aws

Guida per l'utente

EC2 Image Builder

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

EC2 Image Builder: Guida per l'utente

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

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

Che cos'è Image Builder? 1
Caratteristiche di Image Builder 2
Sistemi operativi supportati 3
Formati di immagine supportati 4
Quote di default 4
AWS Regioni ed endpoint 4
Concetti 4
Prezzi 8
Correlati Servizi AWS 9
Come funziona Image Builder 9
Elementi AMI 10
Gestione dei componenti 11
Risorse create 11
Distribuzione 12
Condivisione delle risorse 13
Conformità 13
Versione semantica 13
Inizia la configurazione 15
Ruolo collegato al servizio Image Builder 15
Requisiti di configurazione 16
Archivio di contenitori per pipeline di immagini di contenitori 16
Host dedicato per immagini macOS 17
Prerequisiti IAM 17
Prerequisiti per Systems Manager Agent 18
Tutorial di Image Builder 20
Crea la tua prima immagine 20
Crea un componente personalizzato con parametri di input 20
Utilizzo dei parametri di Systems Manager con Image Builder 21
Procedura guidata per la pipeline: crea AMI 21
Fase 1: Specificare i dettagli della pipeline 21
Fase 2: Scegli la ricetta 22
Fase 3: Definizione della configurazione dell'infrastruttura (opzionale) 25
Fase 4: Definizione delle impostazioni di distribuzione (opzionale) 26
Fase 5. Esamina 26

Fase 6: pulizia ... 26
Pipeline Wizard: crea un'immagine del contenitore ... 29
Fase 1: Specificare i dettagli della pipeline ... 29
Fase 2: Scegli la ricetta ... 30
Fase 3: Definizione della configurazione dell'infrastruttura (opzionale) ... 34
Fase 4: Definizione delle impostazioni di distribuzione (opzionale) ... 34
Fase 5. Esamina ... 34
Fase 6: pulizia ... 35
Componente personalizzato con parametri ... 37
Usa i parametri nel documento del componente YAML ... 38
Imposta i parametri dei componenti in una ricetta di Image Builder dalla console ... 42
Usa un parametro di immagine di base nella tua ricetta ... 43
Fase 1: Trova o crea un parametro Parameter Store ... 43
Passaggio 2: configurare le autorizzazioni IAM ... 44
Passaggio 3: Creare una ricetta di immagini che utilizzi il parametro ... 45
Componenti ... 47
Elenca e visualizza i componenti ... 49
Elenca i componenti di Image Builder ... 50
Elenca le versioni di build dei componenti da AWS CLI ... 52
Ottenete i dettagli dei componenti dal AWS CLI ... 53
Ottieni dettagli sulla politica dei componenti da AWS CLI ... 53
Usa Marketplace AWS i componenti ... 53
Scopri i componenti Marketplace AWS ... 54
Abbonarsi ai Marketplace AWS componenti ... 55
Usa Marketplace AWS i componenti in una ricetta ... 56
Usa componenti gestiti ... 57
Pacchetto di distribuzione: installazione dell'applicazione Windows ... 57
Indurimento CIS ... 62
Componenti di indurimento STIG ... 63
Sviluppa componenti personalizzati ... 103
Creare un documento componente YAML ... 104
Crea componenti personalizzati con Image Builder ... 108
AWSTOE applicazione di gestione dei componenti ... 118
AWSTOE scarica ... 118
Regioni supportate ... 120
Riferimento al comando ... 122

Configurazione manuale 128
Usa il framework dei documenti dei componenti 141
Moduli d'azione 195
Configura l'input 307
Risorse di immagine 312
Elenca le immagini e crea le versioni 312
Elenca immagini 313
Elenca le immagini in attesa di azione 319
Elenca le versioni di build dell'immagine 321
Visualizza i dettagli delle risorse di immagini 324
Visualizza i dettagli dell'immagine nella console Image Builder 324
Per informazioni dettagliate sulla politica relativa alle immagini, consulta il AWS CLI 332
Crea immagini personalizzate con Image Builder 332
Annulla la creazione dell'immagine da AWS CLI 335
Importa ed esporta immagini VM 335
Importazione di una macchina virtuale in Image Builder 336
Distribuisci i dischi VM dalla tua immagine creata da AWS CLI 341
Importa immagini disco ISO 341
Sistemi operativi supportati per l'importazione di immagini disco ISO 342
Prerequisiti 342
Importazione di un'immagine disco ISO in Image Builder 343
Gestisci i risultati di sicurezza 347
Configurare le scansioni di sicurezza 348
Gestisci i risultati di sicurezza 350
Pulisci le risorse di Image Builder 352
Gestisci il ciclo di vita delle immagini 353
Prerequisiti 354
Crea un ruolo IAM per la gestione del ciclo di vita di Image Builder 355
Crea un ruolo IAM per la gestione del ciclo di vita tra account di Image Builder 356
Elenca le politiche relative al ciclo di vita 358
Visualizzare le policy del ciclo di vita 361
Visualizza i dettagli delle policy relative al ciclo di vita nella console Image Builder 361
Crea politiche per il ciclo di vita 364
Creazione di politiche del ciclo di vita (immagine AMI) 364
Crea politiche del ciclo di vita (immagine del contenitore) 367
Come funzionano le regole del ciclo di vita 369

Regole di esclusione del ciclo di vita delle AMI ... 371
Visualizza i dettagli delle regole di gestione del ciclo di vita ... 372
Configura immagini personalizzate ... 373
Ricette ... 373
Elenca e visualizza le ricette illustrate ... 374
Elenca e visualizza le ricette del contenitore ... 376
Crea una nuova versione di una ricetta di immagini ... 378
Crea una nuova versione di una ricetta contenitore ... 392
Pulizia delle risorse ... 401
Configurazioni dell'infrastruttura ... 402
Elenca e visualizza le configurazioni dell'infrastruttura ... 403
Creazione di una configurazione dell'infrastruttura ... 404
Aggiornare una configurazione dell'infrastruttura ... 409
AWS PrivateLink Endpoint VPC ... 411
Distribution Settings (Impostazioni distribuzione) ... 417
Elenca e visualizza le configurazioni di distribuzione ... 419
Crea e aggiorna la distribuzione AMI ... 420
Creare e aggiornare la distribuzione delle immagini dei contenitori ... 434
Configura la distribuzione AMI tra account ... 437
Specificare un modello di lancio AMI ... 445
Condividi risorse ... 449
Proprietari delle risorse ... 450
Prerequisiti per la condivisione delle risorse di Image Builder ... 450
Consumatori di risorse ... 451
Creare una condivisione di risorse ... 451
Opzione 1: creare una condivisione di risorse RAM ... 452
Opzione 2: applicare una politica delle risorse e passare a una condivisione di risorse esistente ... 452
Annulla la condivisione di una risorsa ... 455
Aggiunta di tag alle risorse ... 456
Assegna un tag a una risorsa dal AWS CLI ... 456
Togliere il tag a una risorsa da AWS CLI ... 457
Elenca tutti i tag per una risorsa specifica dal AWS CLI ... 457
Delete resources (Elimina risorse) ... 459
Console: elimina le risorse ... 459
Eliminare risorse ()AWS CLI ... 461

Flussi di lavoro relativi alle immagini 463
Framework del flusso di lavoro: fasi 464
Accesso al servizio 464
Utilizza flussi di lavoro gestiti 465
Elenca i flussi di lavoro con immagini 465
Crea un flusso di lavoro basato su immagini 468
Creare un documento di workflow YAML 472
Struttura di un documento di workflow YAML 472
Azioni graduali 483
Variabili dinamiche 501
Istruzioni condizionali 505
Gestisci le pipeline 511
Elenca e visualizza le pipeline 512
Elenca le pipeline di immagini da AWS CLI 512
Ottieni i dettagli della pipeline di immagini da AWS CLI 512
Creare e aggiornare pipeline (AMI) 512
Crea pipeline AMI da AWS CLI 513
Aggiorna la pipeline dalla console 515
Aggiorna la pipeline dal AWS CLI 519
Crea e aggiorna pipeline (contenitore) 520
Crea una pipeline da AWS CLI 521
Aggiorna la pipeline dalla console 523
Aggiorna la pipeline da AWS CLI 527
Configura i flussi di lavoro della pipeline 529
Definisci gruppi di test per i flussi di lavoro di test 529
Impostare i parametri del flusso di lavoro in una pipeline di Image Builder dalla console 530
Specificare il ruolo del servizio IAM utilizzato da Image Builder per eseguire le azioni del flusso di lavoro 531
Esegui pipeline 531
Usa le espressioni cron 532
Valori supportati per le espressioni cron in Image Builder 533
Esempi di espressioni cron in Image Builder 535
Espressioni della frequenza 537
Usa le regole EventBridge 538
EventBridge termini 538
Visualizza EventBridge le regole per la tua pipeline di Image Builder 540

Usa EventBridge le regole per pianificare la costruzione di una pipeline 540
Integra prodotti e servizi 542
Amazon EventBridge 545
Messaggi di eventi inviati da Image Builder 546
Amazon Inspector 548
Marketplace AWS 549
Marketplace AWS abbonamenti in Image Builder 550
Scopri i prodotti di Marketplace AWS immagini dalla console Image Builder 551
Usa un prodotto di Marketplace AWS immagini nelle ricette di Image Builder 553
Amazon Simple Notification Service 554
Argomenti SNS crittografati 554
Formato del messaggio SNS 556
Prodotti per la conformità 562
Monitora eventi e registri 564
CloudTrail registri 564
Informazioni su Image Builder in CloudTrail 564
CloudWatch Registri 565
Sicurezza in Image Builder 567
Protezione dei dati 568
Crittografia e gestione delle chiavi 569
Archiviazione di dati 575
Riservatezza del traffico Internet 575
Identity and Access Management 575
Destinatari 575
Autenticazione con identità 576
Come Image Builder funziona con le politiche e i ruoli IAM 576
Gestisci i perimetri dei dati 588
Policy basate sulle identità 589
Policy basate su risorse 592
Policy gestite 593
Ruoli collegati ai servizi 608
Risoluzione dei problemi 610
Convalida della conformità 612
Resilienza 613
Sicurezza dell'infrastruttura 614
Gestione delle patch 615

Best practice 616
Pulizia post-compilazione richiesta 617
Sostituisci lo script di pulizia di Linux 628
Risoluzione dei problemi relativi a Image Builder 632
Risolvi i problemi relativi alle build della pipeline 632
Scenari per la risoluzione dei problemi 634
Cronologia dei documenti 640
..... dclv

Che cos'è Image Builder?

EC2 Image Builder è un programma completamente gestito Servizio AWS che consente di automatizzare la creazione, la gestione e l'implementazione di immagini server personalizzate, sicure e up-to-date. Puoi usare AWS Management Console AWS Command Line Interface, o APIs per creare immagini personalizzate nel tuo. Account AWS

Sei il proprietario delle immagini personalizzate che Image Builder crea nel tuo account. Puoi configurare le pipeline per automatizzare gli aggiornamenti e l'applicazione di patch di sistema per le immagini che possiedi. Puoi anche eseguire un comando autonomo per creare un'immagine con le risorse di configurazione che hai definito.

La procedura guidata della pipeline di Image Builder può guidarti attraverso i passaggi per creare un'immagine personalizzata, come segue:

Fase 1: Specificare i dettagli della pipeline

- Assegna un nome alla pipeline e aggiungi tag.
- Definisci le impostazioni di scansione dei metadati e delle vulnerabilità.
- Imposta una pianificazione per la tua pipeline.

Fase 2: Personalizza la tua immagine

Puoi selezionare una ricetta esistente o crearne una nuova.

- Scegli un'immagine di base per le tue personalizzazioni.
- Aggiungi o rimuovi software dall'immagine di base.
- Personalizza le impostazioni e gli script con i componenti di compilazione.
- Seleziona i componenti di test da eseguire.

Fase 3: Definisci il tuo flusso di lavoro

Un flusso di lavoro di immagini definisce la sequenza di passaggi che Image Builder esegue durante le fasi di creazione e test del processo di creazione dell'immagine. Questo fa parte del framework generale del flusso di lavoro di Image Builder.

Fase 4: Configurazione dell'infrastruttura di costruzione

- Seleziona un ruolo IAM da associare al profilo dell'istanza per le istanze che Image Builder avvia durante il processo di creazione dell'immagine.
- Seleziona uno o più tipi di istanza che possono essere applicati all'avvio.
- Seleziona un argomento di Amazon Simple Notification Service (SNS) per ricevere notifiche da Image Builder.
- Specificare VPC, sottorete e gruppi di sicurezza da utilizzare per il processo di creazione dell'immagine.
- Seleziona le impostazioni per la risoluzione dei problemi, ad esempio dove Image Builder scrive i log e se terminare l'istanza di build in caso di errore (impostazione predefinita) o mantenerla in esecuzione per un'ulteriore risoluzione dei problemi.

Fase 5: Definire la distribuzione delle immagini

- Seleziona Regioni AWS dove Image Builder distribuisce la tua Amazon Machine Image (AMI) o l'immagine del contenitore.
- Se la pipeline di Image Builder crea un'AMI, Image Builder supporta anche la seguente configurazione:
  - Seleziona una chiave KMS da utilizzare per la crittografia.
  - Configura la condivisione AMI tra Account AWS e Organizzazioni.
  - Associa una licenza autogestita di License Manager all'immagine distribuita.
  - Configura un modello di lancio per la tua immagine.

Caratteristiche di Image Builder

EC2 Image Builder offre le seguenti funzionalità:

Aumentate la produttività e riducete le operazioni per la conformità e le immagini degli edifici up-to-date

Image Builder riduce la quantità di lavoro necessaria per creare e gestire immagini su larga scala automatizzando le pipeline di creazione. È possibile automatizzare le build specificando le preferenze relative alla pianificazione di esecuzione delle build. L'automazione riduce i costi operativi di manutenzione del software con le patch più recenti del sistema operativo.

Aumentate l'operatività del servizio

Image Builder fornisce l'accesso ai componenti di test che è possibile utilizzare per testare le immagini prima della distribuzione. È inoltre possibile creare componenti di test personalizzati con AWS Task Orchestrator and Executor (AWSTOE) e utilizzarli. Image Builder distribuisce l'immagine solo se tutti i test configurati hanno avuto esito positivo.

Alza il livello di sicurezza per le implementazioni

Image Builder consente di creare immagini che eliminano l'esposizione non necessaria alle vulnerabilità di sicurezza dei componenti. È possibile applicare le impostazioni AWS di sicurezza per creare out-of-the-box immagini sicure che soddisfino i criteri di sicurezza interni e di settore. Image Builder fornisce anche raccolte di impostazioni per le aziende dei settori regolamentati. È possibile utilizzare queste impostazioni per creare immagini conformi agli standard STIG in modo rapido e semplice. Per un elenco completo dei componenti STIG disponibili tramite Image Builder, vedere. Amazon ha gestito i componenti di protezione STIG per Image Builder

Applicazione centralizzata e tracciamento del lignaggio

Utilizzando le integrazioni integrate con AWS Organizations, Image Builder consente di applicare politiche che limitano gli account all'esecuzione di istanze solo se approvate. AMIs

Condivisione semplificata delle risorse tra Account AWS

EC2 Image Builder si integra con AWS Resource Access Manager (AWS RAM) per consentire di condividere determinate risorse con chiunque o Account AWS tramite. AWS Organizations EC2 Le risorse di Image Builder che possono essere condivise sono:

- Componenti
- Immagini
- Ricette immagine
- Ricette container

Per ulteriori informazioni, consulta Condividi le risorse di Image Builder con AWS RAM.

Sistemi operativi supportati

Image Builder supporta le seguenti versioni del sistema operativo:

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

Formati di immagine supportati

Per le tue immagini personalizzate che creano un'Amazon Machine Image (AMI), puoi scegliere un AMI esistente come punto di partenza. Per le immagini dei container Docker, puoi scegliere tra immagini pubbliche ospitate su DockerHub, immagini di container esistenti in Amazon ECR o immagini di container gestite da Amazon come punto di partenza.

Quote di default

Per visualizzare le quote predefinite per Image Builder, vedere Image Builder Endpoints and Quotas.

AWS Regioni ed endpoint

Per visualizzare gli endpoint del servizio per Image Builder, vedere Image Builder Endpoints and Quotas.

Concetti

I seguenti termini e concetti sono fondamentali per la comprensione e l'utilizzo di EC2 Image Builder.

AMI

Amazon Machine Image (AMI) è l'unità di distribuzione di base in Amazon EC2 ed è uno dei tipi di immagini che puoi creare con Image Builder. Un'AMI è un'immagine di macchina virtuale preconfigurata che contiene il sistema operativo (OS) e il software preinstallato per distribuire EC2 le istanze. Per ulteriori informazioni, consulta Amazon Machine Images (AMI).

Pipeline di immagini

Una pipeline di immagini fornisce un framework di automazione su cui creare immagini sicure AMIs e containerizzate. AWS La pipeline di immagini di Image Builder è associata a una ricetta di immagini o una ricetta contenitore che definisce le fasi di creazione, convalida e test per il ciclo di vita di creazione di un'immagine.

Una pipeline di immagini può essere associata a una configurazione dell'infrastruttura che definisce la posizione in cui viene creata l'immagine. Puoi definire attributi, ad esempio il tipo di istanza, le sottoreti, i gruppi di sicurezza, la registrazione e altre configurazioni correlate all'infrastruttura. È inoltre possibile associare la pipeline di immagini a una configurazione di distribuzione per definire la modalità di distribuzione dell'immagine.

Immagine gestita

Un'immagine gestita è una risorsa in Image Builder che consiste in un'AMI o un'immagine contenitore, oltre a metadati, come versione e piattaforma. L'immagine gestita viene utilizzata dalle pipeline di Image Builder per determinare quale immagine di base utilizzare per la build. In questa guida, le immagini gestite vengono talvolta chiamate «immagini», tuttavia un'immagine non è la stessa di un'AMI.

Ricetta dell'immagine

Una ricetta di immagini Image Builder è un documento che definisce l'immagine di base e i componenti che vengono applicati all'immagine di base per produrre la configurazione desiderata per l'immagine AMI di output. È possibile utilizzare una ricetta immagine per duplicare le build. Le ricette di immagini di Image Builder possono essere condivise, ramificate e modificate utilizzando la procedura guidata della console, o l'API AWS CLI. È possibile utilizzare ricette di immagini con il software di controllo delle versioni per mantenere ricette di immagini condivisibili e con versioni diverse.

Ricetta in contenitore

Una ricetta del contenitore Image Builder è un documento che definisce l'immagine di base e i componenti che vengono applicati all'immagine di base per produrre la configurazione desiderata per l'immagine del contenitore di output. È possibile utilizzare una ricetta contenitore per duplicare le build. È possibile condividere, ramificare e modificare le ricette di immagini di Image Builder utilizzando la procedura guidata della console AWS CLI, o l'API. È possibile utilizzare ricette container con il software di controllo delle versioni per mantenere ricette container condivisibili e con versioni diverse.

Immagine di base

L'immagine di base è l'immagine e il sistema operativo selezionati utilizzati nell'immagine o nel documento di ricetta del contenitore, insieme ai componenti. L'immagine di base e le definizioni dei componenti combinate producono la configurazione desiderata per l'immagine di output.

Componenti

Un componente definisce la sequenza di passaggi necessari per personalizzare un'istanza prima della creazione dell'immagine (un componente di compilazione) o per testare un'istanza lanciata dall'immagine creata (un componente di test).

Un componente viene creato da un documento YAML o JSON dichiarativo in testo semplice che descrive la configurazione di runtime per la creazione e la convalida o il test di un'istanza prodotta dalla pipeline. I componenti vengono eseguiti sull'istanza utilizzando un'applicazione di gestione dei componenti. L'applicazione di gestione dei componenti analizza i documenti ed esegue i passaggi desiderati.

Dopo la creazione, uno o più componenti vengono raggruppati utilizzando una ricetta di immagine o una ricetta di contenitore per definire il piano per la creazione e il test di una macchina virtuale o di un'immagine del contenitore. È possibile utilizzare componenti pubblici di proprietà e gestiti da AWS oppure crearne di propri. Per ulteriori informazioni sui componenti, vedereIn che modo Image Builder utilizza l' AWS Task Orchestrator and Executor applicazione per gestire i componenti.

Documento del componente

Un documento YAML o JSON dichiarativo in testo semplice che descrive la configurazione per una personalizzazione che è possibile applicare all'immagine. Il documento viene utilizzato per creare un componente di compilazione o test.

Fasi di runtime

EC2 Image Builder prevede due fasi di esecuzione: build e test. Ogni fase di runtime ha una o più fasi con configurazione definita dal documento del componente.

Fasi di configurazione

L'elenco seguente mostra le fasi che vengono eseguite durante le fasi di compilazione e test:

Fase di costruzione:

Fase di compilazione

Una pipeline di immagini inizia con la fase di compilazione della fase di compilazione quando viene eseguita. L'immagine di base viene scaricata e la configurazione specificata per la fase di compilazione del componente viene applicata per creare e avviare un'istanza.

Fase di convalida

Dopo che Image Builder avvia l'istanza e applica tutte le personalizzazioni della fase di compilazione, inizia la fase di convalida. Durante questa fase, Image Builder assicura che tutte le personalizzazioni funzionino come previsto, in base alla configurazione specificata dal componente per la fase di convalida. Se la convalida dell'istanza ha esito positivo, Image Builder interrompe l'istanza, crea un'immagine e quindi passa alla fase di test.

Fase di test:

Fase di test

Durante questa fase, Image Builder avvia un'istanza dall'immagine creata dopo che la fase di convalida è stata completata con successo. Image Builder esegue i componenti di test durante questa fase per verificare che l'istanza sia integra e funzioni come previsto.

Fase di test dell'host del contenitore

Dopo che Image Builder ha eseguito la fase di test per tutti i componenti selezionati nella ricetta del contenitore, Image Builder esegue questa fase per i flussi di lavoro dei container. La fase di test dell'host del contenitore può eseguire test aggiuntivi che convalidano la gestione dei contenitori e le configurazioni di runtime personalizzate.

Flusso di lavoro

I flussi di lavoro definiscono la sequenza di passaggi che Image Builder esegue quando crea una nuova immagine. Tutte le immagini hanno flussi di lavoro di compilazione e test. I contenitori dispongono di un flusso di lavoro aggiuntivo per la distribuzione.

Tipi di flussi di lavoro

`BUILD`

Copre la configurazione della fase di costruzione per ogni immagine creata.

`TEST`

Copre la configurazione della fase di test per ogni immagine creata.

`DISTRIBUTION`

Copre il flusso di lavoro di distribuzione delle immagini dei container.

Prezzi

L'utilizzo di EC2 Image Builder per creare AMI o immagini di container personalizzate è gratuito. Tuttavia, per gli altri servizi utilizzati nel processo si applicano i prezzi standard. L'elenco seguente include l'utilizzo di alcune immagini Servizi AWS che possono comportare costi durante la creazione, la creazione, l'archiviazione e la distribuzione di AMI o immagini di container personalizzate, a seconda della configurazione.

