requise pour créer le rôle lié au service Image Builder. Lorsque vous associez cette politique à une entité IAM, vous devez également associer la stratégie personnalisée suivante et inclure les ressources que vous souhaitez utiliser et qui ne figurent pas `imagebuilder` dans le nom de la ressource :

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

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API AWS CLI ou l' AWS API. Autorisez plutôt l'accès à uniquement aux actions qui correspondent à l'opération d'API que vous tentez d'effectuer.

Politiques basées sur les ressources d'EC2 Image Builder

Pour plus d'informations sur la création d'un composant, consultezGérez les composants avec Image Builder.

Restreindre l'accès aux composants Image Builder à des adresses IP spécifiques

L'exemple suivant autorise tout utilisateur à effectuer des opérations Image Builder sur des composants. Toutefois, la demande doit provenir de la plage d'adresses IP indiquée dans la condition.

La condition dans cette instruction identifie la plage 54.240.143.* d'adresses Internet Protocol version 4 (IPv4) autorisées, avec une exception : 54.240.143.188.

Le `Condition` bloc utilise les `NotIpAddress` conditions `IpAddress` et et la clé de `aws:SourceIp` condition, qui est une clé AWS de condition étendue. Pour plus d'informations sur ces clés de condition, consultez la section Spécification de conditions dans une politique. Les valeurs IPv4 `aws:sourceIp` font appel à la notation CIDR standard. Pour en savoir plus, consultez Opérateurs de condition d'adresse IP dans le guide de l'utilisateur 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"}
      }
    }
  ]
}
```

Utilisation de politiques gérées pour EC2 Image Builder

Une politique AWS gérée est une politique autonome créée et administrée par AWS. AWS les politiques gérées sont conçues pour fournir des autorisations pour de nombreux cas d'utilisation courants afin que vous puissiez commencer à attribuer des autorisations aux utilisateurs, aux groupes et aux rôles.

N'oubliez pas que les politiques AWS gérées peuvent ne pas accorder d'autorisations de moindre privilège pour vos cas d'utilisation spécifiques, car elles sont accessibles à tous les AWS clients. Nous vous recommandons de réduire encore les autorisations en définissant des politiques gérées par le client qui sont propres à vos cas d'utilisation.

Vous ne pouvez pas modifier les autorisations définies dans les politiques AWS gérées. Si les autorisations définies dans une politique AWS gérée sont AWS mises à jour, la mise à jour affecte toutes les identités principales (utilisateurs, groupes et rôles) auxquelles la politique est attachée. AWS est le plus susceptible de mettre à jour une politique AWS gérée lorsqu'une nouvelle politique Service AWS est lancée ou lorsque de nouvelles opérations d'API sont disponibles pour les services existants.

Pour plus d'informations, consultez la section Politiques gérées par AWS dans le Guide de l'utilisateur IAM.

Stratégie AWSImageBuilderFullAccess

La AWSImageBuilderFullAccesspolitique accorde un accès complet aux ressources d'Image Builder pour le rôle auquel il est rattaché, ce qui permet au rôle de répertorier, de décrire, de créer, de mettre à jour et de supprimer les ressources d'Image Builder. La politique accorde également des autorisations ciblées aux personnes associées Services AWS qui sont nécessaires, par exemple, pour vérifier les ressources ou pour afficher les ressources actuelles du compte dans le AWS Management Console.

Détails de l'autorisation

Cette politique inclut les autorisations suivantes :

- Image Builder : l'accès administratif est accordé afin que le rôle puisse répertorier, décrire, créer, mettre à jour et supprimer des ressources Image Builder.
- Amazon EC2 — L'accès est accordé pour les actions Amazon EC2 Descrivez nécessaires pour vérifier l'existence des ressources ou obtenir des listes de ressources appartenant au compte.

- IAM — L'accès est accordé pour obtenir et utiliser des profils d'instance dont le nom contient « imagebuilder », pour vérifier l'existence du rôle lié au service Image Builder via l'action `iam:GetRole` API et pour créer le rôle lié au service Image Builder.
- License Manager — L'accès est accordé pour répertorier les configurations de licence ou les licences pour une ressource.
- Amazon S3 — L'accès est accordé pour répertorier les buckets appartenant au compte, ainsi que les buckets Image Builder dont le nom contient « imagebuilder ».
- Amazon SNS — Des autorisations d'écriture sont accordées à Amazon SNS pour vérifier la propriété des rubriques contenant « imagebuilder ».

Exemple de politique

Voici un exemple de cette AWSImageBuilderFullAccess politique.

```
{
    "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": "*"
        }
    ]
}
```

Stratégie AWSImageBuilderReadOnlyAccess

La AWSImageBuilderReadOnlyAccesspolitique fournit un accès en lecture seule à toutes les ressources d'Image Builder. Des autorisations sont accordées pour vérifier que le rôle lié au service Image Builder existe via l'action d'`iam:GetRole`API.

Détails de l'autorisation

Cette politique inclut les autorisations suivantes :

- Image Builder : l'accès est accordé pour un accès en lecture seule aux ressources Image Builder.
- IAM — L'accès est accordé pour vérifier l'existence du rôle lié au service Image Builder via l'`iam:GetRole`action API.

Exemple de politique

Voici un exemple de cette AWSImageBuilderReadOnlyAccess politique.

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

Stratégie AWSServiceRoleForImageBuilder

La AWSServiceRoleForImageBuilderpolitique autorise Image Builder à appeler Services AWS en votre nom.

Détails de l'autorisation

Cette politique est attachée au rôle lié au service Image Builder lorsque le rôle est créé via Systems Manager. Pour consulter les autorisations spécifiques accordées, consultez l'exemple de politique présenté dans cette section. Pour plus d'informations sur le rôle lié au service Image Builder, consultez. Utilisation de rôles liés à un service pour EC2 Image Builder

La politique inclut les autorisations suivantes :

- CloudWatch Journaux — L'accès est accordé pour créer et télécharger CloudWatch des journaux dans tout groupe de journaux dont le nom commence par`/aws/imagebuilder/`.
- Amazon EC2 — L'accès est accordé à Image Builder pour créer des images et lancer des instances EC2 dans votre compte, en utilisant les instantanés, les volumes, les interfaces réseau, les sous-réseaux, les groupes de sécurité, la configuration des licences et les paires de clés associés selon les besoins, à condition que l'image, l'instance et les volumes créés ou utilisés soient balisés avec ou. `CreatedBy: EC2 Image Builder` `CreatedBy: EC2 Fast Launch`

  Image Builder peut obtenir des informations sur les images Amazon EC2, les attributs des instances, le statut des instances, les types d'instances disponibles pour votre compte, les modèles de lancement, les sous-réseaux, les hôtes et les balises de vos ressources Amazon EC2.

  Image Builder peut mettre à jour les paramètres de l'image afin d'activer ou de désactiver le lancement plus rapide des instances Windows dans votre compte, où l'image est étiquetée avec`CreatedBy: EC2 Image Builder`.

  En outre, Image Builder peut démarrer, arrêter et mettre fin à des instances qui s'exécutent sur votre compte, partager des instantanés Amazon EBS, créer et mettre à jour des images et des modèles de lancement, annuler l'enregistrement d'images existantes, ajouter des balises et répliquer des images sur des comptes auxquels vous avez accordé des autorisations via la politique. Ec2ImageBuilderCrossAccountDistributionAccess Le balisage Image Builder est obligatoire pour toutes ces actions, comme décrit précédemment.
- Amazon ECR — L'accès est accordé à Image Builder pour créer un référentiel si nécessaire pour les analyses de vulnérabilité des images de conteneurs, et pour étiqueter les ressources qu'il crée afin de limiter la portée de ses opérations. Image Builder est également autorisé à

supprimer les images de conteneur qu'il a créées pour les scans après avoir pris des instantanés des vulnérabilités.

- EventBridge— L'accès est accordé à Image Builder pour créer et gérer EventBridge des règles.
- IAM — L'accès est accordé à Image Builder pour transmettre n'importe quel rôle de votre compte à Amazon EC2 et à VM Import/Export.
- Amazon Inspector — L'accès est accordé à Image Builder pour déterminer quand Amazon Inspector termine les scans des instances de build et pour recueillir les résultats des images configurées de manière à le permettre.
- AWS KMS— L'accès est accordé à Amazon EBS pour chiffrer, déchiffrer ou rechiffrer les volumes Amazon EBS. Cela est essentiel pour garantir que les volumes chiffrés fonctionnent lorsque Image Builder crée une image.
- License Manager — L'accès est accordé à Image Builder pour mettre à jour les spécifications du License Manager via`license-manager:UpdateLicenseSpecificationsForResource`.
- Amazon SNS — Les autorisations d'écriture sont accordées pour toutes les rubriques Amazon SNS de votre compte.
- Systems Manager — L'accès est accordé à Image Builder pour répertorier les commandes de Systems Manager et leurs invocations, les entrées d'inventaire, décrire les informations sur les instances et les statuts d'exécution de l'automatisation, et obtenir les détails de l'invocation des commandes. Image Builder peut également envoyer des signaux d'automatisation et arrêter les exécutions automatisées pour toutes les ressources de votre compte.

  Image Builder est capable d'envoyer des appels de commande d'exécution à n'importe quelle instance étiquetée `"CreatedBy": "EC2 Image Builder"` pour les fichiers de script suivants : `AWS-RunPowerShellScriptAWS-RunShellScript`, ou. `AWSEC2-RunSysprep` Image Builder est capable de lancer une exécution automatisée de Systems Manager dans votre compte pour les documents d'automatisation dont le nom commence par`ImageBuilder`.

  Image Builder est également capable de créer ou de supprimer des associations State Manager pour n'importe quelle instance de votre compte, à condition que le document d'association le soit`AWS-GatherSoftwareInventory`, et de créer le rôle lié au service Systems Manager dans votre compte.
- AWS STS— L'accès est accordé à Image Builder pour assumer les rôles nommés EC2ImageBuilderDistributionCrossAccountRoledepuis votre compte sur n'importe quel compte lorsque la politique de confiance relative au rôle l'autorise. Ceci est utilisé pour la distribution d'images entre comptes.

Exemple de politique

Voici un exemple de cette AWSServiceRoleForImageBuilder politique.