- Avvio di un'istanza EC2
- Archiviazione dei log su Amazon S3
- Convalida delle immagini con Amazon Inspector
- Archiviazione di istantanee Amazon EBS per AMIs
- Archiviazione delle immagini dei container in Amazon ECR
- Inserimento e estrazione di immagini di container da e verso Amazon ECR
- Se Systems Manager Advanced Tier è attivato e EC2 le istanze Amazon vengono eseguite con l'attivazione locale, le risorse potrebbero essere addebitate tramite Systems Manager

Correlati Servizi AWS

EC2 Image Builder ne utilizza altri Servizi AWS per creare immagini, a seconda della configurazione della ricetta di Image Builder. Per ulteriori informazioni sull'integrazione di prodotti e servizi per le immagini personalizzate, consulta. Integra prodotti e servizi in Image Builder

Come funziona EC2 Image Builder

Quando si utilizza la console EC2 Image Builder per creare una pipeline di immagini personalizzata, il sistema guida l'utente nei seguenti passaggi.

1. Specificate i dettagli della pipeline: inserite le informazioni sulla pipeline, come nome, descrizione, tag e una pianificazione per eseguire build automatizzate. Puoi scegliere build manuali, se preferisci.
2. Scegli la ricetta: scegli tra la creazione di un'AMI o la creazione di un'immagine contenitore. Per entrambi i tipi di immagini di output, inserisci un nome e una versione per la ricetta, seleziona un'immagine di base e scegli i componenti da aggiungere per la creazione e il test. Puoi anche scegliere il controllo automatico delle versioni, per assicurarti di utilizzare sempre l'ultima versione del sistema operativo (OS) disponibile per l'immagine di base. Le ricette dei container definiscono inoltre i Dockerfile e il repository Amazon ECR di destinazione per l'immagine del contenitore Docker di output.

   > **Note**
   >
   > I componenti sono gli elementi costitutivi utilizzati da una ricetta di immagini o da una ricetta contenitore. Ad esempio, i pacchetti per l'installazione, le fasi di rafforzamento della sicurezza e i test. L'immagine di base e i componenti selezionati costituiscono una ricetta di immagini.

3. Definisci la configurazione dell'infrastruttura: Image Builder avvia EC2 istanze nel tuo account per personalizzare le immagini ed eseguire test di convalida. Le impostazioni di configurazione dell'infrastruttura specificano i dettagli dell'infrastruttura per le istanze che verranno eseguite durante il Account AWS processo di creazione.
4. Definisci le impostazioni di distribuzione: scegli le AWS regioni in cui distribuire l'immagine dopo che la build è stata completata e ha superato tutti i test. La pipeline distribuisce automaticamente l'immagine nella regione in cui esegue la build e puoi aggiungere la distribuzione delle immagini per altre regioni.

Le immagini che crei a partire dall'immagine di base personalizzata si trovano nella tua. Account AWS Puoi configurare la tua pipeline di immagini per produrre versioni aggiornate e patchate dell'immagine inserendo una pianificazione di creazione. Una volta completata la build, puoi ricevere notifiche tramite Amazon Simple Notification Service (SNS). Oltre a produrre un'immagine finale, la procedura guidata della console Image Builder genera una ricetta che può essere utilizzata con i sistemi di controllo delle versioni esistenti e le pipeline di integration/continuous distribuzione continua (CI/CD) per l'automazione ripetibile. Puoi condividere e creare nuove versioni della tua ricetta.

Contenuto della sezione

- Elementi AMI
- Gestione dei componenti
- Risorse create
- Distribuzione
- Condivisione delle risorse
- Conformità

Elementi AMI

Un'Amazon Machine Image (AMI) è un'immagine di macchina virtuale (VM) preconfigurata che contiene il sistema operativo e il software per distribuire EC2 le istanze.

Un AMI include i seguenti elementi:

- Un modello per il volume principale della macchina virtuale. Quando avvii una macchina EC2 virtuale Amazon, il volume del dispositivo root contiene l'immagine per avviare l'istanza. Quando viene utilizzato l'instance store, il dispositivo root è un volume di instance store creato da un modello in Amazon S3. Per ulteriori informazioni, consulta Amazon EC2 Root Device Volume.
- Quando si utilizza Amazon EBS, il dispositivo root è un volume EBS creato da uno snapshot EBS.
- Autorizzazioni di avvio che determinano il Account AWS tipo di avvio VMs con l'AMI.
- Blocca i dati di mappatura dei dispositivi che specificano i volumi da collegare all'istanza dopo il lancio.
- Un identificatore di risorsa univoco per ogni regione, per ogni account.
- Payload di metadati come tag e proprietà, come regione, sistema operativo, architettura, tipo di dispositivo root, provider, autorizzazioni di avvio, archiviazione per il dispositivo root e stato di firma.

- Una firma AMI per le immagini Windows per la protezione da manomissioni non autorizzate. Per ulteriori informazioni, consulta Instance Identity Documents.

Gestione dei componenti

EC2 Image Builder utilizza un'applicazione per la gestione dei componenti AWS Task Orchestrator and Executor (AWSTOE) che consente di orchestrare flussi di lavoro complessi, modificare le configurazioni di sistema e testare i sistemi con componenti di script basati su YAML. Poiché AWSTOE è un'applicazione autonoma, non richiede alcuna configurazione aggiuntiva. Può essere eseguito su qualsiasi infrastruttura cloud e in locale. Per iniziare a utilizzarla AWSTOE come applicazione autonoma, consultaConfigurazione manuale per sviluppare componenti personalizzati con AWSTOE.

Image Builder lo utilizza AWSTOE per eseguire tutte le attività su istanza. Queste includono la creazione e la convalida dell'immagine prima di scattare un'istantanea e il test dell'istantanea per assicurarsi che funzioni come previsto prima di creare l'immagine finale. Per ulteriori informazioni su come Image Builder utilizza AWSTOE per gestire i suoi componenti, vedere. Usa i componenti per personalizzare l'immagine di Image Builder Per ulteriori informazioni sulla creazione di componenti con AWSTOE, vedereIn che modo Image Builder utilizza l' AWS Task Orchestrator and Executor applicazione per gestire i componenti.

Test delle immagini

È possibile utilizzare i componenti di AWSTOE test per convalidare l'immagine e assicurarsi che funzioni come previsto, prima di creare l'immagine finale.

In genere, ogni componente di test è costituito da un documento YAML che contiene uno script di test, un binario di test e metadati di test. Lo script di test contiene i comandi di orchestrazione per avviare il binario di test, che può essere scritto in qualsiasi linguaggio supportato dal sistema operativo. I codici di stato di uscita indicano l'esito del test. I metadati del test descrivono il test e il suo comportamento, ad esempio il nome, la descrizione, i percorsi del test binario e la durata prevista.

Risorse create

Quando si crea una pipeline, non viene creata alcuna risorsa esterna a Image Builder, a meno che non sia vero quanto segue:

- Quando un'immagine viene creata tramite la pianificazione della pipeline

- Quando si sceglie Esegui Pipeline dal menu Azioni nella console Image Builder
- Quando esegui uno di questi comandi dall'API o: oppure AWS CLIStartImagePipelineExecution CreateImage

Le seguenti risorse vengono create durante il processo di creazione dell'immagine:

Pipeline di immagini AMI

- EC2 istanza (temporanea)
- Systems Manager Inventory Association (tramite Systems Manager State Manager se `EnhancedImageMetadata` abilitata) sull' EC2 istanza
- EC2 AMI Amazon
- Lo snapshot di Amazon EBS associato ad Amazon AMI EC2

Pipeline di immagini dei contenitori

- Contenitore Docker in esecuzione su un' EC2 istanza (temporaneo)
- Systems Manager Inventory Association (tramite Systems Manager State Manager) `EnhancedImageMetadata` è abilitata sull' EC2 istanza
- Immagine di container Docker
- Dockerfile

Dopo la creazione dell'immagine, tutte le risorse temporanee vengono eliminate.

Distribuzione

EC2 Image Builder può distribuire AMIs o contenere immagini in qualsiasi AWS regione. L'immagine viene copiata in ogni regione specificata nell'account utilizzato per creare l'immagine.

Per le immagini di output AMI, è possibile definire le autorizzazioni di avvio AMI per controllare quali Account AWS sono autorizzate ad avviare EC2 istanze con l'AMI creata. Ad esempio, è possibile rendere l'immagine privata, pubblica o condividerla con account specifici. Se si distribuisce l'AMI ad altre regioni e si definiscono le autorizzazioni di avvio per altri account, le autorizzazioni di avvio vengono propagate AMIs in tutte le regioni in cui è distribuito l'AMI.

Puoi anche utilizzare il tuo AWS Organizations account per imporre limitazioni agli account dei membri e avviare istanze solo con istanze approvate e conformi. AMIs Per ulteriori informazioni, consulta Gestire i dati nella propria organizzazione Account AWS.

Per aggiornare le impostazioni di distribuzione utilizzando la console Image Builder, segui i passaggi perCrea una nuova versione di image recipe dalla console, o. Crea una nuova versione di ricetta in contenitore con la console

Condivisione delle risorse

Per condividere componenti, ricette o immagini con altri account o all'interno AWS Organizations, consultaCondividi le risorse di Image Builder con AWS RAM.

Conformità

Per i benchmark Center for Internet Security (CIS), Image EC2 Builder utilizza Amazon Inspector per eseguire valutazioni di esposizione, vulnerabilità e deviazioni dalle migliori pratiche e dagli standard di conformità. Ad esempio, Image Builder valuta l'accessibilità involontaria della rete, la connettività Internet pubblica senza patch e l' CVEsattivazione dell'accesso root remoto. Amazon Inspector è offerto come componente di test che puoi scegliere di aggiungere alla ricetta dell'immagine. Per ulteriori informazioni su Amazon Inspector, consulta la Amazon Inspector User Guide. Per ulteriori informazioni, consulta Center for Internet Security (CIS) Benchmarks.

Image Builder fornisce componenti di protezione STIG per aiutarvi a creare in modo più efficiente immagini conformi agli standard STIG di base. Questi componenti STIG analizzano eventuali configurazioni errate ed eseguono uno script di correzione. Non sono previsti costi aggiuntivi per l'utilizzo di componenti conformi a STIG. Per un elenco completo dei componenti STIG disponibili tramite Image Builder, vedere. Amazon ha gestito i componenti di protezione STIG per Image Builder

Versionamento semantico in Image Builder

Image Builder utilizza il controllo delle versioni semantiche per organizzare le risorse e garantire che siano uniche. IDs La versione semantica ha quattro nodi:

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

È possibile assegnare valori per i primi tre e filtrarli tutti.

Il controllo delle versioni semantiche è incluso nell'Amazon Resource Name (ARN) di ogni oggetto, al livello che si applica a quell'oggetto come segue:

1. Versionless ARNs e Name ARNs non includono valori specifici in nessuno dei nodi. I nodi vengono omessi completamente oppure vengono specificati come caratteri jolly, ad esempio: x.x.x.
2. <major>La versione ARNs ha solo i primi tre nodi:. <minor>. <patch>
3. La versione build ARNs ha tutti e quattro i nodi e punta a una build specifica per una versione specifica di un oggetto.

Assegnazione: per i primi tre nodi è possibile assegnare qualsiasi valore intero positivo, o zero, con un limite superiore di 2^30-1 o 1073741823 per ogni nodo. Image Builder assegna automaticamente il numero di build al quarto nodo.

Modelli: è possibile utilizzare qualsiasi modello numerico che rispetti i requisiti di assegnazione per i nodi che è possibile assegnare. Ad esempio, è possibile scegliere un modello di versione del software, ad esempio 1.0.0, o una data, ad esempio 2021.01.01.

Selezione: con il controllo semantico delle versioni, hai la flessibilità di utilizzare i caratteri jolly (x) per specificare le versioni o i nodi più recenti quando selezioni l'immagine o i componenti di base per la tua ricetta. Quando utilizzi un carattere jolly in un nodo, anche tutti i nodi a destra del primo carattere jolly devono essere caratteri jolly.

Ad esempio, date le seguenti versioni recenti: 2.2.4, 1.7.8 e 1.6.8, la selezione della versione mediante caratteri jolly produce i seguenti risultati:

- `x.x.x` = 2.2.4
- `1.x.x` = 1,7,8
- `1.6.x` = 1,6,8
- `x.2.x` non è valido e genera un errore
- `1.x.8` non è valido e genera un errore

Preparati per creare immagini personalizzate con Image Builder

Prima di creare immagini con EC2 Image Builder, verifica di aver soddisfatto i seguenti prerequisiti per creare una pipeline di immagini. Salvo diversa indicazione, questi prerequisiti sono necessari per tutti i tipi di pipeline.

Prerequisiti

- Ruolo collegato al servizio Image Builder
- Requisiti di configurazione
- Archivio di contenitori per pipeline di immagini di contenitori
- Host dedicato per immagini macOS
- Prerequisiti IAM
- Prerequisiti per Systems Manager Agent

Dopo aver soddisfatto i prerequisiti, è possibile gestire EC2 Image Builder da una delle seguenti interfacce.

- EC2 Console Image Builder
- Comandi Image Builder in AWS CLI
- EC2 Riferimento all'API Image Builder
- AWS SDKs e strumenti

Ruolo collegato al servizio Image Builder

EC2 Image Builder utilizza un ruolo collegato al servizio per concedere autorizzazioni ad altri AWS servizi per conto dell'utente. Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando crei la tua prima risorsa Image Builder nella AWS Management Console, o nell' AWS API AWS CLI, Image Builder crea automaticamente il ruolo collegato al servizio. Per ulteriori informazioni sul ruolo collegato ai servizi che Image Builder crea nel tuo account, consulta. Usa i ruoli collegati ai servizi IAM per Image Builder

Requisiti di configurazione

- Image Builder supporta. AWS PrivateLink Per ulteriori informazioni sulla configurazione degli endpoint VPC per Image Builder, vedere. Image Builder e endpoint AWS PrivateLink VPC di interfaccia
- Le istanze utilizzate da Image Builder per creare immagini di container devono avere accesso a Internet per AWS CLI scaricarle da Amazon S3 e per scaricare un'immagine di base dal repository Docker Hub, se applicabile. Image Builder utilizza il AWS CLI per ottenere il Dockerfile dalla ricetta del contenitore, dove viene archiviato come dati.
- Le istanze utilizzate da Image Builder per creare immagini ed eseguire test devono avere accesso al servizio Systems Manager. I requisiti di installazione dipendono dal sistema operativo in uso.

  Per visualizzare i requisiti di installazione per l'immagine di base, scegli la scheda corrispondente al sistema operativo dell'immagine di base.

  Linux

  Per le istanze Amazon EC2 Linux, Image Builder installa Systems Manager Agent sull'istanza di build se non è già presente e lo rimuove prima di creare l'immagine.

  Windows

  Image Builder non installa Systems Manager Agent sulle istanze di build di Amazon EC2 Windows Server. Se l'immagine di base non è stata preinstallata con Systems Manager Agent, è necessario avviare un'istanza dall'immagine di origine, installare manualmente Systems Manager sull'istanza e creare una nuova immagine di base dall'istanza.

  Per installare manualmente l'agente Systems Manager sulla tua istanza di Amazon EC2 Windows Server, consulta Installazione manuale di Systems Manager Agent sulle EC2 istanze per Windows Server nella Guida per l'AWS Systems Manager utente.

  macOS

  TBD: hai bisogno di qualcosa sui prerequisiti

Archivio di contenitori per pipeline di immagini di contenitori

Per le pipeline di immagini dei contenitori, la ricetta definisce la configurazione per le immagini Docker prodotte e archiviate nel repository del contenitore di destinazione. È necessario creare il repository di destinazione prima di creare la ricetta del contenitore per l'immagine Docker.

Image Builder utilizza Amazon ECR come repository di destinazione per le immagini dei container. Per creare un repository Amazon ECR, segui i passaggi descritti in Creazione di un repository nella Amazon Elastic Container Registry User Guide.

Host dedicato per immagini macOS

Le istanze Amazon EC2 Mac richiedono un host dedicato in esecuzione su un tipo di istanza metal. Prima di creare un'immagine macOS personalizzata, devi allocare un host dedicato al tuo account. Per ulteriori informazioni sulle istanze Mac e un elenco di tipi di istanze che supportano nativamente il sistema operativo macOS, consulta le istanze Amazon EC2 Mac nella Amazon User Guide. EC2

Dopo aver creato un host dedicato, puoi configurare le impostazioni nella risorsa di configurazione dell'infrastruttura per la tua immagine. La configurazione dell'infrastruttura include proprietà di posizionamento in cui è possibile specificare l'host, il gruppo di posizionamento degli host o la zona di disponibilità a cui devono andare le istanze avviate dall'immagine.

Prerequisiti IAM

Il ruolo IAM che associ al tuo profilo di istanza deve disporre delle autorizzazioni per eseguire i componenti di compilazione e test inclusi nell'immagine. Le seguenti policy di ruolo IAM devono essere collegate al ruolo IAM associato al profilo dell'istanza:

- EC2InstanceProfileForImageBuilder
- EC2InstanceProfileForImageBuilderECRContainerBuilds
- Amazon SSMManaged InstanceCore

Se configuri la registrazione, il profilo dell'istanza specificato nella configurazione dell'infrastruttura deve disporre delle `s3:PutObject` autorizzazioni per il bucket di destinazione (). `arn:aws:s3:::BucketName/*` Per esempio:

```
{
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "s3:PutObject"
```

```
        ],
        "Resource": "arn:aws:s3:::bucket-name/*"
    }
]
}
```

Collegamento della policy

I seguenti passaggi guidano l'utente attraverso il processo di associazione delle policy IAM a un ruolo IAM per concedere le autorizzazioni precedenti.

1. Accedi alla console di AWS gestione e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel riquadro di navigazione sinistro, scegli Policy.
3. Filtra l'elenco delle politiche con EC2InstanceProfileForImageBuilder
4. Seleziona il bullet accanto alla politica e, dall'elenco a discesa Azioni relative alle politiche, seleziona Allega.
5. Seleziona il nome del ruolo IAM a cui allegare la policy.
6. Scegli Collega policy.
7. Ripeti i passaggi 3-6 per le SSMManaged InstanceCore politiche EC2InstanceProfileForImageBuilderECRContainerBuildse Amazon.

> **Note**
>
> Se si desidera copiare un'immagine creata con Image Builder su un altro account, è necessario creare il `EC2ImageBuilderDistributionCrossAccountRole` ruolo in tutti gli account di destinazione e allegare la politica Policy Ec2ImageBuilderCrossAccountDistributionAccess gestita al ruolo. Per ulteriori informazioni, consulta Condividi le risorse di Image Builder con AWS RAM.

Prerequisiti per Systems Manager Agent

EC2 Image Builder esegue AWS Systems Manager (Systems Manager) Agent sulle EC2 istanze che avvia per creare e testare l'immagine. Image Builder raccoglie informazioni aggiuntive sull'istanza utilizzata durante la fase di creazione con Systems Manager Inventory. Queste informazioni

includono il nome e la versione del sistema operativo (OS), nonché l'elenco dei pacchetti e delle rispettive versioni riportate dal sistema operativo.

Per disattivare la raccolta di queste informazioni, seleziona il metodo che corrisponde al tuo ambiente preferito:

- Console Image Builder: deselezionate la casella di controllo Abilita raccolta avanzata di metadati.
- AWS CLI— Specificare l'opzione `--no-enhanced-image-metadata-enabled`
- API Image Builder o SDKs: imposta il `enhancedImageMetadataEnabled` parametro su. `false`

Image Builder lo utilizza `RunCommand` per inviare azioni all'istanza di compilazione e test come parte del flusso di lavoro di creazione e test delle immagini. Non puoi disattivare l'invio di azioni `RunCommand` all'istanza di build e test.

Scopri come creare immagini personalizzate con i tutorial di Image Builder

Esistono molti modi per creare immagini e componenti personalizzati con EC2 Image Builder. I tutorial ti aiutano a conoscere i concetti chiave di Image Builder. Ogni tutorial presenta un caso d'uso con passaggi che è possibile seguire per la prima volta. Le istruzioni utilizzano i valori predefiniti, ove possibile, per facilitare l'apprendimento dell'intero processo. Dopo aver utilizzato uno dei tutorial, puoi esplorare altri modi per personalizzare le tue immagini.

Crea la tua prima immagine

I seguenti tutorial mostrano come creare la prima immagine con la procedura guidata della console Image Builder. Alla fine del tutorial avrai creato il seguente set di risorse per Image Builder. L'ultimo passaggio del tutorial consiste nel ripulire le risorse che hai creato.

- Una ricetta di immagine per la tua Amazon Machine Image (AMI) o una ricetta di contenitore per la tua immagine del contenitore.
- Una risorsa di configurazione dell'infrastruttura con impostazioni predefinite.
- Una risorsa per le impostazioni di distribuzione con impostazioni predefinite che distribuisce l'output nella regione di origine (l'account nella regione utilizzata per eseguire la procedura guidata della console).
- Una pipeline di immagini che utilizza le risorse elencate per creare l'immagine di output con i flussi di lavoro di creazione di immagini predefiniti.
- Un AMI di output o un'immagine del contenitore.

Tutorial sulle procedure guidate per la console

- Procedura guidata per la pipeline: crea AMI
- Pipeline Wizard: crea un'immagine del contenitore

Crea un componente personalizzato con parametri di input

Il seguente tutorial mostra come creare un componente personalizzato che definisce i parametri di input e quindi impostare i valori dalla ricetta di Image Builder.

Componente personalizzato con parametri

Utilizzo dei parametri di Systems Manager con Image Builder

Il seguente tutorial mostra come creare un AWS Systems Manager parametro Parameter Store e utilizzarlo in una ricetta di immagini.

Usa un parametro di immagine di base nella tua ricetta

Puoi anche utilizzare i parametri Parameter Store nelle impostazioni di distribuzione AMI per memorizzare l'ID dell'immagine di output e nei componenti personalizzati. Per ulteriori informazioni, consulta Crea e aggiorna le configurazioni di distribuzione AMI per le distribuzioni e Usa i parametri del Parameter Store di Systems Manager per i componenti personalizzati.

Tutorial: crea una pipeline di immagini con l'AMI di output dalla procedura guidata della console Image Builder

Questo tutorial illustra come creare una pipeline automatizzata per creare e gestire un' EC2 immagine Image Builder personalizzata utilizzando la procedura guidata della console Create image pipeline. Per aiutarti a eseguire i passaggi in modo efficiente, vengono utilizzate le impostazioni predefinite quando sono disponibili e le sezioni facoltative vengono ignorate.

Crea un flusso di lavoro basato sulla pipeline di immagini

- Fase 1: Specificare i dettagli della pipeline
- Fase 2: Scegli la ricetta
- Fase 3: Definizione della configurazione dell'infrastruttura (opzionale)
- Fase 4: Definizione delle impostazioni di distribuzione (opzionale)
- Fase 5. Esamina
- Fase 6: pulizia

Fase 1: Specificare i dettagli della pipeline

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Per iniziare a creare la pipeline, scegli Crea pipeline di immagini.
3. Nella sezione Generale, inserisci il nome della tua pipeline (obbligatorio).

> **Tip**
>
> La raccolta avanzata di metadati è attivata per impostazione predefinita. Per garantire la compatibilità tra i componenti e le immagini di base, tienila accesa.