```
{
    "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-*"
            ]
        }
    ]
}
```

Stratégie Ec2ImageBuilderCrossAccountDistributionAccess

La Ec2ImageBuilderCrossAccountDistributionAccesspolitique autorise Image Builder à distribuer des images entre les comptes des régions cibles. Image Builder peut également décrire, copier et appliquer des balises à n'importe quelle image Amazon EC2 du compte. La politique donne également la possibilité de modifier les autorisations de l'AMI via l'action de l'`ec2:ModifyImageAttribute`API.

Détails de l'autorisation

Cette politique inclut les autorisations suivantes :

- Amazon EC2 — L'accès est accordé à Amazon EC2 pour décrire, copier et modifier les attributs d'une image, et pour créer des balises pour toutes les images Amazon EC2 présentes dans le compte.

Exemple de politique

Voici un exemple de cette Ec2ImageBuilderCrossAccountDistributionAccess politique.

```
{
    "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": "*"
        }
    ]
}
```

Stratégie EC2ImageBuilderLifecycleExecutionPolicy

La EC2ImageBuilderLifecycleExecutionPolicypolitique autorise Image Builder à effectuer des actions telles que la dépréciation, la désactivation ou la suppression des ressources d'image Image Builder et de leurs ressources sous-jacentes (AMI, instantanés) afin de prendre en charge les règles automatisées pour les tâches de gestion du cycle de vie des images.

Détails de l'autorisation

Cette politique inclut les autorisations suivantes :

- Amazon EC2 — L'accès est accordé à Amazon EC2 pour effectuer les actions suivantes pour les Amazon Machine Images (AMI) du compte associé au tag. `CreatedBy: EC2 Image Builder`
  - Activez et désactivez une AMI.
  - Activez et désactivez la dépréciation des images.
  - Décrivez et annulez l'enregistrement d'une AMI.
  - Décrivez et modifiez les attributs d'image de l'AMI.

  - Supprimez les instantanés de volume associés à l'AMI.
  - Récupérez les balises d'une ressource.
  - Ajoutez ou supprimez des balises dans une AMI par souci de dépréciation.
- Amazon ECR — L'accès est accordé à Amazon ECR pour effectuer les actions par lots suivantes sur les référentiels ECR dotés de la balise. `LifecycleExecutionAccess: EC2 Image Builder` Les actions par lots prennent en charge les règles de cycle de vie automatisées des images des conteneurs.
  - `ecr:BatchGetImage`
  - `ecr:BatchDeleteImage`

  L'accès est accordé au niveau du référentiel pour les référentiels ECR étiquetés avec. `LifecycleExecutionAccess: EC2 Image Builder`
- AWS Groupes de ressources — L'accès est accordé à Image Builder pour obtenir des ressources en fonction de balises.
- EC2 Image Builder — L'accès est accordé à Image Builder pour supprimer les ressources d'image Image Builder.

Exemple de politique

Voici un exemple de cette EC2ImageBuilderLifecycleExecutionPolicy politique.

```
{
    "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": "*"
        }
    ]
}
```

Stratégie EC2InstanceProfileForImageBuilder

La EC2InstanceProfileForImageBuilderpolitique accorde les autorisations minimales requises pour qu'une instance EC2 fonctionne avec Image Builder. Cela n'inclut pas les autorisations requises pour utiliser l'agent Systems Manager.

Détails de l'autorisation

Cette politique inclut les autorisations suivantes :

- CloudWatch Journaux — L'accès est accordé pour créer et télécharger CloudWatch des journaux dans tout groupe de journaux dont le nom commence par`/aws/imagebuilder/`.
- Image Builder — L'accès est accordé pour obtenir n'importe quel composant Image Builder.
- AWS KMS— L'accès est accordé pour déchiffrer un composant Image Builder, s'il a été chiffré via AWS KMS.
- Amazon S3 — L'accès est accordé pour obtenir des objets stockés dans un compartiment Amazon S3 dont le nom commence par`ec2imagebuilder-`.

Exemple de politique

Voici un exemple de cette EC2InstanceProfileForImageBuilder politique.

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

Stratégie EC2InstanceProfileForImageBuilderECRContainerBuilds

La EC2InstanceProfileForImageBuilderECRContainerBuildspolitique accorde les autorisations minimales requises pour une instance EC2 lorsque vous travaillez avec Image Builder pour créer des images Docker, puis enregistrer et stocker les images dans un référentiel de conteneurs Amazon ECR. Cela n'inclut pas les autorisations requises pour utiliser l'agent Systems Manager.

Détails de l’autorisation

Cette politique inclut les autorisations suivantes :

- CloudWatch Journaux — L'accès est accordé pour créer et télécharger CloudWatch des journaux dans tout groupe de journaux dont le nom commence par`/aws/imagebuilder/`.
- Amazon ECR — L'accès est accordé à Amazon ECR pour obtenir, enregistrer et stocker une image de conteneur, ainsi que pour obtenir un jeton d'autorisation.
- Image Builder — L'accès est accordé pour obtenir un composant Image Builder ou une recette de conteneur.
- AWS KMS— L'accès est accordé pour déchiffrer un composant ou une recette de conteneur Image Builder, s'il a été chiffré via AWS KMS.
- Amazon S3 — L'accès est accordé pour obtenir des objets stockés dans un compartiment Amazon S3 dont le nom commence par`ec2imagebuilder-`.

Exemple de politique

Voici un exemple de cette EC2InstanceProfileForImageBuilderECRContainerBuilds politique.

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

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

Image Builder met à jour les politiques AWS gérées

Cette section fournit des informations sur les mises à jour apportées aux politiques AWS gérées pour Image Builder depuis que ce service a commencé à suivre ces modifications. Pour recevoir des

alertes automatiques concernant les modifications apportées à cette page, abonnez-vous au flux RSS sur la page d'historique du document Image Builder.

Modification	Description	Date
EC2ImageBuilderLifecycleExecutionPolicy – Nouvelle politique	Image Builder a ajouté la nouvelle `EC2ImageBuilderLifecycleExecutionPolicy` politique qui contient des autorisations pour la gestion du cycle de vie des images.	17 novembre 2023
AWSServiceRoleForImageBuilder - mise à jour d'une politique existante	Image Builder a apporté les modifications suivantes au rôle de service afin de fournir la prise en charge de macOS. • Ajout d'ec2 : DescribeHosts permet à Image Builder d'interroger l'HostID afin de déterminer s'il est dans un état valide pour lancer une instance. • Ajout de l'action d'API ssm :GetCommandInvocation, pour améliorer la méthode utilisée par Image Builder pour obtenir les détails de l'appel de commande.	28 août 2023
AWSServiceRoleForImageBuilder - mise à jour d'une politique existante	Image Builder a apporté les modifications suivantes au rôle de service afin de	30 mars 2023

Modification	Description	Date
	permettre aux flux de travail Image Builder de collecter des informations sur les vulnérabilités pour les builds d'images de conteneur AMI et ECR. Les nouvelles autorisations prennent en charge la fonctionnalité de détection et de signalement des CVE. • Ajout de inspector2 : ListCoverage et inspector2 : pour permettre à ListFindings Image Builder de déterminer à quel moment Amazon Inspector effectue les scans des instances de test et de recueillir les résultats des images configurées pour l'autoriser. • Ajout de ecr :CreateRepository, avec l'obligation pour Image Builder de baliser le référentiel avec `CreatedBy: EC2 Image Builder` (tag-on-create). Nous avons également ajouté ecr : TagResource (required for tag-on-create) avec la même contrainte de CreatedBy balise, et une contrainte supplémentaire qui nécessite de commencer `image-bui`	

Modification	Description	Date
	`lder-*` par le nom du dépôt. La contrainte de nom empêche l'augment ation des privilèges et empêche les modifications des référentiels qu'Image Builder n'a pas créés. • Ajout de ecr : BatchDele teImage pour les référenti els ECR étiquetés avec. `CreatedBy: EC2 Image Builder` Cette autorisation nécessite de commencer par le nom du référentiel`image-bui lder-*` . • Ajout d'autorisations d'événement permettant à Image Builder de créer et de gérer les règles EventBrid ge gérées par Amazon qui figurent `ImageBuilder-*` dans le nom.	

Modification	Description	Date
AWSServiceRoleForImageBuilder - mise à jour d'une politique existante	Image Builder a apporté les modifications suivantes au rôle de service : • Ajout de licences License Manager en tant que ressource pour l'RunInstance appel ec2 : afin de permettre aux clients d'utiliser des AMI d'image de base associées à une configuration de licence.	22 mars 2022
AWSServiceRoleForImageBuilder - mise à jour d'une politique existante	Image Builder a apporté les modifications suivantes au rôle de service : • Des autorisations ont été ajoutées pour EnableFastLaunch l'action de l'API EC2, afin d'activer et de désactiver le lancement plus rapide pour les instances Windows. • Champ d'application plus restreint pour ec2 : conditions des balises CreateTags d'action et de ressource.	21 février 2022

Modification	Description	Date
AWSServiceRoleForImageBuilder - mise à jour d'une politique existante	Image Builder a apporté les modifications suivantes au rôle de service : • Des autorisations ont été ajoutées pour appeler le service VMIE afin d'importer une machine virtuelle et de créer une AMI de base à partir de celle-ci. • Champ d'application renforcé pour ec2 : conditions CreateTags d'action et de balise de ressource.	20 novembre 2021
AWSServiceRoleForImageBuilder - mise à jour d'une politique existante	Image Builder a ajouté de nouvelles autorisations pour résoudre les problèmes liés au blocage de la création de l'image par plusieurs associations d'inventaire.	11 août 2021

Modification	Description	Date
AWSImageBuilderFullAccess - mise à jour d'une politique existante	Image Builder a apporté les modifications suivantes au rôle d'accès complet : • Autorisations ajoutées à autoriser `ec2:DescribeInstanceTypeOfferings` . • Des autorisations d'appel ont été ajoutées `ec2:DescribeInstanceTypeOffereings` pour permettre à la console Image Builder de refléter avec précision les types d'instances disponibles dans le compte.	13 avril 2021
Image Builder a commencé à suivre les modifications	Image Builder a commencé à suivre les modifications apportées AWS à ses politiques gérées.	2 avril 2021

Utilisation de rôles liés à un service pour EC2 Image Builder

EC2 Image Builder AWS Identity and Access Management utilise des rôles liés à un service (IAM). Un rôle lié à un service est un type unique de rôle IAM directement lié à Image Builder. Les rôles liés au service sont prédéfinis par Image Builder et incluent toutes les autorisations dont le service a besoin pour appeler d'autres personnes en votre Services AWS nom.

Un rôle lié à un service rend la configuration d'Image Builder plus efficace, car il n'est pas nécessaire d'ajouter les autorisations nécessaires manuellement. Image Builder définit les autorisations associées à ses rôles liés aux services et, sauf indication contraire, seul Image Builder peut

assumer ses rôles. Les autorisations définies comprennent la stratégie d'approbation et la stratégie d'autorisations. La stratégie d'autorisations ne peut pas être attachée à une autre entité IAM.

Pour plus d'informations sur les autres services qui prennent en charge les rôles liés à un service, consultez Services AWS That Work with IAM et recherchez les services dont la valeur est Oui dans la colonne Rôle lié au service. Choisissez un Oui ayant un lien permettant de consulter les détails du rôle pour ce service.

Autorisations de rôle liées à un service pour Image Builder

Image Builder utilise le rôle AWSServiceRoleForImageBuilderlié au service pour permettre à EC2 Image Builder d'accéder aux AWS ressources en votre nom. Le rôle lié au service fait confiance au service imagebuilder.amazonaws.com pour assumer le rôle.

Vous n'avez pas besoin de créer manuellement ce rôle lié à un service. Lorsque vous créez votre première image Image Builder dans la console AWS de gestion AWS CLI, ou dans l' AWS API, Image Builder crée pour vous le rôle lié au service.

Les actions suivantes créent une nouvelle image :

- Exécutez l'assistant de pipeline dans la console Image Builder pour créer une image personnalisée.
- Utilisez l'une des actions d'API suivantes ou la AWS CLI commande correspondante :
  - L'action de l'CreateImageAPI (create-imagedans le AWS CLI).
  - L'action de l'ImportVmImageAPI (import-vm-imagedans le AWS CLI).
  - L'action de l'StartImagePipelineExecutionAPI (start-image-pipeline-executiondans le AWS CLI).

> **Important**
>
> Si le rôle lié au service est supprimé de votre compte, vous pouvez suivre le même processus pour le créer à nouveau. Lorsque vous créez votre première ressource EC2 Image Builder, Image Builder crée à nouveau le rôle lié au service pour vous.

Pour voir les autorisations pour le AWSServiceRoleForImageBuilder, consultez la Stratégie AWSServiceRoleForImageBuilder page. Pour en savoir plus sur la configuration des autorisations pour un rôle lié à un service, consultez la section Autorisations relatives aux rôles liés à un service dans le guide de l'utilisateur IAM.

Supprimer un rôle lié au service Image Builder de votre compte

Vous pouvez utiliser la console IAM, le AWS CLI, ou l' AWS API pour supprimer manuellement le rôle lié à un service pour Image Builder de votre compte. Toutefois, avant de procéder, vous devez vous assurer qu'aucune ressource Image Builder n'y fait référence.

> **Note**
>
> Si le service Image Builder utilise le rôle lorsque vous essayez de supprimer les ressources, la suppression risque d'échouer. Si cela se produit, patientez quelques minutes et réessayez.

Nettoyez les ressources Image Builder utilisées par le **`AWSServiceRoleForImageBuilder`** rôle

1. Vérifiez qu'aucune version de pipeline n'est en cours d'exécution avant de commencer. Pour annuler une compilation en cours, utilisez la `cancel-image-creation` commande du AWS CLI.

   ```
   aws imagebuilder cancel-image-creation --image-build-version-
   arn arn:aws:imagebuilder:us-east-1:123456789012:image-pipeline/sample-pipeline
   ```

2. Modifiez tous les plannings de pipeline pour utiliser un processus de génération manuel, ou supprimez-les si vous ne les réutilisez pas. Pour plus d'informations sur la suppression de ressources, consultezSupprimer les ressources EC2 Image Builder.

Supprimer le rôle lié au service à l'aide d'IAM

Vous pouvez utiliser la console IAM, le AWS CLI, ou l' AWS API pour supprimer le `AWSServiceRoleForImageBuilder` rôle de votre compte. Pour plus d'informations, consultez Suppression d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Régions prises en charge pour les rôles liés au service EC2 Image Builder

Image Builder prend en charge l'utilisation de rôles liés à un service dans toutes les AWS régions où le service est disponible. Pour la liste des AWS régions prises en charge, consultezAWS Régions et points de terminaison.

Résolution des problèmes d'identité et d'accès à EC2 Image Builder

Rubriques

- Je ne suis pas autorisé à effectuer une action dans Image Builder
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je souhaite autoriser des personnes extérieures à moi Compte AWS à accéder à mes ressources Image Builder

Je ne suis pas autorisé à effectuer une action dans Image Builder

Si vous recevez une erreur qui indique que vous n'êtes pas autorisé à effectuer une action, vos politiques doivent être mises à jour afin de vous permettre d'effectuer l'action.

L'exemple d'erreur suivant se produit quand l'utilisateur IAM `mateojackson` tente d'utiliser la console pour afficher des informations détaillées sur une ressource *`my-example-widget`* fictive, mais ne dispose pas des autorisations `imagebuilder:`*`GetWidget`* fictives.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 imagebuilder:GetWidget on resource: my-example-widget
```