4. Nella sezione Pianificazione della creazione, puoi mantenere le impostazioni predefinite per le opzioni di pianificazione. Tieni presente che il fuso orario mostrato per la pianificazione predefinita è Universal Coordinated Time (UTC). Per ulteriori informazioni sull'ora UTC e per trovare lo scostamento del fuso orario, vedere Abbreviazioni dei fusi orari - Elenco mondiale.

   Per le impostazioni di aggiornamento delle dipendenze, scegli l'opzione Esegui la pipeline all'ora pianificata se sono disponibili aggiornamenti delle dipendenze. Questa impostazione fa sì che la pipeline verifichi la presenza di aggiornamenti prima di iniziare la compilazione. Se non ci sono aggiornamenti, salta la compilazione pianificata della pipeline.

   > **Note**
   >
   > Per garantire che la pipeline riconosca gli aggiornamenti delle dipendenze e crei le build come previsto, è necessario utilizzare il controllo delle versioni semantiche (x.x.x) per l'immagine di base e i componenti. Per ulteriori informazioni sul controllo delle versioni semantiche per le risorse di Image Builder, vedere. Versionamento semantico in Image Builder

5. Scegli Avanti per procedere al passaggio successivo.

Fase 2: Scegli la ricetta

1. L'impostazione predefinita di Image Builder è Usa la ricetta esistente nella sezione Ricetta. Per la prima volta, scegli l'opzione Crea nuova ricetta.
2. Nella sezione Tipo di immagine, scegli l'opzione Amazon Machine Image (AMI) per creare una pipeline di immagini che produrrà e distribuirà un'AMI.
3. Nella sezione Generale, inserisci le seguenti caselle obbligatorie:

   - Nome: il nome della tua ricetta
   - Versione: la versione della tua ricetta (usa il formato) <major>. <minor>. <patch>, dove major, minor e patch sono valori interi). Le nuove ricette generalmente iniziano con`1.0.0`.

4. Nella sezione Immagine sorgente, mantieni i valori predefiniti per Select image, Image Operating System (OS) e Image origin. Il risultato è un elenco AMIs di Linux gestiti da Amazon. Per questo tutorial, seleziona l'`Amazon Linux 2 x86`immagine.

   a. Dal menu a discesa Nome immagine, scegli un'immagine.

   b. Mantieni l'impostazione predefinita per le opzioni di controllo automatico delle versioni (utilizza l'ultima versione del sistema operativo disponibile).

   > **Note**
   >
   > Questa impostazione garantisce che la pipeline utilizzi il controllo delle versioni semantiche per l'immagine di base, per rilevare gli aggiornamenti delle dipendenze per i lavori pianificati automaticamente. Per ulteriori informazioni sul controllo delle versioni semantiche per le risorse di Image Builder, vedere. Versionamento semantico in Image Builder

5. Nella sezione Configurazione dell'istanza, mantieni i valori predefiniti per l'agente Systems Manager. Ciò fa sì che Image Builder mantenga l'agente Systems Manager dopo il completamento della build e dei test, per includere l'agente Systems Manager nella nuova immagine.

   Mantieni vuoti i dati utente per questo tutorial. Puoi utilizzare quest'area altre volte per fornire comandi o uno script di comando da eseguire all'avvio dell'istanza di build. Tuttavia, sostituisce tutti i comandi che Image Builder potrebbe aver aggiunto per garantire l'installazione di Systems Manager. Quando lo usi, assicurati che l'agente Systems Manager sia preinstallato sull'immagine di base o che includa l'installazione nei dati utente.

6. Nella sezione Componenti, è necessario scegliere almeno un componente di compilazione.

   Nel pannello Crea componenti, scegliete Aggiungi componenti di compilazione e selezionatelo `Amazon managed` dall'elenco dei filtri del proprietario del componente. Si apre un pannello di selezione sul lato destro dell'interfaccia della console in cui è possibile sfogliare e filtrare i componenti disponibili.

   Per questo tutorial, scegli un componente che aggiorni Linux con gli ultimi aggiornamenti di sicurezza, come segue:

   a. Filtra i risultati inserendo la parola `update` nella barra di ricerca che si trova nella parte superiore del pannello.

b. Seleziona la casella di controllo per il componente di `update-linux` compilazione.

c. Mantieni l'impostazione predefinita per le opzioni di controllo delle versioni (utilizza l'ultima versione disponibile).

> **Note**
>
> Questa impostazione garantisce che la pipeline utilizzi il controllo delle versioni semantiche per il componente selezionato, per rilevare gli aggiornamenti delle dipendenze per i lavori pianificati automaticamente. Per ulteriori informazioni sul controllo delle versioni semantiche per le risorse di Image Builder, vedere. Versionamento semantico in Image Builder

d. Scegliete Aggiungi alla ricetta per aggiungere il componente alla ricetta. Questo chiude il pannello di selezione dei componenti.

e. Tornando al pannello Build components, viene visualizzato il componente che avete aggiunto.

7. Riordina i componenti (opzionale)

Se avete scelto più di un componente da includere nell'immagine, potete utilizzare l' drag-and-dropazione per riorganizzarli nell'ordine in cui devono essere eseguiti durante il processo di creazione.

> **Note**
>
> I componenti di protezione avanzata CIS non seguono le regole standard di ordinamento dei componenti nelle ricette di Image Builder. I componenti di protezione avanzata CIS funzionano sempre per ultimi per garantire che i test di benchmark vengano eseguiti rispetto all'immagine di output.

a. Ripeti i passaggi precedenti per aggiungere il `update-linux-kernel-5` componente alla tua ricetta.

b. Il componente che hai appena aggiunto ha un parametro di input per la versione del kernel. Per espandere le impostazioni per le opzioni di controllo delle versioni o i parametri di input, puoi scegliere la freccia accanto al nome dell'impostazione. Per espandere tutte le impostazioni per tutti i componenti selezionati, è possibile disattivare e attivare l'opzione Espandi tutto. Per ulteriori informazioni sull'utilizzo dei parametri di input nei componenti e

sulla loro impostazione nelle ricette, consulta. Tutorial: creare un componente personalizzato con parametri di input

c. Scegliete uno dei componenti e trascinatelo verso l'alto o verso il basso per modificare l'ordine in cui verranno eseguiti i componenti.

d. Per rimuovere il `update-linux-kernel-5` componente, X sceglietelo nell'angolo in alto a destra della casella dei componenti.

   Ripeti questo passaggio per rimuovere eventuali altri componenti che potresti aver aggiunto, lasciando selezionato solo il `update-linux` componente.

8. Scegli Avanti per procedere al passaggio successivo.

Fase 3: Definizione della configurazione dell'infrastruttura (opzionale)

Image Builder avvia EC2 istanze nell'account per personalizzare le immagini ed eseguire test di convalida. Le impostazioni di configurazione dell'infrastruttura specificano i dettagli dell'infrastruttura per le istanze che verranno eseguite nell'utente durante il Account AWS processo di creazione.

Nella sezione Configurazione dell'infrastruttura, l'impostazione predefinita delle opzioni di configurazione è`Create infrastructure configuration using service defaults`. Questo crea un ruolo IAM e un profilo di istanza associato per le istanze di EC2 compilazione e test utilizzate per configurare l'immagine. Per ulteriori informazioni sulle impostazioni di configurazione dell'infrastruttura, vedere CreateInfrastructureConfigurationEC2 Image Builder API Reference.

Per questo tutorial, utilizziamo le impostazioni predefinite.

> **Note**
>
> Per specificare una sottorete da utilizzare per un VPC privato, puoi creare la tua configurazione di infrastruttura personalizzata o utilizzare le impostazioni che hai già creato.

- Scegli Avanti per procedere al passaggio successivo.

Fase 4: Definizione delle impostazioni di distribuzione (opzionale)

Le configurazioni di distribuzione includono il nome dell'AMI di output, impostazioni regionali specifiche per la crittografia, le autorizzazioni di avvio e Account AWS le organizzazioni e le unità organizzative (OUs) che possono avviare l'AMI di output e le configurazioni di licenza.

Nella sezione Impostazioni di distribuzione, l'impostazione predefinita delle opzioni di configurazione è. `Create distribution settings using service defaults` Questa opzione distribuirà l'AMI di uscita nella regione corrente. Per ulteriori informazioni sulla configurazione delle impostazioni di distribuzione, consultaGestione delle impostazioni di distribuzione di Image Builder.

Per questo tutorial, utilizziamo le impostazioni predefinite.

- Scegli Avanti per procedere al passaggio successivo.

Fase 5. Esamina

La sezione Revisione mostra tutte le impostazioni che hai configurato. Per modificare le informazioni in una determinata sezione, scegli il pulsante Modifica situato nell'angolo in alto a destra della sezione del passaggio. Ad esempio, se desideri modificare il nome della pipeline, scegli il pulsante Modifica nell'angolo in alto a destra della sezione Fase 1: Dettagli della pipeline.

1. Dopo aver esaminato le impostazioni, scegli Crea pipeline per creare la pipeline.
2. Puoi visualizzare i messaggi di successo o di fallimento nella parte superiore della pagina, man mano che le risorse vengono create per le impostazioni di distribuzione, la configurazione dell'infrastruttura, la nuova ricetta e la pipeline. Per visualizzare i dettagli di una risorsa, incluso l'identificatore della risorsa, scegli Visualizza dettagli.
3. Dopo aver visualizzato i dettagli di una risorsa, puoi visualizzare i dettagli su altre risorse scegliendo il tipo di risorsa dal riquadro di navigazione. Ad esempio, per visualizzare i dettagli della nuova pipeline, scegli Image pipelines dal pannello di navigazione. Se la build ha avuto successo, la nuova pipeline viene visualizzata nell'elenco Image pipelines.

Fase 6: pulizia

Il tuo ambiente Image Builder, proprio come la tua abitazione, necessita di una manutenzione regolare per aiutarti a trovare ciò di cui hai bisogno e a completare le tue attività senza sprecare terreno. Assicurati di pulire regolarmente le risorse temporanee che hai creato per i test. Altrimenti,

potresti dimenticarti di quelle risorse e poi non ricordare per cosa sono state utilizzate. A quel punto, potrebbe non essere chiaro se è possibile liberarsene in sicurezza.

> **Tip**
>
> Per evitare errori di dipendenza quando elimini risorse, assicurati di eliminare le risorse nell'ordine seguente:
>
> 1. Pipeline di immagini
> 2. Ricetta di immagini
> 3. Tutte le risorse rimanenti

Per ripulire le risorse che hai creato per questo tutorial, segui questi passaggi:

Eliminare la pipeline

1. Per visualizzare un elenco delle pipeline di compilazione create con il tuo account, scegli Image pipelines dal pannello di navigazione.
2. Seleziona la casella di controllo accanto a Pipeline name per selezionare la pipeline che desideri eliminare.
3. Nella parte superiore del pannello Image pipelines, nel menu Azioni, scegliete Elimina.
4. Per confermare l'eliminazione, inserite `Delete` nella casella e scegliete Elimina.

Elimina la ricetta

1. Per visualizzare un elenco delle ricette create con il tuo account, scegli Ricette con immagini dal pannello di navigazione.
2. Seleziona la casella di controllo accanto a Nome della ricetta per selezionare la ricetta che desideri eliminare.
3. Nella parte superiore del pannello Ricette con immagini, nel menu Azioni, scegli Elimina ricetta.
4. Per confermare l'eliminazione, entra `Delete` nella casella e scegli Elimina.

Elimina la configurazione dell'infrastruttura

1. Per visualizzare un elenco delle configurazioni dell'infrastruttura create con il tuo account, scegli Configurazione dell'infrastruttura dal pannello di navigazione.
2. Seleziona la casella di controllo accanto a Nome della configurazione per selezionare la configurazione dell'infrastruttura che desideri eliminare.
3. Nella parte superiore del pannello Configurazioni dell'infrastruttura, scegli Elimina.
4. Per confermare l'eliminazione, entra `Delete` nella casella e scegli Elimina.

Eliminare le impostazioni di distribuzione

1. Per visualizzare un elenco delle impostazioni di distribuzione create nel tuo account, scegli Impostazioni di distribuzione dal pannello di navigazione.
2. Seleziona la casella di controllo accanto a Nome di configurazione per selezionare le impostazioni di distribuzione che hai creato per questo tutorial.
3. Nella parte superiore del pannello delle impostazioni di distribuzione, scegli Elimina.
4. Per confermare l'eliminazione, entra `Delete` nella casella e scegli Elimina.

Elimina l'immagine

Segui questi passaggi per verificare di aver eliminato qualsiasi immagine creata dalla pipeline del tutorial. È improbabile che questo tutorial crei un'immagine a meno che non sia trascorso abbastanza tempo dalla creazione della pipeline affinché venga eseguita, secondo la pianificazione di compilazione.

1. Per visualizzare un elenco delle immagini create con il tuo account, scegli Immagini dal pannello di navigazione.
2. Scegli la versione dell'immagine che desideri rimuovere. Si apre la pagina delle versioni di compilazione dell'immagine.
3. Seleziona la casella di controllo accanto alla Versione per qualsiasi immagine che desideri eliminare. È possibile selezionare più di una versione dell'immagine alla volta.
4. Nella parte superiore del pannello Image build versions, scegli Elimina versione.
5. Per confermare l'eliminazione, entra `Delete` nella casella e scegli Elimina.

Tutorial: crea una pipeline di immagini con l'immagine del contenitore Docker in uscita dalla procedura guidata della console Image Builder

Questo tutorial illustra la creazione di una pipeline automatizzata per creare e gestire un' EC2 immagine Docker Image Builder personalizzata utilizzando la procedura guidata della console Create image pipeline. Per aiutarti a eseguire i passaggi in modo efficiente, vengono utilizzate le impostazioni predefinite quando sono disponibili e le sezioni facoltative vengono ignorate.

Crea un flusso di lavoro basato sulla pipeline di immagini

- Fase 1: Specificare i dettagli della pipeline
- Fase 2: Scegli la ricetta
- Fase 3: Definizione della configurazione dell'infrastruttura (opzionale)
- Fase 4: Definizione delle impostazioni di distribuzione (opzionale)
- Fase 5. Esamina
- Fase 6: pulizia

Fase 1: Specificare i dettagli della pipeline

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Per iniziare a creare la pipeline, scegli Crea pipeline di immagini.
3. Nella sezione Generale, inserisci il nome della tua pipeline (obbligatorio).
4. Nella sezione Pianificazione della creazione, puoi mantenere le impostazioni predefinite per le opzioni di pianificazione. Tieni presente che il fuso orario mostrato per la pianificazione predefinita è Universal Coordinated Time (UTC). Per ulteriori informazioni sull'ora UTC e per trovare lo scostamento del fuso orario, vedere Abbreviazioni dei fusi orari - Elenco mondiale.

   Per le impostazioni di aggiornamento delle dipendenze, scegli l'opzione Esegui la pipeline all'ora pianificata se sono disponibili aggiornamenti delle dipendenze. Questa impostazione fa sì che la pipeline verifichi la presenza di aggiornamenti prima di iniziare la compilazione. Se non ci sono aggiornamenti, salta la compilazione pianificata della pipeline.

> **Note**
>
> Per garantire che la pipeline riconosca gli aggiornamenti delle dipendenze e crei le build come previsto, è necessario utilizzare il controllo delle versioni semantiche (x.x.x) per l'immagine di base e i componenti. Per ulteriori informazioni sul controllo delle versioni semantiche per le risorse di Image Builder, vedere. Versionamento semantico in Image Builder

5. Scegli Avanti per procedere al passaggio successivo.

Fase 2: Scegli la ricetta

1. L'impostazione predefinita di Image Builder è Usa la ricetta esistente nella sezione Ricetta. Per la prima volta, scegli l'opzione Crea nuova ricetta.
2. Nella sezione Tipo di immagine, scegli l'opzione Docker image per creare una pipeline di container che produrrà un'immagine Docker e la distribuirà nei repository Amazon ECR nelle regioni di destinazione.
3. Nella sezione Generale, inserisci le seguenti caselle obbligatorie:
   - Nome: il nome della tua ricetta
   - Versione: la versione della tua ricetta (usa il formato) <major>. <minor>. <patch>, dove major, minor e patch sono valori interi). Le nuove ricette generalmente iniziano con`1.0.0`.
4. Nella sezione Immagine sorgente, mantieni i valori predefiniti per Select image, Image Operating System (OS) e Image origin. Il risultato è un elenco di immagini di container Amazon Linux 2, gestite da Amazon, tra cui scegliere per l'immagine di base.
   a. Dal menu a discesa Nome immagine, scegli un'immagine.
   b. Mantieni l'impostazione predefinita per le opzioni di controllo automatico delle versioni (utilizza l'ultima versione del sistema operativo disponibile).

> **Note**
>
> Questa impostazione garantisce che la pipeline utilizzi il controllo delle versioni semantiche per l'immagine di base, per rilevare gli aggiornamenti delle dipendenze per i lavori pianificati automaticamente. Per ulteriori informazioni sul controllo

delle versioni semantiche per le risorse di Image Builder, vedere. Versionamento semantico in Image Builder

5. Nella sezione Componenti, è necessario scegliere almeno un componente di compilazione.

   Nel pannello Build components — Amazon Linux, puoi sfogliare i componenti elencati nella pagina. Usa il controllo di impaginazione nell'angolo in alto a destra per navigare tra i componenti aggiuntivi disponibili per il tuo sistema operativo con immagini di base. Puoi anche cercare componenti specifici o creare il tuo componente di compilazione utilizzando il Gestore dei componenti.

   Per questo tutorial, scegli un componente che aggiorni Linux con gli ultimi aggiornamenti di sicurezza, come segue:

   a. Filtra i risultati inserendo la parola `update` nella barra di ricerca che si trova nella parte superiore del pannello.
   b. Seleziona la casella di controllo per il componente di `update-linux` compilazione.
   c. Scorri verso il basso e, nell'angolo in alto a destra dell'elenco Componenti selezionati, scegli Espandi tutto.
   d. Mantieni l'impostazione predefinita per le opzioni di controllo delle versioni (utilizza l'ultima versione del componente disponibile).

      > **Note**
      >
      > Questa impostazione garantisce che la pipeline utilizzi il controllo delle versioni semantiche per il componente selezionato, per rilevare gli aggiornamenti delle dipendenze per i lavori pianificati automaticamente. Per ulteriori informazioni sul controllo delle versioni semantiche per le risorse di Image Builder, vedere. Versionamento semantico in Image Builder

      Se avessi selezionato un componente con parametri di input, i parametri sarebbero visibili anche in quest'area. I parametri non sono trattati in questo tutorial. Per ulteriori informazioni sull'utilizzo dei parametri di input nei componenti e sulla loro impostazione nelle ricette, consultaTutorial: creare un componente personalizzato con parametri di input.

   Riordinare i componenti (opzionale)

Se avete scelto più di un componente da includere nell'immagine, potete utilizzare l' drag-and-dropazione per riorganizzarli nell'ordine in cui devono essere eseguiti durante il processo di creazione.

> **Note**
>
> I componenti di protezione avanzata CIS non seguono le regole standard di ordinamento dei componenti nelle ricette di Image Builder. I componenti di protezione avanzata CIS funzionano sempre per ultimi per garantire che i test di benchmark vengano eseguiti rispetto all'immagine di output.

1. Scorri verso l'alto fino all'elenco dei componenti disponibili.
2. Seleziona la casella di controllo per il componente di `update-linux-kernel-mainline` compilazione (o qualsiasi altro componente a tua scelta).
3. Scorri verso il basso fino all'elenco dei componenti selezionati per vedere che ci sono almeno due risultati.
4. È possibile che il controllo delle versioni dei componenti appena aggiunti non sia stato ampliato. Per espandere le opzioni di controllo delle versioni, puoi scegliere la freccia accanto alle opzioni di controllo delle versioni oppure puoi disattivare e attivare l'opzione Espandi tutto per espandere il controllo delle versioni per tutti i componenti selezionati.
5. Scegliete uno dei componenti e trascinatelo verso l'alto o verso il basso per modificare l'ordine in cui verranno eseguiti i componenti.
6. Per rimuovere il `update-linux-kernel-mainline` componente, selezionalo `X` dall'angolo in alto a destra della casella dei componenti.
7. Ripeti il passaggio precedente per rimuovere eventuali altri componenti che potresti aver aggiunto, lasciando selezionato solo il `update-linux` componente.

6. Nella sezione modello Dockerfile, seleziona l'opzione Usa esempio. Nel pannello Contenuto, notate le variabili contestuali in cui Image Builder inserisce le informazioni di costruzione o gli script, in base alla ricetta dell'immagine del contenitore.

   Per impostazione predefinita, Image Builder utilizza le seguenti variabili contestuali nel tuo Dockerfile.

parentImage (obbligatorio)

In fase di compilazione, questa variabile diventa l'immagine di base della ricetta.

Esempio:

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

ambienti (richiesti se vengono specificati i componenti)

Questa variabile si risolverà in uno script che esegue i componenti.

Esempio:

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

componenti (opzionale)

Image Builder risolve gli script dei componenti di compilazione e test per i componenti inclusi nella ricetta del contenitore. Questa variabile può essere posizionata ovunque nel Dockerfile, dopo la variabile environment.

Esempio:

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

7. Nella sezione Archivio Target, specifica il nome del repository Amazon ECR che hai creato come prerequisito per questo tutorial. Questo repository viene utilizzato come impostazione predefinita per la configurazione di distribuzione nella regione in cui viene eseguita la pipeline (Regione 1).

> **Note**
>
> Il repository di destinazione deve esistere in Amazon ECR per tutte le regioni di destinazione prima della distribuzione.

8. Scegli Avanti per procedere al passaggio successivo.

Fase 3: Definizione della configurazione dell'infrastruttura (opzionale)

Image Builder avvia EC2 istanze nell'account per personalizzare le immagini ed eseguire test di convalida. Le impostazioni di configurazione dell'infrastruttura specificano i dettagli dell'infrastruttura per le istanze che verranno eseguite durante il Account AWS processo di creazione.

Nella sezione Configurazione dell'infrastruttura, l'impostazione predefinita delle opzioni di configurazione è`Create infrastructure configuration using service defaults`. Questo crea un ruolo IAM e un profilo di istanza associato che vengono utilizzati dalle istanze di compilazione per configurare le immagini dei contenitori. Puoi anche creare la tua configurazione di infrastruttura personalizzata o utilizzare impostazioni che hai già creato. Per ulteriori informazioni sulle impostazioni di configurazione dell'infrastruttura, vedere CreateInfrastructureConfigurationEC2 Image Builder API Reference.

Per questo tutorial, utilizziamo le impostazioni predefinite.

- Scegli Avanti per procedere al passaggio successivo.

Fase 4: Definizione delle impostazioni di distribuzione (opzionale)

Le impostazioni di distribuzione sono costituite dalle regioni di destinazione e dal nome del repository Amazon ECR di destinazione. Le immagini di Output Docker vengono distribuite nel repository Amazon ECR denominato in ogni regione.

Nella sezione Impostazioni di distribuzione, l'impostazione predefinita delle opzioni di configurazione è. `Create distribution settings using service defaults` Questa opzione distribuirà l'immagine Docker di output nel repository Amazon ECR specificato nella ricetta del contenitore per la regione in cui viene eseguita la pipeline (Regione 1). Se lo desideri`Create new distribution settings`, puoi sovrascrivere il repository ECR per la regione corrente e aggiungere altre regioni per la distribuzione.