Dans ce cas, la politique qui s'applique à l'utilisateur `mateojackson` doit être mise à jour pour autoriser l'accès à la ressource *`my-example-widget`* à l'aide de l'action `imagebuilder:`*`GetWidget`*.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez un message d'erreur indiquant que vous n'êtes pas autorisé à effectuer l'`iam:PassRole`action, vos politiques doivent être mises à jour pour vous permettre de transmettre un rôle à Image Builder.

Certains vous Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, un utilisateur doit disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour effectuer une action dans Image Builder. Toutefois, l'action nécessite que le service ait des autorisations accordées par un rôle de service. Mary ne dispose pas des autorisations nécessaires pour transférer le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dans ce cas, les stratégies de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je souhaite autoriser des personnes extérieures à moi Compte AWS à accéder à mes ressources Image Builder

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACL), vous pouvez utiliser ces politiques pour donner l'accès à vos ressources.

Pour en savoir plus, consultez les éléments suivants :

- Pour savoir si Image Builder prend en charge ces fonctionnalités, consultezFonctionnement d'EC2 Image Builder avec IAM.
- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez dans le Guide de l'utilisateur IAM.
- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section Fournir un accès à des ressources Comptes AWS détenues par des tiers dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez Fournir un accès à des utilisateurs authentifiés en externe (fédération d'identité) dans le Guide de l'utilisateur IAM.
- Pour découvrir quelle est la différence entre l'utilisation des rôles et l'utilisation des politiques basées sur les ressources pour l'accès intercompte, consultez Différence entre les rôles IAM et les politiques basées sur les ressources dans le Guide de l'utilisateur IAM.

Validation de conformité pour EC2 Image Builder

EC2 Image Builder n'est concerné par AWS aucun programme de conformité.

Pour une liste des programmes Services AWS de conformité spécifiques concernés, voir AWS Services concernés par programme Services AWS de conformité dans Champ . Pour des informations générales, voir Programmes de AWS conformité Programmes AWS de .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir Téléchargement de rapports dans AWS Artifact .

Lorsque vous utilisez Image Builder, votre responsabilité en matière de conformité dépend de la sensibilité de vos données, des objectifs de conformité de votre entreprise et des lois et réglementations applicables. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- Guides démarrage rapide de la sécurité et de la conformité. Ces guides de déploiement traitent des considérations architecturales et fournissent des étapes pour déployer des environnements de base axés sur la sécurité et la conformité sur AWS.
- AWS Ressources de https://aws.amazon.com/compliance/resources/ de conformité — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- Évaluation des ressources à l'aide des règles du guide du AWS Config développeur : le AWS Config service évalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- AWS Security Hub— Ce AWS service fournit une vue complète de l'état de votre sécurité interne, AWS ce qui vous permet de vérifier votre conformité aux normes et aux meilleures pratiques du secteur de la sécurité.

Vous pouvez intégrer des produits conformes AWS Marketplace ou des composants de AWS Task Orchestrator and Executor (AWSTOE) dans vos images Image Builder pour garantir la conformité de vos images. Pour plus d'informations, consultez Produits de conformité pour vos images Image Builder.

Résilience dans EC2 Image Builder

L'infrastructure AWS mondiale est construite autour des AWS régions et des zones de disponibilité. AWS Les régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées,

connectées par un réseau à faible latence, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone à l'autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

Le service EC2 Image Builder vous permet de distribuer des images créées dans une région avec d'autres régions, ce qui leur confère une résilience multirégionale pour les AMI. Il n'existe aucun mécanisme permettant de « sauvegarder » les pipelines d'images, les recettes ou les composants. Vous pouvez stocker les documents relatifs à la recette et aux composants en dehors du service Image Builder, par exemple dans un compartiment Amazon S3.

L'EC2 Image Builder ne peut pas être configuré pour la haute disponibilité (HA). Vous pouvez distribuer des images dans plusieurs régions afin de les rendre plus disponibles.

Pour plus d'informations sur AWS les régions et les zones de disponibilité, consultez la section Infrastructure AWS mondiale.

Sécurité de l'infrastructure dans Image Builder

Le réseau AWS mondial fournit des fonctionnalités de sécurité et contrôle l'accès au réseau pour des services tels que EC2 Image Builder. Pour plus d'informations sur la sécurité de l'infrastructure qui AWS assure ses services, consultez la section Sécurité de l'infrastructure dans le livre blanc Introduction à la AWS sécurité.

Pour envoyer des demandes via le réseau AWS mondial pour les actions de l'API Image Builder, votre logiciel client doit respecter les consignes de sécurité suivantes :

- Pour envoyer des demandes d'actions d'API Image Builder, le logiciel client doit utiliser une version compatible de Transport Layer Security (TLS).

> **Note**
>
> AWS supprime progressivement le support pour les versions 1.0 et 1.1 du protocole TLS. Nous vous recommandons vivement de mettre à jour votre logiciel client pour utiliser la version TLS 1.2 ou ultérieure afin de pouvoir toujours vous connecter. Pour plus d'informations, consultez ce billet AWS de blog sur la sécurité.

- Les logiciels clients doivent prendre en charge les suites de chiffrement dotées d'un secret de transmission parfait (PFS), telles que Ephemeral Diffie-Hellman (DHE) ou Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La plupart des systèmes actuels, tels que Java 7 et versions ultérieures, prennent en charge ces modes.
- Vous devez signer vos demandes d'API avec un identifiant de clé d'accès et une clé d'accès secrète associés à un principal AWS Identity and Access Management (IAM). Vous pouvez également utiliser le AWS Security Token Service(AWS STS) pour générer des informations de sécurité temporaires pour vos demandes.

En outre, les instances EC2 qu'Image Builder utilise pour créer et tester des images doivent y avoir accès AWS Systems Manager.

Gestion des correctifs dans EC2 Image Builder

EC2 Image Builder fournit les dernières AMI Amazon Linux 2, Amazon Linux 2023, Red Hat Enterprise Linux (RHEL), CentOS, Ubuntu, SUSE Linux Enterprise Server et Windows 2012 R2 et versions ultérieures en tant que sources d'images gérées. Vous conservez la responsabilité de l'application des correctifs au système Amazon EC2, conformément au modèle de responsabilité partagée. Si les instances EC2 de la charge de travail de votre application peuvent être facilement remplacées, il peut être plus efficace de mettre à jour l'AMI de base et de redéployer tous les nœuds de calcul en fonction de cette image.