Per questo tutorial, utilizziamo le impostazioni predefinite.

- Scegli Avanti per procedere al passaggio successivo.

Fase 5. Esamina

La sezione Revisione mostra tutte le impostazioni che hai configurato. Per modificare le informazioni in una determinata sezione, scegli il pulsante Modifica situato nell'angolo in alto a destra della

sezione del passaggio. Ad esempio, se desideri modificare il nome della pipeline, scegli il pulsante Modifica nell'angolo in alto a destra della sezione Fase 1: Dettagli della pipeline.

1. Dopo aver esaminato le impostazioni, scegli Crea pipeline per creare la pipeline.
2. Puoi visualizzare i messaggi di successo o di fallimento nella parte superiore della pagina, man mano che le risorse vengono create per le impostazioni di distribuzione, la configurazione dell'infrastruttura, la nuova ricetta e la pipeline. Per visualizzare i dettagli di una risorsa, incluso l'identificatore della risorsa, scegli Visualizza dettagli.
3. Dopo aver visualizzato i dettagli di una risorsa, puoi visualizzare i dettagli su altre risorse scegliendo il tipo di risorsa dal riquadro di navigazione. Ad esempio, per visualizzare i dettagli della nuova pipeline, scegli Image pipelines dal pannello di navigazione. Se la build ha avuto successo, la nuova pipeline viene visualizzata nell'elenco Image pipelines.

Fase 6: pulizia

Il tuo ambiente Image Builder, proprio come la tua abitazione, necessita di una manutenzione regolare per aiutarti a trovare ciò di cui hai bisogno e a completare le tue attività senza sprecare terreno. Assicurati di pulire regolarmente le risorse temporanee che hai creato per i test. Altrimenti, potresti dimenticarti di quelle risorse e poi non ricordare per cosa sono state utilizzate. A quel punto, potrebbe non essere chiaro se è possibile liberarsene in sicurezza.

> **Tip**
>
> Per evitare errori di dipendenza quando elimini risorse, assicurati di eliminare le risorse nell'ordine seguente:
>
> 1. Pipeline di immagini
> 2. Ricetta di immagini
> 3. Tutte le risorse rimanenti

Per ripulire le risorse che hai creato per questo tutorial, segui questi passaggi:

Eliminare la pipeline

1. Per visualizzare un elenco delle pipeline di compilazione create con il tuo account, scegli Image pipelines dal pannello di navigazione.

2. Seleziona la casella di controllo accanto a Pipeline name per selezionare la pipeline che desideri eliminare.
3. Nella parte superiore del pannello Image pipelines, nel menu Azioni, scegliete Elimina.
4. Per confermare l'eliminazione, inserite `Delete` nella casella e scegliete Elimina.

Elimina la ricetta del contenitore

1. Per visualizzare un elenco delle ricette in contenitore create con il tuo account, scegli Ricette in contenitore dal pannello di navigazione.
2. Seleziona la casella di controllo accanto a Nome della ricetta per selezionare la ricetta che desideri eliminare.
3. Nella parte superiore del pannello Ricette del contenitore, nel menu Azioni, scegli Elimina ricetta.
4. Per confermare l'eliminazione, entra `Delete` nella casella e scegli Elimina.

Elimina la configurazione dell'infrastruttura

1. Per visualizzare un elenco delle configurazioni dell'infrastruttura create con il tuo account, scegli Configurazione dell'infrastruttura dal pannello di navigazione.
2. Seleziona la casella di controllo accanto a Nome della configurazione per selezionare la configurazione dell'infrastruttura che desideri eliminare.
3. Nella parte superiore del pannello Configurazioni dell'infrastruttura, scegli Elimina.
4. Per confermare l'eliminazione, entra `Delete` nella casella e scegli Elimina.

Eliminare le impostazioni di distribuzione

1. Per visualizzare un elenco delle impostazioni di distribuzione create nel tuo account, scegli Impostazioni di distribuzione dal pannello di navigazione.
2. Seleziona la casella di controllo accanto a Nome di configurazione per selezionare le impostazioni di distribuzione che hai creato per questo tutorial.
3. Nella parte superiore del pannello delle impostazioni di distribuzione, scegli Elimina.
4. Per confermare l'eliminazione, entra `Delete` nella casella e scegli Elimina.

Elimina l'immagine

Segui questi passaggi per verificare di aver eliminato qualsiasi immagine creata dalla pipeline del tutorial. È improbabile che questo tutorial crei un'immagine a meno che non sia trascorso abbastanza tempo dalla creazione della pipeline affinché venga eseguita, secondo la pianificazione di compilazione.

1. Per visualizzare un elenco delle immagini create con il tuo account, scegli Immagini dal pannello di navigazione.
2. Scegli la versione dell'immagine che desideri rimuovere. Si apre la pagina delle versioni di compilazione dell'immagine.
3. Seleziona la casella di controllo accanto alla Versione per qualsiasi immagine che desideri eliminare. È possibile selezionare più di una versione dell'immagine alla volta.
4. Nella parte superiore del pannello Image build versions, scegli Elimina versione.
5. Per confermare l'eliminazione, entra `Delete` nella casella e scegli Elimina.

Tutorial: creare un componente personalizzato con parametri di input

È possibile gestire i componenti di Image Builder, inclusa la creazione e l'impostazione dei parametri dei componenti, direttamente dalla EC2 console Image Builder, da o AWS CLI dall'API Image Builder oppure. SDKs In questa sezione, tratteremo la creazione e l'utilizzo dei parametri nel componente e l'impostazione dei parametri dei componenti tramite la console Image Builder e AWS CLI i comandi in fase di esecuzione.

> **Important**
>
> I parametri dei componenti sono valori di testo semplice e vengono registrati. AWS CloudTrail Ti consigliamo di utilizzare AWS Secrets Manager o AWS Systems Manager Parameter Store per archiviare i tuoi segreti. Per ulteriori informazioni su Secrets Manager, vedi Cos'è Secrets Manager? nella Guida AWS Secrets Manager per l'utente. Per ulteriori informazioni su AWS Systems Manager Parameter Store, vedere AWS Systems Manager Parameter Store nella Guida AWS Systems Manager per l'utente.

Usa i parametri nel documento del componente YAML

Per creare un componente, devi fornire un documento relativo al componente dell'applicazione YAML o JSON. Il documento contiene il codice che viene eseguito durante le fasi e i passaggi definiti per personalizzare l'immagine. La ricetta che fa riferimento al componente può impostare i parametri per personalizzare i valori in fase di esecuzione, con valori predefiniti che hanno effetto se il parametro non è impostato su un valore specifico.

Crea un documento componente con parametri di input

Questa sezione mostra come definire e utilizzare i parametri di input nel documento del componente YAML.

Per creare un documento componente dell'applicazione YAML che utilizzi parametri ed esegua comandi nelle istanze di build o test di Image Builder, segui i passaggi corrispondenti al sistema operativo dell'immagine:

Linux

Crea un documento componente YAML

Utilizzate uno strumento di modifica dei file per creare un file di documento componente. Gli esempi di documentazione utilizzano un file denominato *`hello-world-test.yaml`* che include il seguente contenuto:

```
# 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**
>
> Usa uno strumento come questo validatore YAML online o un'estensione YAML lint nel tuo ambiente di codice per verificare che il tuo YAML sia ben formato.

Windows

Crea un documento componente YAML

Utilizzate uno strumento di modifica dei file per creare un file di documento componente. Gli esempi di documentazione utilizzano un file denominato *`hello-world-test.yaml`* che include il seguente contenuto:

```
# 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**
>
> Usa uno strumento come questo validatore YAML online o un'estensione YAML lint nel tuo ambiente di codice per verificare che il tuo YAML sia ben formato.

macOS

Crea un documento componente YAML

Utilizzate uno strumento di modifica dei file per creare un file di documento componente. Gli esempi di documentazione utilizzano un file denominato *hello-world-test.yaml* che include il seguente contenuto:

```
# Document Start
#
name: "HelloWorldTestingDocument-macOS"
description: "Hello world document to demonstrate parameters."
schemaVersion: 1.0
parameters:
  - MyInputParameter:
      type: string
      default: "It's me!"
      description: This is an input parameter.
phases:
  - name: build
    steps:
      - name: HelloWorldStep
        action: ExecuteBash
        inputs:
          commands:
            - echo "Hello World! Build phase. My input parameter value is
 {{ MyInputParameter }}"

  - name: validate
    steps:
      - name: HelloWorldStep
        action: ExecuteBash
        inputs:
          commands:
            - echo "Hello World! Validate phase. My input parameter value is
 {{ MyInputParameter }}"

  - name: test
    steps:
      - name: HelloWorldStep
        action: ExecuteBash
        inputs:
          commands:
            - echo "Hello World! Test phase. My input parameter value is
 {{ MyInputParameter }}"
# Document End
```

> **Tip**
>
> Usa uno strumento come questo validatore YAML online o un'estensione YAML lint nel tuo ambiente di codice per verificare che il tuo YAML sia ben formato.

Per ulteriori informazioni sulle fasi, i passaggi e la sintassi dei documenti dei componenti dell'applicazione, consulta Use documents in. AWSTOEAWSTOE Per ulteriori informazioni sui parametri e sui relativi requisiti, consultate la Parametri sezione della pagina Definire e fare riferimento alle variabili nella AWSTOE pagina.

Crea un componente dal documento del componente YAML

Qualunque sia il metodo utilizzato per creare un AWSTOE componente, il documento del componente dell'applicazione YAML è sempre necessario come riferimento.

- Per creare un componente direttamente dal documento YAML con la console Image Builder, vedere. Crea un componente personalizzato dalla console
- Per creare un componente dalla riga di comando con il comando Image Builder, create-component vedere. Creare un componente personalizzato dal AWS CLI Sostituite il nome del documento YAML in questi esempi con il nome del documento YAML di Hello World (). *`hello-world-test.yaml`*

Imposta i parametri dei componenti in una ricetta di Image Builder dalla console

L'impostazione dei parametri dei componenti funziona allo stesso modo per le ricette di immagini e le ricette dei contenitori. Quando create una nuova ricetta o una nuova versione di una ricetta, scegliete quali componenti includere dagli elenchi Build components e Test components. Gli elenchi dei componenti includono componenti applicabili al sistema operativo di base scelto per l'immagine.

Dopo aver selezionato un componente, questo viene visualizzato nella sezione Componenti selezionati, direttamente sotto gli elenchi dei componenti. Le opzioni di configurazione sono mostrate per ogni componente selezionato. Se il componente ha dei parametri di input definiti, questi vengono visualizzati come una sezione espandibile denominata Parametri di input.

Le seguenti impostazioni dei parametri sono mostrate per ogni parametro definito per il componente:

- Nome del parametro (non modificabile): il nome del parametro.
- Descrizione (non modificabile): la descrizione del parametro
- Tipo (non modificabile): il tipo di dati per il valore del parametro.
- Valore: il valore del parametro. Se si utilizza questo componente per la prima volta in questa ricetta ed è stato definito un valore predefinito per il parametro di input, il valore predefinito viene visualizzato nella casella Valore con testo in grigio. Se non viene immesso nessun altro valore, Image Builder utilizza il valore predefinito.

Usa un parametro di immagine di base nella tua ricetta

Quando si crea una ricetta per la personalizzazione delle immagini, esistono diversi modi per identificare l'immagine di base da cui iniziare. Se specifichi l'ID Amazon Machine Image (AMI) per l'immagine di base e tale immagine di base viene aggiornata, l'ID AMI potrebbe cambiare e dovrai aggiornare la ricetta in modo che corrisponda.

Invece di modificare la ricetta ogni volta che l'ID dell'immagine di base cambia, puoi definire un AWS Systems Manager parametro Parameter Store (parametro SSM) per memorizzare il valore dell'ID AMI dell'immagine di base e quindi utilizzare il parametro per specificare l'immagine di base nella ricetta. Per AWS managed AMIs, puoi utilizzare un parametro pubblico per la versione più recente.

Questo tutorial illustra il processo di creazione di un parametro ID AMI e il suo utilizzo in una ricetta di immagine. I passaggi di Image Builder di questo tutorial sono basati su console.

Indice

- Fase 1: Trova o crea un parametro Parameter Store
- Passaggio 2: configurare le autorizzazioni IAM
- Passaggio 3: Creare una ricetta di immagini che utilizzi il parametro

Fase 1: Trova o crea un parametro Parameter Store

Il processo di questo passaggio dipende dal tipo di AMI specificato per l'immagine di base. Per AWS managed AMIs, è possibile utilizzare un parametro pubblico che si riferisce alla versione corrente. Alcuni parametri potrebbero non essere disponibili in tutti Regioni AWS.

Per iniziare, apri la scheda corrispondente al tuo AMI.

AWS managed AMI

Se l'immagine di base è un'AMI AWS gestita, puoi utilizzare parametri pubblici per specificare l'ID AMI, anziché creare un parametro personalizzato. Per trovare il parametro pubblico per la tua AMI, consulta Discovering public parameters nella AWS Systems Manager User Guide.

Custom AMI

Per creare un parametro AMI ID, segui le istruzioni per la creazione dei parametri di Parameter Store in Systems Manager con la console AWS CLI, oppure PowerShell. Fornisci i seguenti valori per assicurarti che il valore del parametro sia un ID AMI.

Livello di parametro: `Standard`

Tipo: `String`

Tipo di dati: Selezionaaws:ec2:image. Quando si specifica questo tipo, il sistema convalida il valore immesso per assicurarsi che si tratti di un ID AMI.

Valore: inserisci un ID AMI valido (ad esempio,***ami-1234567890abcdef1***).

Passaggio 2: configurare le autorizzazioni IAM

Per utilizzare un parametro Systems Manager Parameter Store (parametro SSM), pubblico o privato, è necessario specificare le seguenti azioni di Systems Manager Parameter Store nel ruolo di esecuzione di Image Builder, con il parametro elencato come risorsa.

- `ssm:GetParameter`— Questa azione consente di utilizzare un parametro SSM per specificare l'immagine di base nella ricetta.
- `ssm:PutParameter`— Questa azione consente di memorizzare l'ID AMI di output in un parametro SSM durante la distribuzione. La definizione della politica sembra la stessa, ma questo tutorial non include l'azione put nella politica di esempio.

Per utilizzare i parametri SSM in un componente personalizzato, è necessario specificare invece il `ssm:GetParameter` ruolo del profilo dell'istanza. Per ulteriori informazioni, consulta Usa i parametri del Parameter Store di Systems Manager.

Quando si crea una pipeline o si utilizza il comando create-image in AWS CLI, è possibile specificare un solo ruolo di esecuzione di Image Builder. Se è stato definito un ruolo di esecuzione del flusso di lavoro di Image Builder, è necessario aggiungere i permessi dei parametri a tale ruolo. In caso

contrario, è necessario creare un nuovo ruolo personalizzato che include le autorizzazioni necessarie per i parametri SSM.

1. Crea un ruolo personalizzato (opzionale)

   Se hai già definito un ruolo personalizzato per le autorizzazioni di Image Builder, puoi saltare questo passaggio.

   Segui la procedura per la creazione di un ruolo per delegare le autorizzazioni a un AWS servizio nella Guida per l'utente.AWS Identity and Access Management
2. Aggiungi le autorizzazioni al tuo ruolo personalizzato

   Per aggiungere le autorizzazioni del parametro SSM al tuo ruolo personalizzato, segui il processo di aggiornamento della politica di autorizzazione per un ruolo nella Guida per l'utente.AWS Identity and Access Management

   Il seguente esempio di politica mostra l'`ssm:GetParameter`azione con un parametro creato nel tuo account.

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

Per ulteriori informazioni sulle risorse dei parametri pubblici, vedere Chiamata dei parametri pubblici AMI nella Guida per l'AWS Systems Manager utente.

Passaggio 3: Creare una ricetta di immagini che utilizzi il parametro

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Scegli Ricette con immagini, quindi scegli Crea ricetta con immagini dalla pagina dell'elenco.
3. Compila la sezione Immagine di base, come segue:

a. Scegli l'opzione Usa AMI personalizzata. Vengono visualizzati campi aggiuntivi in cui è possibile inserire l'ID AMI o un parametro SSM che contiene l'ID AMI.

b. Scegli l'opzione del parametro SSM.

c. Nel campo del parametro SSM, inserisci il nome del parametro o Amazon Resource Name (ARN) del parametro che hai creato nella fase 1. Se inserisci il nome, non avrà il prefisso nella console.

4. Se necessario, completa la configurazione della ricetta rimanente.

> **Note**
>
> Se impostate l'immagine principale tramite altre interfacce, come la AWS CLI, il nome del parametro deve avere il prefisso `ssm:` (ad esempio,. `ssm:/ImageBuilder-Tutorial/BaseAMI`

Usa i componenti per personalizzare l'immagine di Image Builder

Image Builder utilizza l'applicazione di gestione dei componenti AWS Task Orchestrator and Executor (AWSTOE) per orchestrare flussi di lavoro complessi. I componenti di compilazione e test che funzionano con l' AWSTOE applicazione si basano su documenti YAML che definiscono gli script per personalizzare o testare l'immagine. Per le immagini AMI, Image Builder installa i componenti e l'applicazione di gestione dei AWSTOE componenti nelle sue istanze di EC2 build e test di Amazon. Per le immagini dei container, i componenti e l'applicazione di gestione dei AWSTOE componenti vengono installati all'interno del contenitore in esecuzione.

Image Builder lo utilizza AWSTOE per eseguire tutte le attività su istanza. Non è richiesta alcuna configurazione aggiuntiva con cui interagire AWSTOE quando si eseguono i comandi di Image Builder o si utilizza la console Image Builder.

> **Note**
>
> Quando un componente gestito da Amazon raggiunge la fine del ciclo di vita del supporto, non viene più sottoposto a manutenzione. Circa quattro settimane prima che ciò si verifichi, tutti gli account che utilizzano il componente ricevono una notifica e un elenco delle ricette interessate presenti nel proprio account. AWS Health Dashboard Per ulteriori informazioni AWS Health, consulta la Guida AWS Health per l'utente.

Fasi del flusso di lavoro per la creazione di una nuova immagine

Il flusso di lavoro di Image Builder per la creazione di nuove immagini include le due fasi distinte seguenti.

1. Fase di creazione (pre-snapshot): durante la fase di creazione, apporti modifiche all'istanza di EC2 build di Amazon che esegue la tua immagine di base, per creare la linea di base per la tua nuova immagine. Ad esempio, la ricetta può includere componenti che installano un'applicazione o modificano le impostazioni del firewall del sistema operativo.

   Le seguenti fasi del documento del componente vengono eseguite durante la fase di creazione:

   - build
   - validate

Una volta completata con successo questa fase, Image Builder crea un'istantanea o un'immagine contenitore che utilizza per la fase di test e oltre.

2. Fase di test (post-snapshot): durante la fase di test, ci sono alcune differenze tra le immagini di creazione e le immagini del contenitore. AMIs Per i flussi di lavoro AMI, Image Builder avvia EC2 un'istanza dall'istantanea creata come fase finale della fase di creazione. I test vengono eseguiti sulla nuova istanza per convalidare le impostazioni e garantire che l'istanza funzioni come previsto. Per i flussi di lavoro in container, i test vengono eseguiti sulla stessa istanza utilizzata per la creazione.

   La fase seguente del documento relativo ai componenti viene eseguita per ogni componente incluso nella ricetta durante la fase di test di creazione dell'immagine:

   - test

   Questa fase dei componenti si applica a entrambi i tipi di componenti Build e Test. Una volta completata con successo questa fase, Image Builder può creare e distribuire l'immagine finale dall'istantanea o dall'immagine del contenitore.

> **Note**
>
> Sebbene il framework AWSTOE applicativo consenta di definire molte fasi in un documento componente, Image Builder ha regole rigide sulle fasi di esecuzione e sulle fasi in cui le esegue. Affinché un componente venga eseguito durante la fase di creazione dell'immagine, il documento del componente deve definire almeno una di queste fasi: `build` `ovalidate`. Affinché un componente venga eseguito durante la fase di test dell'immagine, il documento del componente deve definire la `test` fase e non altre fasi.
> Poiché Image Builder esegue le fasi in modo indipendente, il concatenamento dei riferimenti nei documenti dei componenti non può oltrepassare i limiti delle fasi. Non è possibile concatenare un valore da una fase in esecuzione nella fase di creazione a una fase in esecuzione nella fase di test. Tuttavia, è possibile definire i parametri di input alla destinazione desiderata e passare i valori tramite la riga di comando. Per ulteriori informazioni sull'impostazione dei parametri dei componenti nelle ricette di Image Builder, vedere. Tutorial: creare un componente personalizzato con parametri di input

Per facilitare la risoluzione dei problemi sull'istanza di build o test, AWSTOE crea una cartella di registro che contiene il documento di input e i file di registro per tenere traccia di ciò che accade

ogni volta che viene eseguito un componente. Se hai configurato un bucket Amazon S3 nella configurazione della pipeline, anche i log vengono scritti lì. Per ulteriori informazioni sui documenti YAML e sull'output dei log, consulta. Usa il framework AWSTOE dei documenti dei componenti per i componenti personalizzati

> **Tip**
>
> Quando hai molti componenti di cui tenere traccia, l'etichettatura ti aiuta a identificare un componente o una versione specifici in base ai tag che gli hai assegnato. Per ulteriori informazioni sull'etichettatura delle risorse utilizzando i comandi Image Builder in, AWS CLI consultate Aggiunta di tag alle risorse la sezione di questa guida.

Questa sezione spiega come elencare, visualizzare, creare e importare componenti utilizzando la console Image Builder o i comandi di. AWS CLI

Argomenti

- Elenca e visualizza i dettagli dei componenti
- Usa Marketplace AWS i componenti per personalizzare l'immagine
- Usa componenti gestiti per personalizzare l'immagine di Image Builder
- Sviluppa componenti personalizzati per la tua immagine Image Builder
- In che modo Image Builder utilizza l' AWS Task Orchestrator and Executor applicazione per gestire i componenti

Elenca e visualizza i dettagli dei componenti

Questa sezione descrive come trovare informazioni e visualizzare dettagli sui componenti utilizzati nelle ricette di EC2 Image Builder.

Dettagli dei componenti

- Elenca i componenti di Image Builder
- Elenca le versioni di build dei componenti da AWS CLI
- Ottenete i dettagli dei componenti dal AWS CLI
- Ottieni dettagli sulla politica dei componenti da AWS CLI

Elenca i componenti di Image Builder

È possibile utilizzare uno dei seguenti metodi per elencare e filtrare i componenti di Image Builder.