Les deux méthodes suivantes vous permettent de maintenir vos AMI Image Builder à jour.

- AWS-composants de correctif fournis — EC2 Image Builder fournit deux composants de compilation, qui installent toutes les mises à `update-linux` jour `update-windows` du système d'exploitation en attente. Ces composants utilisent le module `Update0S` d'action. Pour plus d'informations, consultez Mettre à jour le système d'exploitation. Les composants peuvent être ajoutés à vos pipelines de création d'images en les sélectionnant dans la liste des composants AWS fournis.
- Composants de génération personnalisés avec opérations de correction : pour installer ou mettre à jour de manière sélective des correctifs sur les systèmes d'exploitation des AMI prises en charge, vous pouvez créer un composant Image Builder pour installer les correctifs requis. Un composant personnalisé peut installer des correctifs à l'aide de scripts shell (Bash ou PowerShell), ou il peut utiliser le module `Update0S` d'action pour spécifier les correctifs à installer ou à exclure. Pour plus d'informations, consultez Modules d'action pris en charge par le gestionnaire de AWSTOE composants.

Composant utilisant le module `UpdateOS` d'action (Linux et Windows)

```
schemaVersion: 1.0
phases:
  - name: build
 steps:
   - name: UpdateOS
  action: UpdateOS
```

Composant qui utilise Bash pour installer les mises à jour yum

```
schemaVersion: 1.0
phases:
  - name: build
 steps:
   - name: InstallYumUpdates
  action: ExecuteBash
  inputs:
    commands:
   - sudo yum update -y
```

Bonnes pratiques de sécurité pour EC2 Image Builder

EC2 Image Builder fournit un certain nombre de fonctionnalités de sécurité à prendre en compte lors de l'élaboration et de la mise en œuvre de vos propres politiques de sécurité. Les bonnes pratiques suivantes doivent être considérées comme des instructions générales et ne représentent pas une solution de sécurité complète. Étant donné que ces bonnes pratiques peuvent ne pas être appropriées ou suffisantes pour votre environnement, considérez-les comme des remarques utiles plutôt que comme des recommandations.

- N'utilisez pas de groupes de sécurité trop permissifs dans les recettes d'Image Builder.
- Ne partagez pas d'images avec des comptes auxquels vous ne faites pas confiance.
- Ne publiez pas d'images contenant des données privées ou sensibles.
- Appliquez tous les correctifs de sécurité Windows ou Linux disponibles lors de la création d'images.

Nous vous recommandons vivement de tester vos images pour valider le niveau de sécurité et les niveaux de conformité en matière de sécurité applicables. Des solutions telles qu'Amazon Inspector peuvent aider à valider le niveau de sécurité et de conformité des images.

IMDSv2 pour les pipelines Image Builder

Lorsque votre pipeline Image Builder s'exécute, il envoie des requêtes HTTP pour lancer des instances EC2 qu'Image Builder utilise pour créer et tester votre image. Pour configurer la version d'IMDS que votre pipeline utilise pour les demandes de lancement, définissez le `httpTokens` paramètre dans les paramètres de métadonnées de votre instance de configuration d'infrastructure Image Builder.

> **Note**
>
> Nous vous recommandons de configurer toutes les instances EC2 lancées par Image Builder à partir d'un pipeline de manière à utiliser IMDSv2 afin que les demandes de récupération de métadonnées d'instance nécessitent un en-tête de jeton signé.

Pour plus d'informations sur la configuration de l'infrastructure Image Builder, consultezGérer la configuration de l'infrastructure EC2 Image Builder. Pour plus d'informations sur les options de métadonnées d'instance EC2 pour les images Linux, consultez Configurer les options de métadonnées d'instance dans le guide de l'utilisateur Amazon EC2 pour les instances Linux. Pour les images Windows, consultez Configurer les options de métadonnées d'instance dans le guide de l'utilisateur Amazon EC2 pour les instances Windows.

Nettoyage requis après la construction

Une fois qu'Image Builder a terminé toutes les étapes de création de votre image personnalisée, Image Builder prépare l'instance de génération à des fins de test et de création d'image. Avant d'arrêter l'instance de génération pour créer l'instantané, Image Builder effectue le nettoyage suivant pour garantir la sécurité de votre image :

Linux

Le pipeline Image Builder exécute un script de nettoyage pour garantir que l'image finale respecte les meilleures pratiques de sécurité et pour supprimer tous les artefacts ou paramètres de construction qui ne devraient pas être répercutés sur votre instantané. Cependant, vous pouvez ignorer certaines sections du script ou remplacer complètement les données utilisateur. Par

conséquent, les images produites par les pipelines Image Builder ne sont pas nécessairement conformes à des critères réglementaires spécifiques.

Lorsque le pipeline a terminé ses étapes de création et de test, Image Builder exécute automatiquement le script de nettoyage suivant juste avant de créer l'image de sortie.

> **Important**
>
> Si vous remplacez les données utilisateur dans votre recette, le script ne s'exécute pas. Dans ce cas, assurez-vous d'inclure une commande dans vos données utilisateur qui crée un fichier vide nommé`perform_cleanup`. Image Builder détecte ce fichier et exécute le script de nettoyage avant de créer la nouvelle image.

```
#!/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

Une fois que le pipeline Image Builder a personnalisé les images Windows, il exécute l'utilitaire Microsoft Sysprep. Ces actions suivent les AWS meilleures pratiques en matière de renforcement et de nettoyage de l'image.

Remplacer le script de nettoyage Linux

Image Builder crée des images sécurisées par défaut et suit nos meilleures pratiques en matière de sécurité. Toutefois, certains cas d'utilisation plus avancés peuvent vous obliger à ignorer une ou plusieurs sections du script de nettoyage intégré. Si vous devez ignorer une partie du nettoyage, nous vous recommandons vivement de tester votre AMI de sortie pour garantir la sécurité de votre image.

Important

Le fait de sauter des sections dans le script de nettoyage peut entraîner l'inclusion d'informations sensibles, telles que les détails du compte du propriétaire ou les clés SSH dans l'image finale, et dans toute instance lancée à partir de cette image. Vous pouvez également rencontrer des problèmes lors du lancement dans différentes zones de disponibilité, régions ou comptes.

Le tableau suivant décrit les sections du script de nettoyage, les fichiers supprimés dans cette section et les noms de fichiers que vous pouvez utiliser pour signaler une section qu'Image Builder doit ignorer. Pour ignorer une section spécifique du script de nettoyage, vous pouvez utiliser le module d'action du CreateFile composant ou une commande dans vos données utilisateur (en cas de remplacement) pour créer un fichier vide portant le nom indiqué dans la colonne Nom de fichier de la section Ignorer.

Note

Les fichiers que vous créez pour ignorer une section du script de nettoyage ne doivent pas comporter d'extension de fichier. Par exemple, si vous souhaitez

ignorer la `CLOUD_INIT_FILES` section du script, mais que vous créez un fichier nommé`skip_cleanup_cloudinit_files.txt`, Image Builder ne reconnaîtra pas le fichier ignoré.

Entrée

Section de nettoyage	Fichiers supprimés	Ignorer le nom du fichier de section

`CLOUD_INIT_FILES`	`/etc/sudoers.d/90-cloud-init-users`
	`/etc/locale.conf`
	`/var/log/cloud-init.log`
	`/var/log/cloud-init-output.log`
		`skip_cleanup_cloudinit_files`

`INSTANCE_FILES`	`/etc/.updated`
	`/etc/aliases.db`
	`/etc/hostname`
	`/var/lib/misc/postfix.aliasesdb-stamp`
	`/var/lib/postfix/master.lock`
	`/var/spool/postfix/pid/master.pid`
	`/var/.updated`
		`skip_cleanup_instance_files`

Section de nettoyage	Fichiers supprimés	Ignorer le nom du fichier de section
	/var/cache/yum/x86_64/2/.gpgkeyschecked.yum	
SSH_FILES	/etc/ssh/ssh_host_rsa_key /etc/ssh/ssh_host_rsa_key.pub /etc/ssh/ssh_host_ecdsa_key /etc/ssh/ssh_host_ecdsa_key.pub /etc/ssh/ssh_host_ed25519_key /etc/ssh/ssh_host_ed25519_key.pub /root/.ssh/authorized_keys /home/<all users>/.ssh/authorized_keys;	skip_cleanup_ssh_files
INSTANCE_LOG_FILES	/var/log/audit/audit.log /var/log/boot.log /var/log/dmesg /var/log/cron	skip_cleanup_instance_log_files

Section de nettoyage	Fichiers supprimés	Ignorer le nom du fichier de section
TOE_FILES	{{workingDirectory}}/TOE_*	skip_cleanup_toe_files
SSM_LOG_FILES	/var/log/amazon/ssm/*	skip_cleanup_ssm_log_files

Résoudre les problèmes liés à EC2 Image Builder

EC2 Image Builder s'intègre à la surveillance et Services AWS au dépannage afin de vous aider à résoudre les problèmes liés à la création d'images. Image Builder suit et affiche la progression de chaque étape du processus de création d'image. Image Builder peut également exporter les journaux vers un emplacement Amazon S3 que vous fournissez.

Pour un dépannage avancé, vous pouvez exécuter des commandes et des scripts prédéfinis à l'aide de AWS Systems Manager Run Command.

Table des matières

Résoudre les problèmes liés aux builds de pipelines

Si la génération d'un pipeline Image Builder échoue, Image Builder renvoie un message d'erreur décrivant l'échec. Image Builder renvoie également un `workflow execution ID` message d'échec, tel que celui de l'exemple de sortie suivant :