AWS Management Console

Per visualizzare un elenco di componenti in AWS Management Console, procedi nel seguente modo:

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Seleziona Componenti dal pannello di navigazione. Per impostazione predefinita, Image Builder mostra un elenco di componenti di proprietà dell'account.
3. Facoltativamente, è possibile filtrare in base alla proprietà dei componenti. Per visualizzare i componenti di cui non sei proprietario, ma a cui hai accesso, espandi l'elenco a discesa del tipo di proprietario e seleziona uno dei valori. L'elenco dei tipi di proprietario si trova nella barra di ricerca, accanto alla casella di testo di ricerca. Puoi scegliere fra i seguenti valori:
   - Marketplace AWS— Componenti associati direttamente all'abbonamento di un Marketplace AWS prodotto.
   - Avvio rapido (gestito da Amazon): componenti disponibili pubblicamente che Amazon crea e gestisce.
   - Di mia proprietà: componenti creati da te. Questa è la selezione predefinita.
   - Condivisi con me: componenti che altri hanno creato e condiviso con te dal loro account.
   - Gestiti da terze parti: componenti di proprietà di terzi a cui ti sei abbonato. Marketplace AWS

AWS CLI

L'esempio seguente mostra come utilizzare il list-components comando per restituire un elenco di componenti Image Builder di proprietà dell'account.

```
aws imagebuilder list-components
```

Facoltativamente, è possibile filtrare in base alla proprietà dei componenti. L'attributo owner definisce chi possiede i componenti che si desidera elencare. Per impostazione predefinita, questa richiesta restituisce un elenco di componenti di proprietà dell'account. Per filtrare i

risultati in base al proprietario del componente, specifica uno dei seguenti valori con il `--owner` parametro quando esegui il list-components comando.

Valori del proprietario del componente

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

Gli esempi seguenti mostrano il list-components comando con il `--owner` parametro per filtrare i risultati.

```
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
```

Elenca le versioni di build dei componenti da AWS CLI

L'esempio seguente mostra come utilizzare il list-component-build-versions comando per elencare le versioni di build dei componenti che hanno una versione semantica specifica. Per ulteriori informazioni sul controllo delle versioni semantiche per le risorse di Image Builder, vedere. Versionamento semantico in Image Builder

```
aws imagebuilder list-component-build-versions --component-version-arn
 arn:aws:imagebuilder:us-west-2:123456789012:component/example-component/1.0.1
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "componentSummaryList": [
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:component/
examplecomponent/1.0.1/1",
            "name": "examplecomponent",
            "version": "1.0.1",
            "platform": "Linux",
            "type": "BUILD",
            "owner": "123456789012",
            "description": "An example component that builds, validates and tests an
 image",
            "changeDescription": "Updated version.",
            "dateCreated": "2020-02-19T18:53:45.940Z",
            "tags": {
                "KeyName": "KeyValue"
            }
        }
    ]
}
```

Ottenete i dettagli dei componenti dal AWS CLI

L'esempio seguente mostra come utilizzare il get-component comando per ottenere i dettagli del componente quando si specifica l'Amazon Resource Name (ARN) del componente.

```
aws imagebuilder get-component --component-build-version-arn arn:aws:imagebuilder:us-
west-2:123456789012:component/example-component/1.0.1/1
   {
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11112",
    "component": {
        "arn": "arn:aws:imagebuilder:us-west-2:123456789012:component/
examplecomponent/1.0.1/1",
        "name": "examplecomponent",
        "version": "1.0.1",
        "type": "BUILD",
        "platform": "Linux",
        "owner": "123456789012",
        "data": "name: HelloWorldTestingDocument\ndescription: This is hello world
 testing document... etc.\"\n",
        "encrypted": true,
        "dateCreated": "2020-09-24T16:58:24.444Z",
        "tags": {}
    }
}
```

Ottieni dettagli sulla politica dei componenti da AWS CLI

L'esempio seguente mostra come utilizzare il get-component-policy comando per ottenere dettagli sulla politica di un componente quando si specifica l'ARN del componente.

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

Usa Marketplace AWS i componenti per personalizzare l'immagine

Oltre a un'ampia selezione di immagini create da Independent Software Vendors (ISVs), Marketplace AWS offre componenti che è possibile utilizzare per personalizzare le immagini di Image Builder. È necessario abbonarsi a questi Marketplace AWS componenti prima di poterli utilizzare nella ricetta delle immagini per creare una nuova immagine.

Quando si specifica un Marketplace AWS componente in una ricetta di immagini, Image Builder convalida l'abbonamento ed esegue controlli di dipendenza per garantire che l'utente disponga delle risorse necessarie per utilizzarlo. Quando la convalida ha esito positivo, Image Builder crea download sicuri per il componente e i relativi artefatti da utilizzare nelle build di pipeline di immagini.

Scopri i componenti Marketplace AWS

È possibile scoprire i componenti Marketplace AWS software da utilizzare nelle ricette dalla pagina Scopri i prodotti nella console Image Builder, come segue.

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Dal pannello di navigazione, scegli Scopri i prodotti nella Marketplace AWSsezione.
3. Seleziona la scheda Components (Componenti). Questa scheda elenca tutti i Marketplace AWS prodotti che utilizzano l'opzione di consegna che include i componenti associati in Marketplace AWS.
4. Per cercare prodotti software specifici che includono componenti, puoi inserire parte del nome nella barra di ricerca o filtrare per `StatusOperating System`,`Publisher`, o`Categories`. La barra di ricerca contiene anche i controlli di impaginazione per i risultati.

Risultati

Ogni Marketplace AWS prodotto ha il proprio pannello di dettaglio che include le seguenti informazioni.

Il nome e il logo del Marketplace AWS prodotto

Il nome del prodotto software è collegato ai dettagli del prodotto in Marketplace AWS. È possibile selezionare il collegamento per ulteriori informazioni sul prodotto in Marketplace AWS. In alternativa, puoi visualizzare un riepilogo delle opzioni di abbonamento e iscriverti direttamente dai risultati della ricerca con il pulsante Visualizza le opzioni di abbonamento se hai già fatto la tua ricerca.

Versione

Contiene la versione principale del componente.

Sistema operativo

Il sistema operativo su cui il componente è progettato per funzionare.

Editore

L'editore del componente. Questo è collegato alla pagina dei dettagli dell'editore in Marketplace AWS. La pagina dei dettagli dell'editore si apre in una nuova scheda del browser.

Categorie

Una o più categorie di Marketplace AWS prodotti applicabili al componente.

Stato

Indica se sei abbonato a questo prodotto. Se non sei abbonato, puoi scegliere Visualizza le opzioni di abbonamento per visualizzare un riepilogo delle opzioni di abbonamento per il Marketplace AWS prodotto e, facoltativamente, abbonarti direttamente dalla console Image Builder.

Componenti associati

Se il Marketplace AWS prodotto ha una o più versioni incluse nell'abbonamento, queste vengono visualizzate nella sezione Componenti associati. La sezione è inizialmente compressa e mostra il conteggio dei componenti associati. Puoi espandere la sezione per visualizzare ulteriori dettagli.

> **Note**
>
> Il componente Center for Internet Security (CIS) associato al relativo prodotto Marketplace AWS illustrato non viene visualizzato nei risultati dei prodotti Discover. Se ti abboni al loro prodotto illustrato, il componente associato viene mostrato nella pagina Abbonamenti e come componente di terze parti nella finestra di dialogo Crea una ricetta di immagini. Per ulteriori informazioni sul componente, vedereComponenti di tempra CIS.

Abbonarsi ai Marketplace AWS componenti

Dopo aver trovato un Marketplace AWS prodotto con componenti che desideri utilizzare nelle tue ricette, puoi abbonarti direttamente dalla console Image Builder, come segue, oppure puoi abbonarti dalla Marketplace AWS console.

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Dal pannello di navigazione, scegli Scopri i prodotti nella Marketplace AWSsezione.

3. Seleziona la scheda Components (Componenti). Questa scheda elenca tutti i Marketplace AWS prodotti che utilizzano l'opzione di consegna che include i componenti associati in Marketplace AWS.
4. Per cercare un Marketplace AWS prodotto specifico, inserisci parte del nome nella barra di ricerca. Se conosci l'editore, ma non conosci il nome esatto del prodotto o come si scrive, puoi anche `Publisher` filtrare per ottenere un elenco dei prodotti disponibili per l'editore.
5. Seleziona il prodotto a cui desideri abbonarti dall'elenco dei risultati e scegli Visualizza le opzioni di abbonamento. Questo mostra un riepilogo delle opzioni di abbonamento per il Marketplace AWS prodotto.
6. Seleziona Abbonati per abbonarti al prodotto senza uscire dalla console Image Builder. Riceverai una notifica che l'abbonamento è in corso di elaborazione. Dopo l'iscrizione, lo stato viene aggiornato a`Subscribed`.

Per ulteriori informazioni sui Marketplace AWS prodotti a cui sei attualmente abbonato, consulta i passaggi della console descritti in. Marketplace AWS abbonamenti in Image Builder

Usare un Marketplace AWS componente in una ricetta di immagini di Image Builder

È possibile utilizzare Marketplace AWS i componenti nelle ricette di immagini di Image Builder nello stesso modo in cui si utilizzano altri tipi di componenti. Per la maggior parte dei componenti associati a un prodotto di Marketplace AWS immagini, la categoria di proprietà è`Marketplace AWS`. Ad esempio, per utilizzare un componente di compilazione di un Marketplace AWS prodotto a cui sei abbonato, scegli Aggiungi componenti di compilazione e selezionalo `Marketplace AWS` dall'elenco. Si apre un pannello di selezione sul lato destro dell'interfaccia della console che elenca Marketplace AWS i componenti.

> **Note**
>
> Se state cercando il componente di protezione avanzata CIS, selezionate`Third party managed`, dall'elenco di proprietà anziché. `Marketplace AWS`

Per ulteriori informazioni su come selezionare, disporre e configurare i parametri per i componenti, consulta. Crea una nuova versione di una ricetta di immagini

Usa componenti gestiti per personalizzare l'immagine di Image Builder

I componenti gestiti vengono creati AWS, a volte in collaborazione con, un'organizzazione di terze parti, come ad esempio il Center for Internet Security (CIS). Quando utilizzi componenti gestiti nelle tue ricette di immagini o contenitori, Amazon fornisce le versioni più recenti dei componenti a cui sono applicate patch e altri aggiornamenti. Per ottenere un elenco dei componenti o per ottenere informazioni sui componenti, consultaElenca e visualizza i dettagli dei componenti.

Il seguente elenco di componenti AWS gestiti in primo piano include un componente che è disponibile per l'utilizzo quando ci si abbona a CIS con protezione avanzata AMIs tramite. Marketplace AWS

Componenti in evidenza

- Installazione dell'applicazione dei componenti gestiti dal pacchetto Distributor per le immagini Windows di Image Builder
- Componenti di tempra CIS
- Amazon ha gestito i componenti di protezione STIG per Image Builder

Installazione dell'applicazione dei componenti gestiti dal pacchetto Distributor per le immagini Windows di Image Builder

AWS Systems Manager Distributor ti aiuta a impacchettare e pubblicare software nei nodi gestiti. AWS Systems Manager È possibile creare pacchetti e pubblicare il proprio software oppure utilizzare Distributor per trovare e pubblicare pacchetti software AWS forniti dagli agenti. Per ulteriori informazioni su Systems Manager Distributor, vedere AWS Systems Manager Distributor nella Guida per l'AWS Systems Manager utente.

Componenti gestiti per Distributor

I seguenti componenti gestiti di Image Builder utilizzano AWS Systems Manager Distributor per installare pacchetti di applicazioni su istanze Windows.

- Il componente `distributor-package-windows` gestito utilizza AWS Systems Manager Distributor per installare i pacchetti di applicazioni specificati nell'istanza di creazione dell'immagine di Windows. Per configurare i parametri quando includi questo componente nella ricetta, consultaConfigura `distributor-package-windows` come componente autonomo.

- Il `aws-vss-components-windows` componente utilizza AWS Systems Manager Distributor per installare il `AwsVssComponents` pacchetto sull'istanza di creazione dell'immagine di Windows. Per configurare i parametri quando includi questo componente nella ricetta, consultaConfigura come componente autonomo `aws-vss-components-windows`.

Per ulteriori informazioni su come utilizzare i componenti gestiti nella ricetta di Image Builder, vedere Crea una nuova versione di una ricetta di immagini per le ricette di immagini o Crea una nuova versione di una ricetta contenitore per le ricette dei contenitori. Per ulteriori informazioni sul *`AwsVssComponents`* pacchetto, consulta Creare uno snapshot coerente con l'applicazione VSS nella Amazon User Guide. EC2

Prerequisiti

Prima di utilizzare i componenti Image Builder che si basano su Systems Manager Distributor per installare i pacchetti di applicazioni, è necessario assicurarsi che siano soddisfatti i seguenti prerequisiti.

- I componenti di Image Builder che utilizzano Systems Manager Distributor per installare i pacchetti di applicazioni sull'istanza necessitano dell'autorizzazione per chiamare l'API Systems Manager. Prima di utilizzare i componenti in una ricetta di Image Builder, è necessario creare la policy e il ruolo IAM che concedono l'autorizzazione. Per configurare le autorizzazioni, consulta. Configurazione delle autorizzazioni di Systems Manager Distributor

> **Note**
>
> Image Builder attualmente non supporta i pacchetti Systems Manager Distributor che riavviano l'istanza. Ad esempio, i pacchetti `AWSNVMeAWSPVDrivers`, e `AwsEnaNetworkDriver` Distributor riavviano l'istanza e quindi non sono consentiti.

Configurazione delle autorizzazioni di Systems Manager Distributor

Il `distributor-package-windows` componente e gli altri componenti che lo utilizzano, ad esempio`aws-vss-components-windows`, richiedono un'autorizzazione aggiuntiva per l'esecuzione dell'istanza di build. L'istanza di build deve essere in grado di chiamare l'API Systems Manager per avviare l'installazione di Distributor e verificare il risultato.

Segui queste procedure AWS Management Console per creare una policy e un ruolo IAM personalizzati che concedano l'autorizzazione ai componenti di Image Builder per installare i pacchetti Systems Manager Distributor dall'istanza di build.

Fase 1: Creare una policy

Crea una policy IAM per le autorizzazioni dei distributori.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy e quindi Crea policy.
3. Nella pagina Crea policy, scegli la scheda JSON, quindi sostituisci il contenuto predefinito con la seguente policy JSON, sostituendo la partizione, la regione e l'ID dell'account se necessario o utilizzando caratteri jolly.

```
{
 "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. Scegliere Review policy (Esamina policy).

5. Per Nome, immettere un nome per identificare la policy, ad esempio *InvokeDistributor* o un altro nome a scelta.
6. (Facoltativo) Per Descrizione, immettere la descrizione dello scopo della policy.
7. Seleziona Create Policy (Crea policy).

Fase 2: Creare un ruolo

Crea un ruolo IAM per le autorizzazioni del distributore.

1. Dal pannello di navigazione della console IAM, scegli Ruoli, quindi scegli Crea ruolo.
2. In Select type of trusted entity (Seleziona tipo di entità attendibile), scegli Servizio AWS.
3. Immediatamente in Scegli il servizio che utilizzerà questo ruolo, scegli EC2, quindi scegli Avanti: Autorizzazioni.
4. In Seleziona il tuo caso d'uso, scegli EC2, quindi scegli Avanti: Autorizzazioni.
5. Nell'elenco delle politiche, seleziona la casella di controllo accanto ad Amazon SSMManaged InstanceCore. (Digitare `SSM` nella casella di ricerca per restringere l'elenco).
6. In questo elenco di politiche, seleziona la casella accanto a EC2InstanceProfileForImageBuilder. (Digitare `ImageBuilder` nella casella di ricerca per restringere l'elenco).
7. Scegliere Next: Tags (Successivo: Tag).
8. (Facoltativo) Aggiungi una o più coppie di tag chiave-valore per organizzare, tracciare o controllare l'accesso per questo ruolo, quindi scegli Avanti: revisione.
9. Per Nome ruolo, immettere un nome per il nuovo ruolo, ad esempio *InvokeDistributor* o un altro nome a scelta.
10. (Facoltativo) Per Role description (Descrizione ruolo), sostituire il testo predefinito con una descrizione dello scopo del ruolo.
11. Selezionare Create role (Crea ruolo). Il sistema visualizza di nuovo la pagina Roles (Ruoli).

Fase 3: Allega la policy al ruolo

Il passaggio finale per configurare le autorizzazioni di distributore consiste nell'associare la policy IAM al ruolo IAM.

1. Dalla pagina Ruoli della console IAM, scegli il ruolo che hai appena creato. Viene visualizzata la pagina Riepilogo del ruolo.

2. Scegliere Collega policy.
3. Cerca la policy che hai creato nella procedura precedente e seleziona la casella di controllo accanto al nome.
4. Scegli Collega policy.

Utilizzate questo ruolo nella risorsa Image Builder Infrastructure Configuration per qualsiasi immagine che include componenti che utilizzano Systems Manager Distributor. Per ulteriori informazioni, consulta Creazione di una configurazione dell'infrastruttura.

Configura **`distributor-package-windows`** come componente autonomo

Per utilizzare il `distributor-package-windows` componente in una ricetta, imposta i seguenti parametri che configurano il pacchetto da installare.

> **Note**
>
> Prima di utilizzare il `distributor-package-windows` componente in una ricetta, è necessario assicurarsi che tutti i requisiti Prerequisiti siano soddisfatti.

- Azione (obbligatoria): specifica se installare o disinstallare il pacchetto. I valori validi includono `Install` e `Uninstall`. Il valore predefinito è. `Install`
- PackageName(Obbligatorio): il nome del pacchetto Distributor da installare o disinstallare. Per un elenco di nomi di pacchetto validi, vedereTrova i pacchetti Distributor.
- PackageVersion(Facoltativo): la versione del pacchetto Distributor da installare. PackageVersion il valore predefinito è la versione consigliata.
- AdditionalArguments(Facoltativo): una stringa JSON che contiene i parametri aggiuntivi da fornire allo script per installare, disinstallare o aggiornare un pacchetto. Per ulteriori informazioni, consulta AdditionalArguments nella sezione AWS:ConfigurePackage Inputs della pagina di riferimento del plug-in del documento Systems Manager Command.

Configura come componente autonomo **`aws-vss-components-windows`**

Quando si utilizza il `aws-vss-components-windows` componente in una ricetta, è possibile facoltativamente impostare il `PackageVersion` parametro per utilizzare una versione specifica del

`AwsVssComponents` pacchetto. Quando si omette questo parametro, per impostazione predefinita il componente utilizza la versione consigliata del pacchetto. `AwsVssComponents`

> **Note**
>
> Prima di utilizzare il `aws-vss-components-windows` componente in una ricetta, è necessario assicurarsi che tutti i parametri Prerequisiti siano soddisfatti.

Trova i pacchetti Distributor

Amazon e terze parti forniscono pacchetti pubblici che puoi installare con Systems Manager Distributor.

Per visualizzare i pacchetti disponibili in AWS Management Console, accedi alla AWS Systems Manager console e scegli Distributor dal pannello di navigazione. La pagina Distributore mostra tutti i pacchetti disponibili. Per ulteriori informazioni su come elencare i pacchetti disponibili con AWS CLI, consulta Visualizza i pacchetti (riga di comando) nella Guida per l'AWS Systems Manager utente.

È inoltre possibile creare pacchetti Systems Manager Distributor privati. Per ulteriori informazioni, consulta Creare un pacchetto nella Guida per l'AWS Systems Manager utente.

Componenti di tempra CIS

Il Center for Internet Security (CIS) è un'organizzazione senza scopo di lucro gestita dalla comunità. I suoi esperti di sicurezza informatica collaborano per sviluppare linee guida sulla sicurezza IT che salvaguardino le organizzazioni pubbliche e private dalle minacce informatiche. La loro serie di best practice riconosciute a livello globale, note come CIS Benchmarks, aiuta le organizzazioni IT di tutto il mondo a configurare in modo sicuro i propri sistemi. Per articoli di tendenza, post di blog, podcast, webinar e white paper, consultate CIS Insights sul sito Web del Center for Internet Security.

CIS Benchmarks

Il CIS crea e mantiene una serie di linee guida di configurazione, note come CIS Benchmark, che forniscono le migliori pratiche di configurazione per tecnologie specifiche, tra cui sistemi operativi, piattaforme cloud, applicazioni, database e altro ancora. I benchmark CIS sono riconosciuti come standard di settore da organizzazioni e standard come PCI DSS, HIPAA, DoD Cloud Computing SRG, FISMA, DFARS e FEDRAMP. Per ulteriori informazioni, consulta CIS Benchmarks sul sito Web Center for Internet Security.

Componenti di indurimento CIS

Quando sottoscrivi una CIS Hardened Image in Marketplace AWS, hai anche accesso al componente di protezione avanzata associato che esegue uno script per applicare le linee guida CIS Benchmarks Level 1 per la tua configurazione. L'organizzazione CIS possiede e gestisce i componenti di protezione CIS per garantire che riflettano le linee guida più recenti.

> **Note**
>
> I componenti di protezione avanzata CIS non seguono le regole standard di ordinamento dei componenti nelle ricette di Image Builder. I componenti di protezione avanzata CIS funzionano sempre per ultimi per garantire che i test di benchmark vengano eseguiti rispetto all'immagine di output.

Amazon ha gestito i componenti di protezione STIG per Image Builder

Le guide tecniche di implementazione della sicurezza (STIGs) sono gli standard di rafforzamento della configurazione creati dalla Defense Information Systems Agency (DISA) per proteggere i sistemi informativi e il software. Per la conformità dei propri sistemi agli standard STIG, è necessario installare, configurare e testare una serie di impostazioni di sicurezza.

Image Builder fornisce componenti di protezione STIG per aiutarvi a creare in modo più efficiente immagini conformi agli standard STIG di base. Questi componenti STIG analizzano eventuali configurazioni errate ed eseguono uno script di correzione. Non sono previsti costi aggiuntivi per l'utilizzo di componenti conformi a STIG.

> **Important**
>
> Con poche eccezioni, i componenti di protezione STIG non installano pacchetti di terze parti. Se sull'istanza sono già installati pacchetti di terze parti e se ci sono pacchetti correlati STIGs supportati da Image Builder per quel pacchetto, il componente di protezione avanzata li applica.

Questa pagina elenca tutto ciò STIGs che Image Builder supporta e che viene applicato alle EC2 istanze che Image Builder avvia quando si crea e si testa una nuova immagine. Se si desidera applicare impostazioni STIG aggiuntive all'immagine, è possibile creare un componente

personalizzato per configurarla. Per ulteriori informazioni sui componenti personalizzati e su come crearli, consultaUsa i componenti per personalizzare l'immagine di Image Builder.

Quando create un'immagine, i componenti di protezione STIG registrano se i supporti supportati STIGs vengono applicati o ignorati. Ti consigliamo di esaminare i log di Image Builder per le tue immagini che utilizzano componenti di protezione STIG. Per ulteriori informazioni su come accedere e rivedere i log di Image Builder, vedere. Risolvi i problemi relativi alle build della pipeline

Livelli di conformità

- Alta (Categoria I)

  Il rischio più grave. Include qualsiasi vulnerabilità che può comportare la perdita di riservatezza, disponibilità o integrità.

- Medio (categoria II)

  Include qualsiasi vulnerabilità che può comportare la perdita di riservatezza, disponibilità o integrità, ma i rischi possono essere mitigati.

- Basso (categoria III)

  Eventuali vulnerabilità che degradano le misure per la protezione contro la perdita di riservatezza, disponibilità o integrità.

Argomenti

- Componenti di protezione avanzata di Windows STIG
- Registro della cronologia delle versioni STIG per Windows
- Componenti di indurimento STIG per Linux
- Registro della cronologia delle versioni STIG per Linux
- Componente del validatore di conformità SCAP

Componenti di protezione avanzata di Windows STIG

AWSTOE I componenti di protezione avanzata di Windows STIG sono progettati per server autonomi e applicano i criteri di gruppo locali. I componenti di protezione avanzata conformi a STIG vengono installati InstallRoot dal Dipartimento della Difesa (DoD) sull'infrastruttura Windows per scaricare, installare e aggiornare i certificati DoD. Inoltre rimuovono i certificati non necessari per mantenere

la conformità STIG. Attualmente, le linee di base STIG sono supportate per le seguenti versioni di Windows Server: 2012 R2, 2016, 2019 e 2022.

Questa sezione elenca le impostazioni correnti per ciascuno dei componenti di protezione avanzata di Windows STIG, seguite da un registro della cronologia delle versioni.

STIG-Build-Windows-Low versione 2025.1.x

L'elenco seguente contiene le impostazioni STIG che il componente di protezione avanzata applica all'infrastruttura. Se un'impostazione supportata non è applicabile all'infrastruttura, il componente di protezione avanzata ignora tale impostazione e prosegue. Ad esempio, alcune impostazioni STIG potrebbero non essere applicabili ai server autonomi. Le politiche specifiche dell'organizzazione possono anche influire sulle impostazioni applicate dal componente di protezione avanzata, ad esempio l'obbligo per gli amministratori di rivedere le impostazioni dei documenti.

Per un elenco completo di Windows, consulta la Document STIGs Library. STIGs Per informazioni su come visualizzare l'elenco completo, consulta Strumenti di visualizzazione STIG.

- Windows Server 2022 STIG versione 2 versione 2

  V-254335, V-254336, V-254337, V-254338, V-254351, V-254357, V-254363 e V-254481
- Windows Server 2019 STIG versione 3 versione 2

  V-205691, V-205819, V-205858, V-205859, V-205860, V-205870, V-205871 e V-205923
- Windows Server 2016 STIG versione 2 versione 9

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

  V-225537, V-225536, V-225526, V-225525, V-225514, V-225511, V-225490, V-225489, V-225488, V-225487, V-225485, V-225484, V-225483, V-225482, V-225481, V-225480, V-225479, V-225476, V-225473, V-225468, V-225462, V-225460, V-225459, V-225412, V-225394, V-225392, V-225376, V-225363, V-225362, V-225360, V-225359, V-225358, V-225357, V-225355, V-225343, V-225342, V-225336, V-225335, V-225334, V-225333, V-225332, V-225331, V-225330, V-225328, V-225327, V-225324, V-225319, V-225318 e V-225250
- Microsoft.NET Framework 4.0 STIG Versione 2 Release 2

  Non vengono applicate impostazioni STIG alle vulnerabilità di Microsoft .NET Framework per la categoria III.

- Windows Firewall STIG versione 2 versione 2

  V-241994, V-241995, V-241996, V-241999, V-242000, V-242001, V-242006, V-242007 e V-242008
- Internet Explorer 11 STIG versione 2 versione 5

  V-223016, V-223056 e V-223078
- Microsoft Edge STIG versione 2 versione 2 (solo Windows Server 2022)

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

STIG-Build-Windows-Medium versione 2025.1.x

L'elenco seguente contiene le impostazioni STIG che il componente di protezione avanzata applica all'infrastruttura. Se un'impostazione supportata non è applicabile all'infrastruttura, il componente di protezione avanzata ignora tale impostazione e prosegue. Ad esempio, alcune impostazioni STIG potrebbero non essere applicabili ai server autonomi. Le politiche specifiche dell'organizzazione possono anche influire sulle impostazioni applicate dal componente di protezione avanzata, ad esempio l'obbligo per gli amministratori di rivedere le impostazioni dei documenti.

Per un elenco completo di Windows, consulta la Document STIGs Library. STIGs Per informazioni su come visualizzare l'elenco completo, consulta Strumenti di visualizzazione STIG.

> **Note**
>
> I componenti di STIG-Build-Windows-Medium protezione avanzata includono tutte le impostazioni STIG elencate che AWSTOE si applicano ai componenti di STIG-Build-Windows-Low protezione avanzata, oltre alle impostazioni STIG elencate specificamente per le vulnerabilità di categoria II.

- Windows Server 2022 STIG versione 2 versione 2

  Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria III (bassa), oltre a:

  V-254247, V-254265, V-254269, V-254270, V-254271, V-254272, V-254273, V-254274, V-254276, V-254277, V-254278, V-254285, V-254286, V-254287, V-254288, V-254289, V-254290, V-254291, V-254292, V-254300, V-254301, V-254302, V-254303, V-254304, V-254305, V-254306, V-254307,

V-254308, V-254309, V-254310, V-254311, V-254312, V-254313, V-254314, V-254315, V-254316, V-254317, V-254318, V-254319, V-254320, V-254321, V-254322, V-254323, V-254324, V-254325, V-254326, V-254327, V-254328, V-254329, V-254341, V-254342, V-254342, V-254344, V-254345, V-254346, V-254347, V-254348, V-254349, V-254350, V-254355, V-254356, V-254356, V-254358, V-254359, V-254360, V-254361, V-254362, V-254364, V-254366, V-254367, V-254368, V-254369, V-254370, V-254371, V-254372, V-254373, V-254375, V-254376, V-254377, V-254379, V-254380, V-254382, V-254383, V-254384, V-254431, V-254432, V-254433, V-254434, V-254435, V-254436, V-254438, V-254439, V-254442, V-254443, V-254444, V-254445, V-254449, V-254450, V-254451, V-254452, V-254453, V-254454, V-254455, V-254456, V-254459, V-254460, V-254461, V-254462, V-254463, V-254464, V-254468, V-254470, V-254471, V-254472, V-254473, V-254476, V-254477, V-254478, V-254479, V-254480, V-254482, V-254483, V-254484, V-254485, V-254486, V-254487, V-254488, V-254489, V-254490, V-254493, V-254494, V-254495, V-254497, V-254499, V-254501, V-254502, V-254503, V-254504, V-254505, V-254507, V-254508, V-254509, V-254510, V-254511 e V-254512

- Windows Server 2019 STIG versione 3 versione 2

  Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria III (bassa), oltre a:

  V-205625, V-205626, V-205627, V-205629, V-205630, V-205633, V-205634, V-205635, V-205636, V-205637, V-205638, V-205639, V-205643, V-205644, V-205648, V-205649, V-205650, V-205651, V-2056651, V-2056652, V-205655, V-205656, V-205659, V-205660, V-205662, V-205671, V-205672, V-205673, V-205675, V-205676, V-205678, V-205679, V-205680, V-205681, V-205682, V-205683, V-205684, V-205685, V-205686, V-205686, V-205686 V-205688, V-205689, V-205690, V-205692, V-205693, V-205694, V-205697, V-205698, V-205708, V-205709, V-205712, V-205714, V-205716, V-205717, V-205718, V-205719, V-205720, V-205720, V-205722, V-205729, V-205730, V-205733, V-205747, V-205751, V-205752, V-205754, V-205756, V-205758, V-205759, V-205760, V-205761, V-205762, V-205764, V-205765, V-205766, V-205767, V-205768, V-205769, V-205770, V-205771, V-205772, V-205773, V-205774, V-205775, V-205776, V-205777, V-205778, V-205779, V-205780, V-205781, V-205782, V-205783, V-205784, V-205795, V-205796, V-205797, V-792058, V-205788 V-205808, V-205809, V-205810, V-205811, V-205812, V-205813, V-205814, V-205815, V-205816, V-205817, V-205821, V-205822, V-205823, V-205824, V-205825, V-205826, V-205827, V-205828 V-205830, V-205832, V-205833, V-205834, V-205835, V-205836, V-205837, V-205838, V-205839, V-205840, V-205841, V-205842, V-205861, V-205863, V-205865, V-205866, V-205867, V-205868, V-205869, V-205872, V-205873, V-205874, V-205911, V-205912, V-205915, V-205916, V-205917, V-205918, V-205920, V-205921, V-205922, V-205924, V-205925, V-236001 e V-257503

- Windows Server 2016 STIG versione 2 versione 9

  Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria III (bassa), oltre a:

  V-224850, V-224852, V-224853, V-224854, V-224855, V-224856, V-224857, V-224858, V-224859, V-224866, V-224867, V-224868, V-224869, V-224870, V-224871, V-224872, V-224873, V-224881, V-224882, V-224883, V-224884, V-224885, V-224886, V-224887, V-224888, V-224889, V-224890, V-224891, V-224892, V-224893, V-224894, V-224895, V-224896, V-224897, V-224898, V-224899, V-224900, V-224901, V-224902, V-224903, V-224904, V-224905, V-224906, V-224907, V-224908, V-224909, V-224910, V-224911, V-224912, V-224913, V-224914, V-224915, V-224920, V-224922, V-224924, V-224925, V-224926, V-224927, V-224928, V-224929, V-224930, V-224935, V-224936, V-224937, V-224938, V-224939, V-224940, V-224941, V-224943, V-224944, V-224945, V-224946, V-224947, V-224948, V-224949, V-224951, V-224952, V-224953, V-224955, V-224956, V-224957, V-224959, V-224960, V-224962, V-224963, V-225010, V-225013, V-225014, V-225015, V-225016, V-225017, V-225018, V-225019, V-225021, V-225022, V-225023, V-225024, V-225028, V-225029, V-225030, V-225031, V-225032, V-225033, V-225034, V-225035, V-225038, V-225039, V-225040, V-225041, V-225042, V-225043, V-225047, V-225049, V-225050, V-225051, V-225052, V-225055, V-225056, V-225057, V-225058, V-225059, V-225061, V-225062, V-225063, V-225064, V-225065, V-225066, V-225067, V-225068, V-225069, V-225072, V-225073, V-225074, V-225076, V-225078, V-225080, V-225081, V-225082, V-225083, V-225084, V-225086, V-225087, V-225088, V-225089, V-225092, V-225093, V-236000 e V-257502

- Windows Server 2012 R2 MS STIG versione 3 versione 5

  Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria III (bassa), oltre a:

  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-225555, V-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 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-225461 V-225458, V-225457, V-225456, V-225455, V-225454, V-225453, V-225452, V-225448, V-225443, V-225442, V-225441, V-225415, V-225414, V-225413, V-225411, V-225410, V-225409, V-225408, V-225407

V-225406, V-225405, V-225404, V-225402, V-225401, V-225400, V-225398, V-225397, V-225395, V-225393, V-225391, V-225389, V-225386, V-225385, V-225384, V-225383, V-225382, V-225381, V-225380, V-225379, V-225378, V-225377, V-225375, V-225374, V-225373, V-225372, V-225371, V-225370, V-225369, V-225368, V-225367, V-225356, V-225353, V-225352, V-225351, V-225350, V-225349, V-225348, V-225348, V-225348, V-225352 V-225346, V-225345, V-225344, V-225341, V-225340, V-225339, V-225338, V-225337, V-225329, V-225326, V-225325, V-225317, V-225316, V-225315, V-225314, V-225305, V-225304, V-225303, V-225302, V-225301, V-225300, V-225299, V-225298, V-225297, V-225296, V-225295, V-225294, V-225293, V-225292, V-225291, V-225290, V-225289, V-225288, V-225287, V-225286, V-225285, V-225284, V-225283, V-225282, V-225281, V-225280, V-225279, V-225278, V-225277, V-225276, V-225275, V-225273, V-225272, V-225271, V-225270, V-225269, V-225268, V-225267, V-225266, V-225265, V-225264, V-225263, V-225261, V-225260, V-225259 e V-225239

- Microsoft.NET Framework 4.0 STIG Versione 2 Release 2

  Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria III (basse), oltre a V-225238

- Windows Firewall STIG versione 2 versione 2

  Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria III (bassa), oltre a:

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

- Internet Explorer 11 STIG versione 2 versione 5

  Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria III (bassa), oltre a:

  V-223015, V-223017, V-223018, V-223019, V-223020, V-223021, V-223022, V-223023, V-223024, V-223025, V-223026, V-223027, V-223028, V-223029, V-223030, V-223031, V-223032, V-223033, V-223034, V-223035, V-223036, V-223037, V-223038, V-223039, V-223040, V-223041, V-223042, V-223043, V-223044, V-223045, V-223046, V-223048, V-223049, V-223050, V-223051, V-223052, V-223053, V-223054, V-223055, V-223057, V-223058, V-223059, V-223060, V-223061, V-223062, V-223063, V-223064, V-223065, V-223066, V-223067, V-223068, V-223069, V-223070, V-223071, V-223072, V-223073, V-223074, V-223075, V-223076, V-223077, V-223079, V-223080, V-223081, V-223082, V-223083, V-223084, V-223085, V-223086, V-223087, V-223088, V-223089, V-223090, V-223091, V-223092, V-223093, V-223094, V-223095, V-223096, V-223097, V-223098, V-223099, V-223100, V-223101, V-223102, V-223103, V-223104, V-223105, V-223106, V-223107, V-223108,

V-223109, V-223110, V-223111, V-223112, V-223113, V-223114, V-223115, V-223116, V-223117, V-223118, V-223119, V-223120, V-223121, V-223122, V-223123, V-223124, V-223125, V-223126, V-223127, V-223128, V-223129, V-223130, V-223131, V-223132, V-223133, V-223134, V-223135, V-223136, V-223137, V-223138, V-223139, V-223140, V-223141, V-223142, V-223143, V-223144, V-223145, V-223146, V-223147, V-223148, V-223149, V-250540 e V-250541

- Microsoft Edge STIG versione 2 versione 2 (solo Windows Server 2022)

  Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria III (bassa), oltre a:

  V-235720, V-235721, V-235723, V-235724, V-235725, V-235726, V-235728, V-235729, V-235730, V-235732, V-235733, V-235734, V-235735, V-235736, V-235737, V-235738, V-235739, V-235740, V-235741, V-235742, V-235743, V-235744, V-235745, V-235746, V-235747, V-235748, V-235749, V-235750, V-235754, V-235756, V-235760, V-235761, V-235763, V-235764, V-235766, V-235767, V-235768, V-235769, V-235770, V-235771, V-235772, V-235773, V-235774 e V-246736

- Microsoft Defender STIG Versione 2 Release 4

  Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria III (bassa), oltre a:

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

STIG-Build-Windows-High versione 2025.1.x

L'elenco seguente contiene le impostazioni STIG che il componente di protezione avanzata applica all'infrastruttura. Se un'impostazione supportata non è applicabile all'infrastruttura, il componente di protezione avanzata ignora tale impostazione e prosegue. Ad esempio, alcune impostazioni STIG potrebbero non essere applicabili ai server autonomi. Le politiche specifiche dell'organizzazione possono anche influire sulle impostazioni applicate dal componente di protezione avanzata, ad esempio l'obbligo per gli amministratori di rivedere le impostazioni dei documenti.

Per un elenco completo di Windows, consulta la Document STIGs Library. STIGs Per informazioni su come visualizzare l'elenco completo, consulta Strumenti di visualizzazione STIG.

> **Note**
>
> I componenti di STIG-Build-Windows-High protezione avanzata includono tutte le impostazioni STIG elencate STIG-Build-Windows-Low e AWSTOE valide per i componenti di STIG-Build-Windows-Medium protezione avanzata, oltre alle impostazioni STIG elencate specificamente per le vulnerabilità di categoria I.

- Windows Server 2022 STIG versione 2 versione 2

  Include tutte le impostazioni STIG supportate applicate dal componente di protezione avanzata per le vulnerabilità di categoria II e III (media e bassa), oltre a:

  V-254293, V-254352, V-254353, V-254354, V-254374, V-254378, V-254381, V-254446, V-254465, V-254466, V-254467, V-254469, V-254474, V-254475 e V-254500

- Windows Server 2019 STIG versione 3 versione 2

  Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria II e III (media e bassa), oltre a:

  V-205653, V-205654, V-205711, V-205713, V-205724, V-205725, V-205757, V-205802, V-205804, V-205805, V-205806, V-205849, V-205908, V-205913, V-205914 e V-205919

- Windows Server 2016 STIG versione 2 versione 9

  Include tutte le impostazioni STIG supportate applicate dal componente di protezione avanzata per le vulnerabilità di categoria II e III (media e bassa), oltre a:

  V-224874, V-224932, V-224933, V-224934, V-224954, V-224958, V-224961, V-225025, V-225044, V-225045, V-225046, V-225048, V-225053, V-225054 e V-225079

- Windows Server 2012 R2 MS STIG versione 3 versione 5

  Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria II e III (media e bassa), oltre a:

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

- Microsoft.NET Framework 4.0 STIG Versione 2 Release 2

Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità delle categorie II e III (medie e basse) per Microsoft.NET Framework. Non si applicano impostazioni STIG aggiuntive per le vulnerabilità di categoria I.

- Windows Firewall STIG versione 2 versione 2

  Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria II e III (media e bassa), oltre a:

  V-241992, V-241997 e V-242002

- Internet Explorer 11 STIG versione 2 versione 5

  V-252910

- Microsoft Edge STIG versione 2 versione 2 (solo Windows Server 2022)

  Include tutte le impostazioni STIG supportate applicate dal componente di protezione avanzata per le vulnerabilità di categoria II e III (medie e basse), oltre a:

  V-235758 e V-235759

- Microsoft Defender STIG Versione 2 Release 4

  Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria II e III (media e bassa), oltre a:

  V-213426, V-213426, V-213452, V-213452, V-213452, V-213453, V-213453 e V-213453

Registro della cronologia delle versioni STIG per Windows

Questa sezione registra la cronologia delle versioni dei componenti di protezione avanzata di Windows per gli aggiornamenti trimestrali STIG. Per visualizzare le modifiche e le versioni pubblicate per un trimestre, scegliete il titolo per espandere le informazioni.

Modifiche al primo trimestre 2025 - 05/04/2025:

STIGS aggiornato per Internet Explorer 11 STIG versione 2 Release 5 per tutti i componenti STIG per la versione del primo trimestre 2025.

- STIG-Build-Windows-Low versione 2025.1.x
- STIG-Build-Windows-Medium versione 2025.1.x

- STIG-Build-Windows-High versione 2025.1.x

Modifiche al quarto trimestre 2024 - 04/02/2025:

Versioni STIG aggiornate e STIGS applicati per la versione del quarto trimestre 2024 come segue:

STIG-Build-Windows-Low versione 2024.4.0

- Windows Server 2022 STIG versione 2 versione 2
- Windows Server 2019 STIG versione 3 versione 2
- Windows Server 2016 STIG versione 2 versione 9
- Windows Server 2012 R2 MS STIG versione 3 rilascio 5
- Microsoft .NET Framework 4.0 STIG versione 2 rilascio 2
- Windows Firewall STIG versione 2 versione 2
- Internet Explorer 11 STIG versione 2 versione 5
- Microsoft Edge STIG versione 2 versione 2 (solo Windows Server 2022)

STIG-Build-Windows-Medium versione 2024.4.0

- Windows Server 2022 STIG versione 2 versione 2
- Windows Server 2019 STIG versione 3 versione 2
- Windows Server 2016 STIG versione 2 versione 9
- Windows Server 2012 R2 MS STIG versione 3 rilascio 5
- Microsoft .NET Framework 4.0 STIG versione 2 rilascio 2
- Windows Firewall STIG versione 2 versione 2
- Internet Explorer 11 STIG versione 2 versione 5
- Microsoft Edge STIG versione 2 versione 2 (solo Windows Server 2022)
- Defender STIG versione 2 versione 4

STIG-Build-Windows-High versione 2024.4.0

- Windows Server 2022 STIG versione 2 versione 2
- Windows Server 2019 STIG versione 3 versione 2

- Windows Server 2016 STIG versione 2 versione 9
- Windows Server 2012 R2 MS STIG versione 3 rilascio 5
- Microsoft .NET Framework 4.0 STIG versione 2 rilascio 2
- Windows Firewall STIG versione 2 versione 2
- Internet Explorer 11 STIG versione 2 versione 5
- Microsoft Edge STIG versione 2 versione 2 (solo Windows Server 2022)
- Defender STIG versione 2 versione 4

Modifiche al terzo trimestre 2024 - 10/04/2023 (nessuna modifica):

Non sono state apportate modifiche al componente Windows STIGS per la versione del terzo trimestre 2024.

Modifiche al secondo trimestre 2024 - 05/10/2024 (nessuna modifica):

Non sono state apportate modifiche al componente Windows STIGS per la versione del secondo trimestre 2024.

Modifiche al 1° trimestre 2024 - 02/06/2024 (nessuna modifica):

Non sono state apportate modifiche al componente Windows STIGS per la versione del primo trimestre 2024.

Modifiche al quarto trimestre 2023 - 12/04/2023 (nessuna modifica):

Non sono state apportate modifiche al componente Windows STIGS per la versione del quarto trimestre 2023.

Modifiche al terzo trimestre 2023 - 10/04/2023 (nessuna modifica):

Non sono state apportate modifiche al componente Windows STIGS per la versione del terzo trimestre 2023.

Modifiche al secondo trimestre 2023 - 05/03/2023 (nessuna modifica):

Non sono state apportate modifiche al componente Windows STIGS per la versione del secondo trimestre 2023.

Modifiche al 1° trimestre 2023 - 27/03/2023 (nessuna modifica):

Non sono state apportate modifiche al componente Windows STIGS per la versione del primo trimestre 2023.

Modifiche al quarto trimestre 2022 - 01/02/2023:

Versioni STIG aggiornate e STIGS applicati per la versione del quarto trimestre 2022 come segue:

STIG-Build-Windows-Low versione 2022.4.x

- Windows Server 2022 STIG versione 1 rilascio 1
- Windows Server 2019 STIG versione 2 rilascio 5
- Windows Server 2016 STIG versione 2 rilascio 5
- Windows Server 2012 R2 MS STIG versione 3 rilascio 5
- Microsoft .NET Framework 4.0 STIG versione 2 rilascio 2
- Windows Firewall STIG versione 2 rilascio 1
- Internet Explorer 11 STIG versione 2 rilascio 3
- Microsoft Edge STIG versione 1 versione 6 (solo Windows Server 2022)

STIG-Build-Windows-Medium versione 2022.4.x

- Windows Server 2022 STIG versione 1 rilascio 1
- Windows Server 2019 STIG versione 2 rilascio 5
- Windows Server 2016 STIG versione 2 rilascio 5
- Windows Server 2012 R2 MS STIG versione 3 rilascio 5
- Microsoft .NET Framework 4.0 STIG versione 2 rilascio 2
- Windows Firewall STIG versione 2 rilascio 1
- Internet Explorer 11 STIG versione 2 rilascio 3
- Microsoft Edge STIG versione 1 versione 6 (solo Windows Server 2022)
- Defender STIG versione 2 versione 4 (solo Windows Server 2022)

STIG-Build-Windows-High versione 2022.4.x

- Windows Server 2022 STIG versione 1 rilascio 1

- Windows Server 2019 STIG versione 2 rilascio 5
- Windows Server 2016 STIG versione 2 rilascio 5
- Windows Server 2012 R2 MS STIG versione 3 rilascio 5
- Microsoft .NET Framework 4.0 STIG versione 2 rilascio 2
- Windows Firewall STIG versione 2 rilascio 1
- Internet Explorer 11 STIG versione 2 rilascio 3
- Microsoft Edge STIG versione 1 versione 6 (solo Windows Server 2022)
- Defender STIG versione 2 versione 4 (solo Windows Server 2022)

Modifiche al terzo trimestre 2022 - 30/09/2022 (nessuna modifica):

Non sono state apportate modifiche al componente Windows STIGS per la versione del terzo trimestre 2022.

Modifiche al secondo trimestre 2022 - 08/02/2022:

Versioni STIG aggiornate e STIGS applicati per la versione del secondo trimestre 2022.

STIG-Build-Windows-Low versione 1.5.x

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

STIG-Build-Windows-Medium versione 1.5.x

- Windows Server 2019 STIG versione 2 versione 4
- Windows Server 2016 STIG versione 2 versione 4
- Windows Server 2012 R2 MS STIG versione 3 versione 3
- Microsoft.NET Framework 4.0 STIG Versione 2 Release 1

- Windows Firewall STIG versione 2 rilascio 1
- Internet Explorer 11 STIG versione 1 versione 19

STIG-Build-Windows-High versione 1.5.x

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

Modifiche al 1° trimestre 2022 - 08/02/2022 (nessuna modifica):

Non sono state apportate modifiche al componente Windows STIGS per la versione del primo trimestre 2022.

Modifiche al quarto trimestre 2021 - 20/12/2021:

Versioni STIG aggiornate e STIGS applicato per la versione del quarto trimestre 2021.

STIG-Build-Windows-Low versione 1.5.x

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

STIG-Build-Windows-Medium versione 1.5.x

- Windows Server 2019 STIG versione 2 versione 3
- Windows Server 2016 STIG versione 2 versione 3

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

STIG-Build-Windows-High versione 1.5.x

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

Modifiche al terzo trimestre 2021 - 30/09/2021:

Versioni STIG aggiornate e STIGS applicato per la versione del terzo trimestre 2021.

STIG-Build-Windows-Low versione 1.4.x

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

STIG-Build-Windows-Medium versione 1.4.x

- Windows Server 2019 STIG versione 2 versione 2
- Windows Server 2016 STIG versione 2 versione 2
- Windows Server 2012 R2 MS STIG versione 3 versione 2
- Microsoft.NET Framework 4.0 STIG Versione 2 Release 1

- Windows Firewall STIG versione 1 versione 7
- Internet Explorer 11 STIG versione 1 versione 19

STIG-Build-Windows-High versione 1.4.x

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

Componenti di indurimento STIG per Linux

Questa sezione contiene informazioni sui componenti di protezione avanzata STIG di Linux, seguite da un registro della cronologia delle versioni. Se la distribuzione Linux non dispone di impostazioni STIG proprie, il componente di protezione avanzata applica le impostazioni RHEL.

I componenti Linux dispongono di parametri di input opzionali che consentono di personalizzare i seguenti comportamenti per l'istanza Linux.

- InstallPackages (stringa) Se il valore è`No`, il componente non installa alcun pacchetto software aggiuntivo. Se il valore è`Yes`, il componente installa pacchetti software aggiuntivi necessari per la massima conformità. Il valore predefinito è `No`.
- SetDoDConsentBanner (stringa) Se il valore è`No`, il banner di consenso DoD non viene visualizzato quando si collega a un'istanza su cui è installato uno dei componenti STIG Linux. Se il valore è`Yes`, il banner di consenso DoD viene visualizzato prima dell'accesso quando ci si collega a un'istanza su cui è installato uno dei componenti STIG Linux. È necessario confermare il banner prima di poter accedere. Il valore predefinito è `No`.

  Per un esempio del banner di consenso, consulta l'Avviso di esclusione della responsabilità sulla privacy e sul consenso del Dipartimento della Difesa che appare quando accedi al sito Web di DLA Document Services.

Il componente di protezione avanzata applica le impostazioni STIG supportate all'infrastruttura basata sulla distribuzione Linux, come segue:

Impostazioni STIG di Red Hat Enterprise Linux (RHEL) 7

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

Impostazioni RHEL 8 STIG

- RHEL 8
- CentOS 8
- Amazon Linux 2023 (AL2023)

Impostazioni RHEL 9 STIG

- RHEL 9
- CentOS Stream 9

STIG-Build-Linux-Low versione 2025.1.x

L'elenco seguente contiene le impostazioni STIG che il componente di protezione avanzata applica all'infrastruttura. Se un'impostazione supportata non è applicabile all'infrastruttura, il componente di protezione avanzata ignora tale impostazione e prosegue. Ad esempio, alcune impostazioni STIG potrebbero non essere applicabili ai server autonomi. Le politiche specifiche dell'organizzazione possono anche influire sulle impostazioni applicate dal componente di protezione avanzata, ad esempio l'obbligo per gli amministratori di rivedere le impostazioni dei documenti.

Per un elenco completo, consultate la Document Library. STIGs Per informazioni su come visualizzare l'elenco completo, consulta Strumenti di visualizzazione STIG.