```
Workflow Execution ID: wf-12345abc-6789-0123-abc4-567890123abc failed with reason: …
```

Image Builder organise et dirige les actions de création d'images à travers une série d'étapes définies pour les étapes d'exécution de son processus de création d'image standard. Les étapes de création et de test du processus sont chacune associées à un flux de travail. Lorsqu'Image Builder exécute un flux de travail pour créer ou tester une nouvelle image, il génère une ressource de métadonnées de flux de travail qui assure le suivi des détails de l'exécution.

Les images de conteneur disposent d'un flux de travail supplémentaire qui s'exécute pendant la distribution.

Recherchez des informations détaillées sur les défaillances des instances d'exécution de votre flux de travail

Pour résoudre un problème d'échec d'exécution de votre flux de travail, vous pouvez appeler les actions GetWorkflowExecutionet ListWorkflowStepExecutionsAPI avec votre`workflow execution ID`.

Consulter les journaux d'exécution du flux de travail

- Amazon CloudWatch Logs

  Image Builder publie des journaux d'exécution détaillés du flux de travail dans le groupe et le flux de CloudWatch journaux Image Builder suivants :

  LogGroup:

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

  LogStream (x.x.x/x) :

  ```
  ImageVersion/ImageBuildVersion
  ```

  Avec CloudWatch Logs, vous pouvez rechercher les données des journaux à l'aide de modèles de filtre. Pour plus d'informations, consultez la section Rechercher dans les données des journaux à l'aide de modèles de filtre dans le guide de l'utilisateur Amazon CloudWatch Logs.

- AWS CloudTrail

  Toutes les activités de construction sont également enregistrées CloudTrail si elles sont activées dans votre compte. Vous pouvez filtrer CloudTrail les événements par source`imagebuilder.amazonaws.com`. Vous pouvez également rechercher l'ID d'instance Amazon EC2 renvoyé dans le journal d'exécution pour obtenir plus de détails sur l'exécution du pipeline.

- Amazon Simple Storage Service (S3)

  Si vous avez spécifié un nom de compartiment S3 et un préfixe de clé dans la configuration de votre infrastructure, le chemin du journal d'exécution des étapes du flux de travail suit le modèle suivant :

  ```
  S3://S3BucketName/KeyPrefix/ImageName/ImageVersion/ImageBuildVersion/
  WorkflowExecutionId/StepName
  ```

  Les journaux que vous envoyez à votre compartiment S3 indiquent les étapes et les messages d'erreur relatifs à l'activité sur l'instance EC2 pendant le processus de création de l'image. Les journaux incluent les sorties de journal du gestionnaire de composants, les définitions des composants exécutés et le résultat détaillé (au format JSON) de toutes les étapes effectuées sur

l'instance. Si vous rencontrez un problème, vous devez consulter ces fichiers, en commençant par `application.log`, pour diagnostiquer la cause du problème sur l'instance.

Par défaut, Image Builder arrête l'instance de build ou de test Amazon EC2 en cours d'exécution lorsque le pipeline échoue. Vous pouvez modifier les paramètres d'instance pour la ressource de configuration d'infrastructure utilisée par votre pipeline, afin de conserver votre instance de build ou de test à des fins de dépannage.

Pour modifier les paramètres de l'instance dans la console, vous devez décocher la case Terminer l'instance en cas de défaillance située dans la section Paramètres de résolution des problèmes de la ressource de configuration de votre infrastructure.

Vous pouvez également modifier les paramètres de l'instance à l'aide de la update-infrastructure-configuration commande figurant dans le AWS CLI. Définissez la `terminateInstanceOnFailure` valeur sur `false` dans le fichier JSON auquel la commande fait référence avec le `--cli-input-json` paramètre. Pour plus de détails, consultez Mettre à jour une configuration d'infrastructure.

Scénarios de résolution des problèmes

Cette section répertorie les scénarios de dépannage détaillés suivants :

- Accès refusé — code d'état 403
- Expect des délais de construction lors de la vérification de la disponibilité de l'agent Systems Manager sur l'instance de build
- Le disque secondaire Windows est hors ligne au lancement
- La compilation échoue avec l'image de base renforcée CIS
- AssertInventoryCollection échoue (Systems Manager Automation)

Pour voir les détails d'un scénario, choisissez le titre du scénario pour le développer. Vous pouvez étendre plusieurs titres en même temps.

Accès refusé — code d'état 403

Description

La construction du pipeline échoue avec le code d'état « AccessDenied : Accès refusé : 403 ».

Cause

Les causes possibles incluent :

- Le profil d'instance ne dispose pas des autorisations requises pour accéder aux API ou aux ressources des composants.
- Le rôle de profil d'instance ne dispose pas des autorisations requises pour se connecter à Amazon S3. Cela se produit le plus souvent lorsque le rôle de profil d'instance ne dispose pas PutObjectd'autorisations pour vos compartiments S3.

Solution

Selon la cause, ce problème peut être résolu comme suit :

- Le profil d'instance ne contient pas de politiques gérées : ajoutez les politiques manquantes à votre rôle de profil d'instance. Exécutez ensuite à nouveau le pipeline.
- Le profil d'instance ne dispose pas d'autorisations d'écriture pour le compartiment S3 : ajoutez une politique à votre rôle de profil d'instance qui accorde PutObjectles autorisations d'écriture dans votre compartiment S3. Exécutez ensuite à nouveau le pipeline.

Expect des délais de construction lors de la vérification de la disponibilité de l'agent Systems Manager sur l'instance de build

Description

La construction du pipeline échoue avec « status = 'TimedOut' » et « message d'échec = 'L'étape a expiré pendant que l'étape vérifie la disponibilité de l'agent Systems Manager sur la ou les instances cibles ».

Cause

Les causes possibles incluent :

- L'instance qui a été lancée pour effectuer les opérations de génération et exécuter les composants n'a pas pu accéder au point de terminaison Systems Manager.
- Le profil d'instance ne dispose pas des autorisations requises.

Solution

En fonction de la cause possible, ce problème peut être résolu comme suit :

- Problème d'accès, sous-réseau privé — Si vous créez un sous-réseau privé, assurez-vous d'avoir configuré des PrivateLink points de terminaison pour Systems Manager, Image Builder et, si vous souhaitez vous connecter, Amazon S3/. CloudWatch Pour plus d'informations sur la configuration des points de PrivateLink terminaison, consultez la section Concepts des points de terminaison VPC ().AWS PrivateLink
- Autorisations manquantes : ajoutez les politiques gérées suivantes à votre rôle lié au service IAM pour Image Builder :
  - EC2 InstanceProfileForImageBuilder
  - EC2 ECR InstanceProfileForImageBuilder ContainerBuilds
  - Amazon SMS ManagedInstanceCore

  Pour plus d'informations sur le rôle lié au service Image Builder, consultez. Utilisation de rôles liés à un service pour EC2 Image Builder

Le disque secondaire Windows est hors ligne au lancement

Description

Lorsque le type d'instance utilisé pour créer une AMI Windows Image Builder ne correspond pas au type d'instance utilisé pour le lancement à partir de l'AMI, un problème peut survenir lorsque les volumes non root sont hors ligne au lancement. Cela se produit principalement lorsque l'instance de construction utilise une architecture plus récente que l'instance de lancement.

L'exemple suivant montre ce qui se passe lorsqu'une AMI Image Builder est créée sur un type d'instance EC2 Nitro et lancée sur une instance EC2 Xen :

Type d'instance de build : m5.large (Nitro)

Type d'instance de lancement : 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
```

Cause

En raison des paramètres par défaut de Windows, les disques récemment découverts ne sont pas automatiquement mis en ligne ni formatés. Lorsque le type d'instance est modifié sur EC2, Windows le traite comme de nouveaux disques découverts. Cela est dû au changement de moteur sous-jacent.

Solution

Nous vous recommandons d'utiliser le même système de types d'instances lors de la création de l'AMI Windows à partir duquel vous souhaitez le lancer. N'incluez pas de types d'instances créés sur différents systèmes dans la configuration de votre infrastructure. Si l'un des types d'instances que vous spécifiez utilise le système Nitro, il doit tous utiliser le système Nitro.

Pour plus d'informations sur les instances créées sur le système Nitro, consultez Instances créées sur le système Nitro dans le guide de l'utilisateur Amazon EC2 pour les instances Windows.

La compilation échoue avec l'image de base renforcée CIS

Description

Vous utilisez une image de base renforcée CIS et la compilation échoue.

Cause

Lorsque le `/tmp` répertoire est classé comme`noexec`, cela peut entraîner l'échec d'Image Builder.

Solution

Choisissez un autre emplacement pour votre répertoire de travail dans le `workingDirectory` champ de la recette d'image. Pour plus d'informations, consultez la description du type de ImageRecipedonnées.

AssertInventoryCollection échoue (Systems Manager Automation)

Description

Systems Manager Automation montre un échec lors de l'étape `AssertInventoryCollection` d'automatisation.

Cause

Vous ou votre organisation avez peut-être créé une association Systems Manager State Manager qui collecte des informations d'inventaire pour les instances EC2. Si la collecte améliorée de métadonnées d'image est activée pour votre pipeline Image Builder (il s'agit de la valeur par défaut), Image Builder tente de créer une nouvelle association d'inventaire pour l'instance de génération. Toutefois, Systems Manager n'autorise pas plusieurs associations d'inventaire pour les instances gérées et empêche toute nouvelle association si elle existe déjà. Cela entraîne l'échec de l'opération et l'échec de la construction du pipeline.

Solution

Pour résoudre ce problème, désactivez la collecte améliorée des métadonnées d'image à l'aide de l'une des méthodes suivantes :

- Mettez à jour votre pipeline d'images dans la console, pour désactiver la case à cocher Activer la collecte améliorée de métadonnées. Enregistrez vos modifications et exécutez une génération de pipeline.

  Pour plus d'informations sur la mise à jour de votre pipeline d'images AMI à l'aide de la console EC2 Image Builder, Mettre à jour les pipelines d'images AMI (console) consultez. Pour plus d'informations sur la mise à jour de votre pipeline d'images de conteneur à l'aide de la console EC2 Image Builder, Mettre à jour un pipeline d'images de conteneur (console) consultez.
- Vous pouvez également mettre à jour votre pipeline d'images à l'aide de la update-image-pipeline commande figurant dans le AWS CLI. Pour ce faire, incluez la `EnhancedImageMetadataEnabled` propriété dans votre fichier JSON, définie sur`false`. L'exemple suivant montre la propriété définie sur`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"
}
```

Pour éviter que cela ne se produise pour les nouveaux pipelines, désactivez la case à cocher Activer la collecte améliorée de métadonnées lorsque vous créez un nouveau pipeline à l'aide de la console EC2 Image Builder, ou définissez la valeur de `EnhancedImageMetadataEnabled` la propriété dans votre fichier JSON sur lorsque vous créez votre pipeline `false` à l'aide AWS CLI du.

Historique du document pour le guide de l'utilisateur d'EC2 Image Builder

Le tableau suivant décrit les modifications importantes apportées à la documentation par date. Pour recevoir les notifications de mise à jour de cette documentation, abonnez-vous à un flux RSS.

- Version de l'API : 23/12/12

Modification	Description	Date
Mises à jour du STIG Q1	Versions de Linux STIG mises à jour et application de STIGS pour la version du premier trimestre 2024. Aucune modification n'a été apportée aux versions de Windows.	23 février 2024
Version de fonctionnalité : gestion du flux de production d'images	Grâce aux flux de productio n d'images, vous bénéficiez d'une flexibilité, d'une visibilit é et d'un contrôle accrus sur le processus de création d'images. Vous pouvez personnaliser les étapes de création et de test pour vos flux de travail, ou vous pouvez utiliser le flux de travail par défaut d'Image Builder.	12 décembre 2023
Mises à jour du STIG Q4	Versions de Linux STIG mises à jour et application de STIGS pour la version du quatrième trimestre 2023. Aucune modification n'a été apportée aux versions de Windows. Linux et Windows SCAP ont	7 décembre 2023

	également été mis à jour pour les nouveaux composant s, logiciels et numéros de référence.	
Version de fonctionnalité : gestion du cycle de vie des images	Grâce aux politiques et règles de gestion du cycle de vie des images, vous pouvez définir votre stratégie de gestion des ressources afin de garantir que les images obsolètes et les ressources associées passent par un processus de balisage et de suppression.	17 novembre 2023
Mises à jour du STIG Q3	Versions STIG mises à jour et STIGS appliquées pour la version du troisième trimestre 2023. Messagerie également mise à jour pour préciser que les packages tiers ne sont pas automatiquement installés, à de très rares exceptions près. Tous les STIG ignorés sont enregistrés.	5 octobre 2023
Nouvelles versions de STIG	Versions STIG mises à jour et STIGS appliquées pour la version du deuxième trimestre 2023.	3 mai 2023
Nouvelles versions de STIG	Versions STIG mises à jour et STIGS appliqués pour la version du premier trimestre 2023. Ajout du support pour AL2023.	14 avril 2023

Mettre à jour les régions prises en charge pour AWSTOE	Ajout de la prise en charge des éléments suivants Régions AWS : Asie-Pacifique (Hyderabad), Asie-Pacifique (Jakarta), Europe (Zurich), Europe (Espagne) et Moyen-Orient (Émirats arabes unis).	13 avril 2023
AWSTOE mises à jour de téléchargement des applications	Signature mise à jour pour le téléchargement de AWSTOE l'installation sous Windows. Le protocole TLS a également été mis à jour. Notez que les téléchargements d'applications à partir de compartiments S3 nécessitent désormais la version 1.2 ou ultérieure du protocole TLS.	31 mars 2023
Version de fonctionnalité : flux de travail de création améliorés	Ajout de détails d'exécution pour les compilations d'images dans le nouvel onglet de flux de travail dans les détails de la version de génération d'image. Informations améliorées pour le dépannage des versions.	30 mars 2023

Version de fonctionnalité : détection et génération de rapports CVE	Pour les comptes qui ont activé les scans Amazon Inspector, Image Builder peut capturer les découvertes courantes d'Amazon Inspector relatives aux vulnérabilités et aux expositions (CVE) lors de la phase de test du processus de création pour les nouvelles images, y compris les images de conteneur stockées dans Amazon ECR. Image Builder crée un instantané des résultats pour faciliter l'analyse détaillée. Image Builder indique également le nombre de résultats qui peuvent être filtrés par compte, par pipeline ou par image, avec la possibilité d'approfondir les détails.	30 mars 2023
Historique des versions ajouté	Ajout de l'historique des versions dans les sections Windows et Linux.	17 février 2023
Nouvelles versions de STIG	Versions STIG mises à jour et STIGS appliquées pour la version du quatrième trimestre 2022.	1er février 2023

Version des fonctionnalités : AWS Marketplace intégration et renforcement du CIS	AWS Marketplace Intégration ajoutée pour trouver et utiliser facilement une image abonnée comme base de référence pour une nouvelle image personnalisée, y compris des images renforcées CIS et un nouveau composant CIS Hardening du Center for Internet Security.	13 janvier 2023
composants de durcissement CIS	Ajout de composants de durcissement CIS détenus et entretenus par CIS.	13 janvier 2023
Nouvelles versions de STIG	Introduction de la prise en charge d'Ubuntu, mise à jour des versions de STIG et application de STIGS pour la version du deuxième trimestre 2022.	20 juillet 2022
Mise à jour du document : navigation pour la page de document de création d'un composant YAML	Le contenu du document du composant Create YAML a été déplacé vers sa propre page et les autres pages ont été mises à jour pour y faire référence.	7 juin 2022
Nouvelles versions de STIG	Versions STIG mises à jour et STIGS appliqués pour la version du premier trimestre 2022.	25 avril 2022

Module ExecuteDocument d'action ajouté	Ajout de la documentation pour le module `ExecuteDocument` d'action ci-dessous `General execution` .	28 mars 2022
Version de fonctionnalité : Support pour un lancement plus rapide de l'AMI Windows	Ajout de paramètres de configuration de distribution pour accélérer le lancement des AMI Windows.	21 février 2022
Version de maintenance : mise à jour de l'empreinte AWSTOE numérique binaire	Empreinte binaire mise à jour pour le certificat du AWSTOE signataire.	18 février 2022
Version de fonctionnalité : configurer la saisie pour AWSTOE	Ajout de la prise en charge de l'utilisation d'un fichier de configuration JSON comme entrée pour la AWSTOE run commande.	3 février 2022
Nouvelles versions de STIG	Versions STIG mises à jour et STIGS appliquées pour la version du quatrième trimestre 2021. Une section a également été ajoutée pour les nouveaux composants du SCAP Compliance Checker (SCC).	22 décembre 2021

Version de fonctionnalité : intégration de VM Import/Export (VMIE)	Ajout du support pour l'importation de machines virtuelles via tous les canaux (console, API/CLI, etc.) et pour l'exportation de machines virtuelles via API/CLI. L'exportation de machines virtuelles n'est actuellement pas disponible depuis la console Image Builder.	20 décembre 2021
Version de fonctionnalité : partage d'AMI pour AWS Organizations et UO	Configuration de distribution mise à jour pour ajouter la prise en charge du partage des AMI de sortie avec AWS Organizations et des unités d'organisation.	24 novembre 2021
Mise à jour du document : étapes et phases de mise à jour des composants	Contenu étendu sur les étapes des composants dans Image Builder et sur la manière dont celles-ci interagissent avec les phases des AWSTOE composants.	22 septembre 2021
Mise à jour du document : ajout de contenu CloudTrail d'intégration	Résumé de la surveillance et contenu CloudTrail d'intégration ajoutés.	17 septembre 2021
Nouvelles versions de STIG	Versions STIG mises à jour et STIGS appliquées pour la version du troisième trimestre 2021.	10 septembre 2021

Version de fonctionnalité : EventBridge intégration avec Amazon	Ajout d' EventBridge un support qui vous permet de connecter Image Builder à des événements liés Serv ices AWS et de lancer des événements en fonction des règles définies dans EventBrid ge.	18 août 2021
Mise à jour du document : réorganiser les pages AWSTOE	AWSTOE Pages réorganisées pour plus de clarté.	11 août 2021
Version des fonctionnalités : composants paramétrés et configuration d'instance supplémentaire	Ajout de la prise en charge de la spécification de paramètre s afin de personnaliser les composants des recettes. Configuration étendue des instances EC2 utilisées pour créer et tester des images, notamment la possibilité de spécifier les commandes à exécuter au lancement et un meilleur contrôle de l'install ation et de la suppression de l'agent Systems Manager.	7 juillet 2021
Nouvelles versions de STIG	Versions STIG mises à jour et STIGS appliquées pour la sortie du deuxième trimestre 2021.	30 Juin 2021
Amélioration : améliorations du balisage	Messagerie améliorée concernant le balisage des ressources.	25 juin 2021

Version de fonctionnalité : intégration du modèle de lancement	Ajout de la prise en charge de l'utilisation des modèles de lancement Amazon EC2 pour la distribution d'AMI dans les paramètres de distribution.	7 avril 2021
Version de fonctionnalité : améliorations apportées à la création de conteneurs	Ajout de la prise en charge de la configuration des mappages de périphériques par blocs et de la spécification des AMI à utiliser comme image de base pour les builds de conteneurs.	7 avril 2021
Nouvelles versions de STIG	Versions STIG mises à jour et STIGS appliqués.	5 mars 2021
Mettre à jour les expressions cron	Le traitement cron d'Image Builder est mis à jour pour augmenter la granularité des expressions cron à la minute près et utilise un moteur de planification cron standard. Les exemples sont mis à jour avec le nouveau format.	8 février 2021
Version de fonctionnalité : prise en charge des conteneurs	Ajout de la prise en charge de la création d'images de conteneurs Docker à l'aide d'Image Builder, avec enregistrement et stockage des images obtenues sur Amazon Elastic Container Registry (Amazon ECR). Le contenu a été réorganisé pour refléter les nouvelles fonctionnalités et tenir compte de la croissance future.	17 décembre 2020

Documentation cron restructurée	Cette page présente désormais plus d'informations sur le fonctionnement de cron avec les builds du pipeline Image Builder, ainsi que des informations sur l'heure UTC. Les caractères génériques qui ne sont pas autorisés pour des champs spécifiques ont été supprimés. Les exemples incluent désormais des exemples d'expressions pour la console et la CLI.	13 novembre 2020
Version console 2.0 : mise à jour de l'édition du pipeline	Le contenu change lors de la mise en route et des didacticiels de création de pipelines, ainsi que de la page de gestion des pipelines d'images, afin d'intégrer les nouvelles fonctionnalités et le nouveau flux de la console.	13 novembre 2020
Nouvelles versions de STIG	Versions STIG mises à jour et STIGS appliqués. Remarque : le format de liste a été modifié pour afficher les STIG appliqués par défaut.	15 octobre 2020
Support pour les constructions en boucle dans AWSTOE	Créez des structures en boucle pour définir une séquence répétée d'instructions dans l' AWSTOE application.	29 juillet 2020

Support au développement local de AWSTOE composants	Développez et testez des composants d'image localement avec l' AWSTOE application.	28 juillet 2020
AMI chiffrées	EC2 Image Builder ajoute la prise en charge de la distribution d'AMI chiffrées.	1er juillet 2020
AutoScaling dépréciation	Obsolète de l'utilisation de AutoScaling pour lancer des instances.	15 juin 2020
Support pour la connectivité via AWS PrivateLink	Vous pouvez établir une connexion privée entre votre VPC et EC2 Image Builder en créant un point de terminaison VPC d'interface. Les points de terminaison de l'interface sont alimentés par AWS PrivateLink une technologie qui vous permet d'accéder en privé aux API Image Builder sans passerelle Internet, appareil NAT, connexion VPN ou connexion AWS Direct Connect. Les instances de votre VPC n'ont pas besoin d'adresses IP publiques pour communiquer avec les API Image Builder. Le trafic entre votre VPC et Image Builder ne quitte pas le réseau Amazon.	10 juin 2020
Nouvelles versions de STIG	Versions STIG mises à jour et STIGS appliqués.	23 janvier 2020

Dépannage	Scénarios de dépannage généraux ajoutés.	22 janvier 2020
Composants STIG	Vous pouvez créer des images conformes aux normes STIG à l'aide de composants AWSTOE STIG.	22 janvier 2020

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.