RHEL 7 STIG versione 3 Release 15

- RHEL 7/CentOS 7/ AL2

  V-204452, V-204576 e V-204605

RHEL 8 STIG versione 2 versione 2

- RHEL 8/CentOS 8/AL 2023

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

RHEL 9 STIG versione 2 Release 3

- RHEL 9/CentOS Stream 9

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

Ubuntu 18.04 STIG versione 2 versione 15

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

Ubuntu 20.04 STIG versione 2 versione 2

V-238203, V-238202, V-238234, V-238235, V-238237, V-238323, V-238373, V-238221, V-238222, V-238223, V-238224, V-238226, V-238362, V-238357 e V-238308

Ubuntu 22.04 STIG versione 2 versione 3

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

Ubuntu 24.04 STIG versione 1 versione 1

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

STIG-Build-Linux-Medium versione 2025.1.x

L'elenco seguente contiene le impostazioni STIG che il componente di protezione avanzata applica all'infrastruttura. Se un'impostazione supportata non è applicabile all'infrastruttura, il componente di protezione avanzata ignora tale impostazione e prosegue. Ad esempio, alcune impostazioni STIG potrebbero non essere applicabili ai server autonomi. Le politiche specifiche dell'organizzazione possono anche influire sulle impostazioni applicate dal componente di protezione avanzata, ad esempio l'obbligo per gli amministratori di rivedere le impostazioni dei documenti.

Per un elenco completo, consultate la Document Library. STIGs Per informazioni su come visualizzare l'elenco completo, consulta Strumenti di visualizzazione STIG.

> **Note**
>
> I componenti di STIG-Build-Linux-Medium protezione avanzata includono tutte le impostazioni STIG elencate che AWSTOE si applicano ai componenti di STIG-Build-Linux-Low protezione avanzata, oltre alle impostazioni STIG elencate specificamente per le vulnerabilità di categoria II.

RHEL 7 STIG versione 3 Release 15

Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria III (basse) per questa distribuzione Linux, oltre a:

- RHEL 7/CentOS 7/ AL2

  V-204405, V-204406, V-204407, V-204408, V-204409, V-204410, V-204411, V-204412, V-204413, V-204414, V-204415, V-204416, V-204417, V-204418, V-204422, V-204423, V-204426, V-204427, V-204431, V-204434, V-204435, V-204437, V-204449, V-204450, V-204451, V-204457, V-204466, V-204490, V-204491, V-204503, V-204507, V-204508, V-204510, V-204511, V-204512, V-204514, V-204515, V-204516, V-204517, V-204521, V-204524, V-204531, V-204536, V-204537, V-204538, V-204539, V-204540, V-204541, V-204542, V-204543, V-204544, V-204545, V-204546, V-204547, V-204548, V-204549, V-204550, V-204551, V-204552, V-204553, V-204554, V-204555, V-204556, V-204557, V-204558, V-204559, V-204560, V-204562, V-204563, V-204564, V-204565, V-204566, V-204567, V-204568, V-204572, V-204578, V-204579, V-204584, V-204585, V-204587, V-204588, V-204589, V-204590, V-204591, V-204592, V-204593, V-204595, V-204596, V-204597, V-204598, V-204599, V-204600, V-204601, V-204602, V-204609, V-204610, V-204611, V-204612, V-204613, V-204614, V-204615, V-204616, V-204617, V-204619, V-204622, V-204625, V-204630, V-204631, V-204633, V-233307, V-237634, V-237635, V-251703, -255925, V-255927, V-255928 e V-256970

RHEL 8 STIG versione 2 versione 2

Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria III (basse) per questa distribuzione Linux, oltre a:

- RHEL 8/CentOS 8/AL 2023

V-230238, V-230239, V-230257, V-230258, V-230259, V-230262, V-230273, V-230275, V-230316, V-230325, V-230366, V-230478, V-230488, V-230489, V-230505, V-230523, V-230550, V-230558 9, V-230560, V-230561, V237640, 244547, V-256974, V-230525, V-244544, V-230248, V-230249, V-230250, 230245, V-230246, V-230247, V-230397, V-230399, V-230400, V-230401, -244543, V-230228, V-230298, V-230387, V-230482, V-230231, V-230233, V-230324, V-230365, V-230370, V-230373, V-230378, V-230383, V-230236, V-230314, V-230315, V-244523, V 257258, V-230266, V-230267, V-230268, V-230280, V-230310, V-230311, V-230312, V-230502, V-230532, MA-230535, V-230536, V-230537, V-230538, 230539, 230540, 230541, 230542, 230543, 30544, 30545, V-230546, V-230547, V-230548, V-230549, V-244550, V-244551, V-244552, V-244553, V-244554, V250317, V-251718, V-230237, V-230313, V-230356, V-230357, V-230358, V-230358, V-230358, V-230357, V-230358, V-230358 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-23016, V-23030 332, V-230333, V-230335, V-230337, V-230339, V-230341, V-230343, V-230345, V-230240, V-230282, V250315, V250316, 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-230392, V-230392, V-230392, V-230392, V-230392, V-230392, V-230392, V-230398, V-230402, V-230396, V-230398, V-230398, V-230402, V-230392, V-230392, V-230398, V-230V-230404, V-230405, V-230406, V-230407, V-230408, V-230409, V-230410, V-230411, V-230412, V-230413, V-230418, 230419, V-230421, V-230422, V-230423, V-230423, V-230419, V-230421, V-230423, V-230423, V-230419, V-230421, V-230423, V-230423, V-230419, V-230421, V-230423, V-230423, V-230419, V-230421, V-230423, V-230423, V-23024, V-230425, V-230426, V-230427, V-230428, V-230429, V-230430, V-230431, V-230432, V-230433, 230434, V-230435, V-230436, V-230437, V-230438, V-230439, 230444, V-230446, V-230447, V-230448, V-230449, V-230455, V-230456, V-230462, V-230463, V-230464, V-230465, V-230466, V-230467, V-230471, V-230472, V-230473, V-230474, V-230480, V-230483, V-244542, V-230503, V-230507, V-244525, 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 e V-251711

RHEL 9 STIG versione 2 Release 3

Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria III (basse) per questa distribuzione Linux, oltre a:

- RHEL 9/CentOS Stream 9

V-257981, V-257780, V-257825, V-257827, V-257828, V-257829, V-257830, V-257831, V-257832, V-257833, V-257834, V-257836, V-257838, V-257839, V-257840, V-257841, V-257842, V-257882, V-257883, V-257884, V-257885, V-257889, V-257890, V-257891, V-257892, V-257893, V-257894, V-257895, V-257896, V-257897, V-257898, V-257899, V-257900, V-257901, V-257902, V-257903, V-257904, V-257905, V-257906, V-257907, V-257908, V-257909, V-257910, V-257911, V-257912, V-257913, V-257914, V-257915, V-257918, V-257919, V-257920, V-257921, V-257922, V-257923, V-257928, V-257929, V-257930, V-257934, V-257935, V-257943, V-257948, V-257954, V-257978, V-257980, V-258035, V-258041, V-258046, V-258052, V-258063, V-258068, V-258081, V-258082, V-258083, V-258089, V-258120, V-2580124 126, V-258140, V-258141, V-258151, V-258234, V-257886, V-257916, V-257917, V-257952, V-257936, V-257939, V-257940, V-258036, V-258038, V-257887, V-257924, V-257925, V-258145, V-258152, V-258153, V-257887, V-257924, V-257925, V-258145, V-258152, V-258153, V-257887, V-257924, V-257925, V-258145, V-258152, V-258153, V-257887, V-257924, V-257925, V-258145, V-258152, V-258153, V-257887, V-257924, V-257925, V-258145, V-V-258156, V-258157, V-258158, V-258159, V-258160, V-258161, V-258162, V-258163, V-258164, V-258165, V-258166, V-258167, V-258168, V-258169, V-258170, V-258171, V-258172, V-258176, V-258177, V-258178, V-258179, V-258180, V-258181, V-258182, V-258183, V-258184, V-258185, V-258186, V-258187, V-258188, V-258189, V-258190, V-258191, V-258192, V-258193, V-258194, V-258195, V-258196, V-258197, V-258198, V-258199, V-258200, V-258201, V-258202, V-258203, V-258204, V-258205, V-258206, V-258207, V-258208, V-258209, V-258210, V-258211, V-258212, V-258213, V-258214, V-258215, V-258216, V-258217, V-258219, V-258219, V-2582220, V-258221, V-258222, V-258223, V-258224, V-258225, V-258226, V-258227, V-258228, V-258229, V-257781, V-257783, V-257786, V-257797, V-257798, V-257799, V-257800, V-257801, V-257802, V-257803, V-257809, V-257810, V-257811, V-257812, V-257813, V-257815, V-257816, V-257942, V-257957, V-257958, V-257959, V-257960, V-257961, V-257962, V-257963, V-257964, V-257965, V-257966, V-257967, V-257968, V-257969, V-257970, V-257971, V-257972, V-257973, V-257974, V-257975, V-257976, V-257977, V-258077, V-258128, V V-258129, V-258043, V-258049, V-258071, V-258072, V-258073, V-258074, V-258075, V-258104, V-258105, V-258107, V-258108, V-258109, V-258110, V-258111, V-258117, V-258119, V-257979, V-257982, V-257983, V-257985, V-257987, V-257988, V-257992, V-257993, V-257994, V-257995, V-257996, V-257997, V-257998, V-257999, V-258000, V-258001, V-258002, V-258003, V-258004, V-258005, V-258006, V 258007, V-258008, V-258009, V-258010, V-258011, V-258130, V-257814, V-258054, V-258055, V-258056, V-258057, V-258060, V-258070, V-258080, V-258088, V-258091, V-258092, V-258093, V-258095, V-258095, V-258095, V-258091, V-258092, V-258093, V-258095, V-258095, V-258095, V-258091, V-258092, V-258093, V-258095, V-258095, V-258095, V-258091, V-258092, V-258093, V-258095, V-258095, V-2580V-8097, V-258098, V-258099, V-258100, V-258101, V-258102, V-258103, V-258112, V-258113, V-258114, V-258115, V-258116, V-258118, V-258233,

V-257951, V-257953, V-258142, V-258144, V-258146, V-258147, V-258148, V-258150, V-258064, V-258065, V-258066, V-258232, V-258237, V-258239, V-258240, V-257788, V-257790, V-257791, V-257792, V-257793, V-257794, V-257817, V-258175, V-257804, V-257805, V-257805, V-257805 6, V-257807, V-257808, V-258034, V-258039, V-257888, V-257926, V-257927, V-257933, V-258084, V-258085, V-257949, V-258040, V-257818, V-257849, V-258122, V-258123, V-258133, V-257944, V-258028, 258079, V-258125 e V-258137

Ubuntu 18.04 STIG versione 2 Release 15

Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria III (basse) per questa distribuzione Linux, oltre a:

V-219149, V-219155, V-219156, V-219160, V-219166, V-219168, V-219176, V-219181, V-219184, V-219186, V-219188, V-219189, V-219190, V-219191, V-219192, V-219193, V-219194, V-219195, V-219196, V-219197, V-219198, V-219199, V-219200, V-219201, V-219202, V-219203, V-219204, V-219205, V-219206, V-219207, V-219208, V-219209, V-219213, V-219214, V-219215, V-219216, V-219217, V-219218, V-219219, V-219220, V-219221, V-219222, V-219223, V-219224, V-219225, V-219226, V-219227, V-219228, V-219229, V-219230, V-219231, V-219232, V-219233, V-219234, V-219235, V-219236, V-219238, V-219239, V-219240, V-219241, V-219242, V-219243, V-219244, V-219250, V-219254, V-219257, V-219263, V-219264, V-219265, V-219266, V-219267, V-219268, V-219269, V-219270, V-219271, V-219272, V-219273, V-219274, V-219275, V-219276, V-219277, V-219279, V-219281, V-219287, V-219291, V-219297, V-219298, V-219299, V-219300, V-219303, V-219304, V-219306, V-219309, V-219310, V-219311, V-219312, V-219315, V-219318, V-219319, V-219323, V-219326, V-219328, V-219330, V-219331, V-219335, V-219336, V-219337, V-219338, V-219339, V-219342, V-219344, V-233779, V-233780 e V- 255906

Ubuntu 20.04 STIG versione 2 Release 2

Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria III (basse) per questa distribuzione Linux, oltre a:

V-238200, V-238205, V-238207, V-238231, 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-238371, V-238376, V-238377 V-238378, V-238209, V-238325, V-238330, V-238333, V-238369, V-238230, V-238338, V-238341, V-238342, V-238343, V-238324, V-238353, V-238228, V-238225, V-238227, V-238299, V-238238, V-238239, V-238240, V-238241, V-238241, V-238241, V-238241, V-238238 242, V-238244, V-238245, V-238246, V-238247, V-238248, V-238249, V-238250, V-238251, V-238252, V-238253, V-238254, V-238255, V-238256, V-238257, V-238258, V-238264, V-238268,

V-238271, V-238277, V-238278, V-238279, V-238280, V-238281, V-238282, V-238283, V-238284, V-238285, V-238286, V-238287, V-238288, V-238289, V-238290, V-238291, V-238292, V-238293, V-238294, V-238295, V-238297, V-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-238210, V-238211, V-238212, V-238213, V-238220, V-255912, V-238355, V-238236, V-238303, V-238356, V-238359, V-238370, V-238334 e V-238232

Ubuntu 22.04 STIG versione 2 Release 3

Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria III (basse) per questa distribuzione Linux, oltre a:

V-260546, V-260473, V-260474, V-260478, V-260485, V-260486, V-260487, V-260488, V-260489, V-260490, V-260495, V-260496, V-260496, V-260497, V-260498, V-260499, V-260500, V-260508, V-260513, V-260522, V-260522, V-260522, V-260599 40, V-260542, V-260543, V-260547, V-260554, V-260572, V-260590, V-260475, V-260491, V-260510, V-260511, V-260492, V-260493, V-260494, V-260507, V-260591, V-260594, V-260597, V-260598, V-260505 99, V-260600, V-260601, V-260602, V-260603, V-260604, V-260605, V-260606, V-260607, V-260608, V-260609, V-260610, V-260611, V-260612, V-260613, V-260614, V-260615, V-260616, V-260617, V-260618, V-260619, V-260620, V-260621, V-260622, V-260623, V-260624, V-260625, V-260626, V-260627, V-260628, V-260629, V-260630, V-260631, V-260632, V-260633, V-260634, V-260635, V-260636, V-260637, V-260638, V-260639, V-260640, V-260641, V-260642, V-260643, V-260644, V-260645, V-260646, V-260647, V-260648, V-260649, V-260471, V-260514, V-260553, V-260576, V-260574, V-260560, V-260561, V-260562, V-260563, V-260564, V-260565, V-260566, V-260567, V-260569, V-260527, V-260528, V-260530, V-260533, V-260534, V-260575, V-260505, V-260506, V-260512, V-260582, V-260584, V-260585, V-260586, V-260477, V-260545, V-260555, V-260556, V-260557, V-260509, V-260588, V-260589, V-260537, V-260538 e V-260535

Ubuntu 24.04 STIG versione 1 versione 1

Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria III (basse) per questa distribuzione Linux, oltre a:

V-270649, V-270651, V-270652, V-270653, V-270654, V-270656, V-270657, V-270659, V-270660, V-270661, V-270662, V-270663, V-270669, V-270672, V-270673, V-270674, V-270676, V-270678, V-270679, V-27679, V-27679 0680, V-270681, V-270683, V-270684, V-270685, V-270686, V-270687, V-270688, V-270689, V-270692, V-270693, V-270696, V-270697, V-270698, V-270699, V-270700, V-270701, V-270702, V-270703, V-270703, V-270703, V-270703, V-27070 704, V-270705, V-270709, V-270715, V-270716, V-270718, V-270720, V-270721, V-270722, V-270723,

V-270724, V-270725, V-270726, V-270727, V-270728, V-270729, V-270730, V-270731, V-270732, V-270733, V-270737, V-270739, V-270740, V-270741, V-270742, V-270743, V-270746, V-270750, V-270753, V-270755, V-270756, V-270757, V-270758, V-270759, V-270760, V-270765, V-270766, V-270767, V-270768, V-270769, V-270770, V-270771, V-270772, V-270773, V-270775, V-270776, V-270777, V-270778, V-270779, V-270780, V-270781, V-270782, V-270783, V-270784, V-270785, V-270786, V-270787 0788, V-270789, V-270790, V-270791, V-270792, V-270793, V-270794, V-270795, V-270796, V-270797, V-270798, V-270799, V-270800, V-270801, V-270802, V-270803, V-270804, V-270805, V-270806, V-270807, V-270808, V-270809, V-270810, V-270811, V-270812, V-270813, V-270814, V-270815, V-270821, V-270822, V-270823, V-270824, V-270825, V-270826, V-270827, V-270828, V-270829, V-270830, V-270831 e V-270832

STIG-Build-Linux-High versione 2025.1.x

L'elenco seguente contiene le impostazioni STIG che il componente di protezione avanzata applica all'infrastruttura. Se un'impostazione supportata non è applicabile all'infrastruttura, il componente di protezione avanzata ignora tale impostazione e prosegue. Ad esempio, alcune impostazioni STIG potrebbero non essere applicabili ai server autonomi. Le politiche specifiche dell'organizzazione possono anche influire sulle impostazioni applicate dal componente di protezione avanzata, ad esempio l'obbligo per gli amministratori di rivedere le impostazioni dei documenti.

Per un elenco completo, consultate la Document Library. STIGs Per informazioni su come visualizzare l'elenco completo, consulta Strumenti di visualizzazione STIG.

> **Note**
>
> I componenti di STIG-Build-Linux-High protezione avanzata includono tutte le impostazioni STIG elencate STIG-Build-Linux-Low e AWSTOE valide per i componenti di STIG-Build-Linux-Medium protezione avanzata, oltre alle impostazioni STIG elencate che si applicano specificamente alle vulnerabilità di categoria I.

RHEL 7 STIG versione 3 Release 15

Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria II e III (medie e basse) per questa distribuzione Linux, oltre a:

- RHEL 7/CentOS 7/ AL2

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

RHEL 8 STIG versione 2 versione 2

Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria II e III (medie e basse) per questa distribuzione Linux, oltre a:

- RHEL 8/CentOS 8/AL 2023

  V-230223, V-230265, V-230529, V-230531, V-230264, V-230487, V-230492, V-230533, V-230558 e V-244540

RHEL 9 STIG versione 2 versione 3

Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria II e III (medie e basse) per questa distribuzione Linux, oltre a:

- RHEL 9/CentOS Stream 9

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

Ubuntu 18.04 STIG versione 2 versione 15

Include tutte le impostazioni STIG supportate applicate dal componente di protezione avanzata per le vulnerabilità di categoria II e III (medie e basse) per questa distribuzione Linux, oltre a:

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

Ubuntu 20.04 STIG versione 2 versione 2

Include tutte le impostazioni STIG supportate applicate dal componente di protezione avanzata per le vulnerabilità di categoria II e III (medie e basse) per questa distribuzione Linux, oltre a:

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

Ubuntu 22.04 STIG versione 2 versione 3

Include tutte le impostazioni STIG supportate che il componente di protezione avanzata applica per le vulnerabilità di categoria II e III (medie e basse) per questa distribuzione Linux, oltre a:

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

Ubuntu 24.04 STIG versione 1 versione 1

Include tutte le impostazioni STIG supportate applicate dal componente di protezione avanzata per le vulnerabilità di categoria II e III (medie e basse) per questa distribuzione Linux, oltre a:

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

Registro della cronologia delle versioni STIG per Linux

Questa sezione registra la cronologia delle versioni dei componenti Linux. Per visualizzare le modifiche e le versioni pubblicate per un trimestre, scegli il titolo per espandere le informazioni.

Modifiche al primo trimestre 2025 - 11/04/2025:

Sono state aggiornate le seguenti versioni di STIG, applicato STIGS per la versione del primo trimestre 2025 e aggiunto il supporto per Ubuntu 24.04:

STIG-Build-Linux-Low versione 2025.1.x

- RHEL 7 STIG versione 3 versione 15
- RHEL 8 STIG versione 2 versione 2
- RHEL 9 STIG versione 2 versione 3
- Ubuntu 18.04 STIG versione 2 versione 15
- Ubuntu 20.04 STIG versione 2 Release 2
- Ubuntu 22.04 STIG versione 2 versione 3
- Ubuntu 24.04 STIG versione 1 Release 1

STIG-Build-Linux-Medium versione 2025.1.x

- RHEL 7 STIG versione 3 versione 15
- RHEL 8 STIG versione 2 versione 2
- RHEL 9 STIG versione 2 versione 3
- Ubuntu 18.04 STIG versione 2 versione 15
- Ubuntu 20.04 STIG versione 2 Release 2
- Ubuntu 22.04 STIG versione 2 versione 3

- Ubuntu 24.04 STIG versione 1 Release 1

STIG-Build-Linux-High versione 2025.1.x

- RHEL 7 STIG versione 3 versione 15
- RHEL 8 STIG versione 2 versione 2
- RHEL 9 STIG versione 2 versione 3
- Ubuntu 18.04 STIG versione 2 versione 15
- Ubuntu 20.04 STIG versione 2 Release 2
- Ubuntu 22.04 STIG versione 2 versione 3
- Ubuntu 24.04 STIG versione 1 Release 1

Modifiche al quarto trimestre 2024 - 12/10/2024:

Sono state aggiornate le seguenti versioni di STIG, applicato STIGS per la versione del quarto trimestre 2024 e aggiunto informazioni su due nuovi parametri di input per i componenti Linux:

STIG-Build-Linux-Low versione 2024.4.x

- RHEL 7 STIG versione 3 versione 15
- RHEL 8 STIG versione 2 versione 1
- RHEL 9 STIG versione 2 versione 2
- Ubuntu 18.04 STIG versione 2 versione 15
- Ubuntu 20.04 STIG versione 2 Release 1
- Ubuntu 22.04 STIG versione 2 Release 2

STIG-Build-Linux-Medium versione 2024.4.x

- RHEL 7 STIG versione 3 versione 15
- RHEL 8 STIG versione 2 versione 1
- RHEL 9 STIG versione 2 versione 2
- Ubuntu 18.04 STIG versione 2 versione 15
- Ubuntu 20.04 STIG versione 2 Release 1
- Ubuntu 22.04 STIG versione 2 Release 2

STIG-Build-Linux-High versione 2024.4.x

- RHEL 7 STIG versione 3 versione 15
- RHEL 8 STIG versione 2 versione 1
- RHEL 9 STIG versione 2 versione 2
- Ubuntu 18.04 STIG versione 2 versione 15
- Ubuntu 20.04 STIG versione 2 Release 1
- Ubuntu 22.04 STIG versione 2 Release 2

Modifiche al terzo trimestre 2024 - 10/04/2024 (nessuna modifica):

Non sono state apportate modifiche al componente Linux STIGS per la versione del terzo trimestre 2024.

Modifiche al secondo trimestre 2024 - 05/10/2024:

Versioni STIG aggiornate e STIGS applicato per la versione del secondo trimestre 2024. È stato inoltre aggiunto il supporto per RHEL 9, CentOS Stream 9 e Ubuntu 22.04, come segue:

STIG-Build-Linux-Low versione 2024.2.x

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

STIG-Build-Linux-Medium versione 2024.2.x

- RHEL 7 STIG versione 3 versione 14
- RHEL 8 STIG versione 1 versione 14
- RHEL 9 STIG versione 1 versione 3
- Ubuntu 18.04 STIG versione 2 versione 14
- Ubuntu 20.04 STIG versione 1 versione 12

- Ubuntu 22.04 STIG versione 1 versione 1

STIG-Build-Linux-High versione 2024.2.x

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

Modifiche al primo trimestre 2024 - 02/06/2024:

Versioni STIG aggiornate e STIGS applicato per la versione del primo trimestre 2024 come segue:

STIG-Build-Linux-Low versione 2024.1.x

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

STIG-Build-Linux-Medium versione 2024.1.x

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

STIG-Build-Linux-High versione 2024.1.x

- RHEL 7 STIG versione 3 versione 14
- RHEL 8 STIG versione 1 versione 13
- Ubuntu 18.04 STIG versione 2 versione 13

- Ubuntu 20.04 STIG versione 1 versione 11

Modifiche al quarto trimestre 2023 - 12/07/2023:

Versioni STIG aggiornate e STIGS applicato per la versione del quarto trimestre 2023 come segue:

STIG-Build-Linux-Low versione 2023.4.x

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

STIG-Build-Linux-Medium versione 2023.4.x

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

STIG-Build-Linux-High versione 2023.4.x

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

Modifiche al terzo trimestre 2023 - 10/04/2023:

Versioni STIG aggiornate e STIGS applicato per la versione del terzo trimestre 2023 come segue:

STIG-Build-Linux-Low versione 2023.3.x

- RHEL 7 STIG versione 3 versione 12
- RHEL 8 STIG versione 1 versione 11

- Ubuntu 18.04 STIG versione 2 versione 11
- Ubuntu 20.04 STIG versione 1 versione 9

STIG-Build-Linux-Medium versione 2023.3.x

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

STIG-Build-Linux-High versione 2023.3.x

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

Modifiche al secondo trimestre 2023 - 05/03/2023:

Versioni STIG aggiornate e STIGS applicato per la versione del secondo trimestre 2023 come segue:

STIG-Build-Linux-Low versione 2023.2.x

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

STIG-Build-Linux-Medium versione 2023.2.x

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

STIG-Build-Linux-High versione 2023.2.x

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

Modifiche al 1° trimestre 2023 - 27/03/2023:

Versioni STIG aggiornate e STIGS applicato per la versione del primo trimestre 2023 come segue:

STIG-Build-Linux-Low versione 2023.1.x

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

STIG-Build-Linux-Medium versione 2023.1.x

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

STIG-Build-Linux-High versione 2023.1.x

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

Modifiche al quarto trimestre 2022 - 02/01/2023:

Versioni STIG aggiornate e STIGS applicato per la versione del quarto trimestre 2022 come segue:

STIG-Build-Linux-Low versione 2022.4.x

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

STIG-Build-Linux-Medium versione 2022.4.x

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

STIG-Build-Linux-High versione 2022.4.x

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

Modifiche al terzo trimestre 2022 - 30/09/2022 (nessuna modifica):

Non ci sono state modifiche per il componente Linux STIGS per la versione del terzo trimestre 2022.

Modifiche al secondo trimestre 2022 - 08/02/2022:

È stato introdotto il supporto per Ubuntu, le versioni STIG aggiornate e STIGS applicato per la versione del secondo trimestre 2022 come segue:

STIG-Build-Linux-Low versione 2022.2.x

- RHEL 7 STIG versione 3 Release 7
- RHEL 8 STIG versione 1 versione 6
- Ubuntu 18.04 STIG Versione 2 Release 6 (nuova)

- Ubuntu 20.04 STIG Versione 1 Release 4 (nuova)

STIG-Build-Linux-Medium versione 2022.2.x

- RHEL 7 STIG versione 3 Release 7
- RHEL 8 STIG versione 1 versione 6
- Ubuntu 18.04 STIG Versione 2 Release 6 (nuova)
- Ubuntu 20.04 STIG Versione 1 Release 4 (nuova)

STIG-Build-Linux-High versione 2022.2.x

- RHEL 7 STIG versione 3 Release 7
- RHEL 8 STIG versione 1 versione 6
- Ubuntu 18.04 STIG Versione 2 Release 6 (nuova)
- Ubuntu 20.04 STIG Versione 1 Release 4 (nuova)

Modifiche al 1° trimestre 2022 - 26/04/2022:

Rifattorizzato per includere un migliore supporto per i contenitori. Combinato lo AL2 script precedente con RHEL 7. Versioni STIG aggiornate e STIGS applicato per la versione del primo trimestre 2022 come segue:

STIG-Build-Linux-Low versione 3.6.x

- RHEL 7 STIG versione 3 versione 6
- RHEL 8 STIG versione 1 versione 5

STIG-Build-Linux-Medium versione 3.6.x

- RHEL 7 STIG versione 3 versione 6
- RHEL 8 STIG versione 1 versione 5

STIG-Build-Linux-High versione 3.6.x

- RHEL 7 STIG versione 3 versione 6
- RHEL 8 STIG versione 1 versione 5

Modifiche al quarto trimestre 2021 - 20/12/2021:

Versioni STIG aggiornate e STIGS applicato per la versione del quarto trimestre 2021 come segue:

STIG-Build-Linux-Low versione 3.5.x

- RHEL 7 STIG versione 3 versione 5
- RHEL 8 STIG versione 1 versione 4

STIG-Build-Linux-Medium versione 3.5.x

- RHEL 7 STIG versione 3 versione 5
- RHEL 8 STIG versione 1 versione 4

STIG-Build-Linux-High versione 3.5.x

- RHEL 7 STIG versione 3 versione 5
- RHEL 8 STIG versione 1 versione 4

Modifiche al terzo trimestre 2021 - 30/09/2021:

Versioni STIG aggiornate e STIGS applicato per la versione del terzo trimestre 2021 come segue:

STIG-Build-Linux-Low versione 3.4.x

- RHEL 7 STIG versione 3 versione 4
- RHEL 8 STIG versione 1 versione 3

STIG-Build-Linux-Medium versione 3.4.x

- RHEL 7 STIG versione 3 versione 4
- RHEL 8 STIG versione 1 versione 3

STIG-Build-Linux-High versione 3.4.x

- RHEL 7 STIG versione 3 versione 4
- RHEL 8 STIG versione 1 versione 3

Componente del validatore di conformità SCAP

Il Security Content Automation Protocol (SCAP) è un insieme di standard che i professionisti IT possono utilizzare per identificare le vulnerabilità di sicurezza delle applicazioni ai fini della conformità. Lo SCAP Compliance Checker (SCC) è uno strumento di scansione convalidato dallo SCAP, rilasciato dal Naval Information Warfare Center (NIWC) Atlantic. Per ulteriori informazioni, vedere Security Content Automation Protocol (SCAP) Compliance Checker (SCC) sul sito web di NIWC Atlantic.

I `scap-compliance-checker-linux` componenti AWSTOE `scap-compliance-checker-windows` e scaricano e installano lo scanner SCC sulle istanze di compilazione e test della pipeline. Quando lo scanner è in esecuzione, esegue scansioni di configurazione autenticate utilizzando i benchmark DISA SCAP e fornisce un rapporto che include le seguenti informazioni. AWSTOE scrive inoltre le informazioni nei registri delle applicazioni.

- impostazioni STIG applicate all'istanza.
- Un punteggio di conformità complessivo per l'istanza.

Ti consigliamo di eseguire la convalida SCAP come fase finale del processo di creazione, per assicurarti di riportare risultati accurati di convalida della conformità.

> **Note**
>
> È possibile esaminare i report con uno degli strumenti di visualizzazione STIG. Questi strumenti sono disponibili online tramite DoD Cyber Exchange.

Le sezioni seguenti descrivono i benchmark inclusi nei componenti di convalida SCAP.

scap-compliance-checker-windows versione 2024.03.0

Il `scap-compliance-checker-windows` componente viene eseguito sulle EC2 istanze create da Image Builder per creare e testare l'immagine. AWSTOE registra sia il report che il punteggio prodotto dall'applicazione SCC.

Il componente esegue le seguenti fasi del flusso di lavoro:

1. Scarica e installa l'applicazione SCC.

2. Importa i benchmark di conformità.
3. Esegue la convalida utilizzando l'applicazione SCC.
4. Salva il rapporto di conformità e il punteggio localmente sul desktop dell'istanza di compilazione.
5. Registra il punteggio di conformità dal rapporto locale ai file di registro dell' AWSTOE applicazione.

> **Note**
>
> AWSTOE attualmente supporta la convalida della conformità SCAP per Windows Server 2012 R2 MS, 2016, 2019 e 2022.

Il componente di verifica della conformità SCAP per Windows include i seguenti benchmark:

Versione SCC: 5.10

Benchmark per il quarto trimestre 2023:

- U_MS_Defender_Antivirus_V2R5_STIG_SCAP_1-2_Benchmark
- U_MS_ _Framework_4-0_V2R2_STIG_SCAP_1-2_Benchmark DotNet
- U_MS_ IE11 _V2R6_STIG_SCAP_1-2_Benchmark
- U_MS_Windows_2012_and_2012_R2_DC_V3R5_STIG_SCAP_1-2_Benchmark
- U_MS_Windows_2012_and_2012_R2_MS_V3R5_STIG_SCAP_1-2_benchmark
- U_MS_Windows_Defender_Firewall_V2R3_STIG_SCAP_1-2_benchmark
- U_MS_Windows_Server_2016_V2R7_STIG_SCAP_1-2_benchmark
- U_MS_Windows_Server_2019_V3R2_STIG_SCAP_1-2_benchmark
- U_MS_Windows_Server_2022_V2R2_STIG_SCAP_1-2_benchmark
- U_CAN_Ubuntu_20-04_LTS_V1R10_STIG_SCAP_1-2_Benchmark
- U_RHEL_7_V3R15_STIG_SCAP_1-3_benchmark
- U_RHEL_8_V1R13_STIG_SCAP_1-3_benchmark
- U_RHEL_9_V2R1_STIG_SCAP_1-3_benchmark

scap-compliance-checker-linux versione 2021.04.0

Il `scap-compliance-checker-linux` componente viene eseguito sulle EC2 istanze create da Image Builder per creare e testare l'immagine. AWSTOE registra sia il report che il punteggio prodotto dall'applicazione SCC.

Il componente esegue le seguenti fasi del flusso di lavoro:

1. Scarica e installa l'applicazione SCC.
2. Importa i benchmark di conformità.
3. Esegue la convalida utilizzando l'applicazione SCC.
4. Salva il rapporto di conformità e il punteggio localmente, nella seguente posizione sull'istanza di build:. `/opt/scc/SCCResults`
5. Registra il punteggio di conformità dal rapporto locale ai file di registro dell' AWSTOE applicazione.

> **Note**
>
> AWSTOE attualmente supporta la convalida della conformità SCAP per RHEL 7/8 e Ubuntu 18.04/20.04. L'applicazione SCC attualmente supporta l'architettura x86 per la convalida.

Il componente SCAP Compliance Checker per Linux include i seguenti benchmark:

Versione SCC: 5.10

Benchmark per il quarto trimestre 2023:

- U_CAN_Ubuntu_20-04_LTS_V1R10_STIG_SCAP_1-2_benchmark
- U_RHEL_7_V3R15_STIG_SCAP_1-3_benchmark
- U_RHEL_8_V1R13_STIG_SCAP_1-3_benchmark
- U_RHEL_9_V2R1_STIG_SCAP_1-3_benchmark
- U_MS_Defender_Antivirus_V2R5_STIG_SCAP_1-2_Benchmark
- U_MS_ _Framework_4-0_V2R2_STIG_SCAP_1-2_Benchmark DotNet
- U_MS_ IE11 _V2R6_STIG_SCAP_1-2_Benchmark
- U_MS_Windows_2012_and_2012_R2_DC_V3R5_STIG_SCAP_1-2_Benchmark

- U_MS_Windows_2012_and_2012_R2_MS_V3R5_STIG_SCAP_1-2_benchmark
- U_MS_Windows_Defender_Firewall_V2R3_STIG_SCAP_1-2_benchmark
- U_MS_Windows_Server_2016_V2R7_STIG_SCAP_1-2_benchmark
- U_MS_Windows_Server_2019_V3R2_STIG_SCAP_1-2_benchmark
- U_MS_Windows_Server_2022_V2R2_STIG_SCAP_1-2_benchmark

Cronologia delle versioni di SCAP

La tabella seguente descrive le modifiche importanti all'ambiente e alle impostazioni SCAP descritte in questo documento.

Modifica	Descrizione	Data
Aggiornamenti SCAP del primo trimestre 2025	• scap-compliance-checker-windows versione 2024.03.0 (versione SCC: 5.10) • scap-compliance-checker-windows versione 2025.01.0 (versione SCC: 5.10)	11 aprile 2025
Aggiornamenti SCAP per il quarto trimestre del 2023	• scap-compliance-checker-windows versione 2023.04.0 (versione SCC: 5.8) • scap-compliance-checker-windows versione 2023.04.0 (versione SCC: 5.8)	20 dicembre 2021
Aggiornamenti SCAP per il terzo trimestre del 2023	• scap-compliance-checker-windows versione 2023.03.0 (versione SCC: 5.7.2) • scap-compliance-checker-linux versione 2023.03.0 (versione SCC: 5.7.2)	13 novembre 2023
		20 dicembre 2021

Modifica	Descrizione	Data
Componenti SCAP aggiunti	Sono stati introdotti i seguenti componenti SCAP: • scap-compliance-checker-linuxVersione creata 2021.04.0 (versione SCC: 5.4.2) • scap-compliance-checker-linuxVersione creata 2021.04.0 (versione SCC: 5.4.2)	

Sviluppa componenti personalizzati per la tua immagine Image Builder

È possibile creare componenti personalizzati per personalizzare le immagini di Image Builder in base alle specifiche esatte. Utilizza i seguenti passaggi per sviluppare componenti personalizzati per le tue ricette di immagini o contenitori di Image Builder.

1. Se desideri sviluppare il documento del componente e convalidarlo localmente, puoi installare l'applicazione AWS Task Orchestrator and Executor (AWSTOE) e configurarla sul tuo computer locale. Per ulteriori informazioni, consulta Configurazione manuale per sviluppare componenti personalizzati con AWSTOE.
2. Crea un documento componente che utilizzi il framework dei documenti AWSTOE dei componenti. Per ulteriori informazioni sul framework dei documenti, vedereUsa il framework AWSTOE dei documenti dei componenti per i componenti personalizzati.
3. Specificate il documento del componente quando create un componente personalizzato. Per ulteriori informazioni, consulta Crea un componente personalizzato con Image Builder.

Argomenti

- Creare un documento componente YAML per componenti personalizzati in Image Builder
- Crea un componente personalizzato con Image Builder

Creare un documento componente YAML per componenti personalizzati in Image Builder

Per creare un componente, è necessario fornire un documento relativo al componente dell'applicazione YAML o JSON. Il documento contiene il codice che viene eseguito durante le fasi e i passaggi definiti per personalizzare l'immagine.

Alcuni esempi in questa sezione creano un componente di compilazione che richiama il modulo di `UpdateOS` azione nell'applicazione di gestione dei AWSTOE componenti. Il modulo aggiorna il sistema operativo. Per ulteriori informazioni sul modulo di `UpdateOS` azione, vedereAggiorna il sistema operativo.

L'esempio del sistema operativo macOS utilizza il modulo di `ExecuteBash` azione per installare e verificare l'`wget`utilità. Il modulo di `UpdateOS` azione non supporta macOS. Per ulteriori informazioni sul modulo di `ExecuteBash` azione, consultaExecuteBash. Per ulteriori informazioni sulle fasi, i passaggi e la sintassi dei documenti dei componenti AWSTOE dell'applicazione, consulta Use documents in AWSTOE.

> **Note**
>
> Image Builder determina il tipo di componente dalle fasi definite nel documento del componente come segue:
>
> - Build: questo è il tipo di componente predefinito. Tutto ciò che non è classificato come componente di test è un componente di compilazione. Questo tipo di componente viene eseguito durante la fase di creazione dell'immagine. Se questo componente di compilazione ha una `test` fase definita, tale fase viene eseguita durante la fase di test.
> - Test: per qualificarsi come componente di test, il documento del componente deve includere solo una fase, denominata`test`. Per i test relativi alle configurazioni dei componenti di compilazione, ti consigliamo di non utilizzare un componente di test autonomo. Piuttosto, usa la `test` fase nel componente di compilazione associato.
>
> Per ulteriori informazioni su come Image Builder utilizza fasi e fasi per gestire il flusso di lavoro dei componenti nel processo di creazione, vedere. Usa i componenti per personalizzare l'immagine di Image Builder

Per creare un documento relativo al componente dell'applicazione YAML per un'applicazione di esempio, seguite i passaggi nella scheda corrispondente al sistema operativo dell'immagine.

Linux

Crea un file componente YAML

Usa uno strumento di modifica dei file per creare il documento componente. Gli esempi di documentazione utilizzano un file denominato*update-linux-os.yaml*, con il seguente contenuto:

```
# 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**
>
> Usa uno strumento come questo validatore YAML online o un'estensione YAML lint nel tuo ambiente di codice per verificare che il tuo YAML sia ben formato.

Windows

Crea un file componente YAML

Usa uno strumento di modifica dei file per creare il documento componente. Gli esempi di documentazione utilizzano un file denominato*update-windows-os.yaml*, con il seguente contenuto:

```
# 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**
>
> Usa uno strumento come questo validatore YAML online o un'estensione YAML lint nel tuo ambiente di codice per verificare che il tuo YAML sia ben formato.

macOS

Crea un file componente YAML

Usa uno strumento di modifica dei file per creare il documento componente. Gli esempi di documentazione utilizzano un file denominato*wget-macos.yaml*, con il seguente contenuto:

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

phases:
  - name: build
    steps:
      - name: WgetBuildStep
        action: ExecuteBash
        inputs:
          commands:
            - |
              PATH=/usr/local/bin:$PATH
              sudo -u ec2-user brew install wget


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

              type wget
```

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

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

> **Tip**
>
> Usa uno strumento come questo validatore YAML online o un'estensione YAML lint nel tuo ambiente di codice per verificare che il tuo YAML sia ben formato.

Crea un componente personalizzato con Image Builder

Dopo aver completato il documento del componente, è possibile utilizzarlo per creare un componente personalizzato utilizzabile nelle ricette di Image Builder. È possibile creare un componente personalizzato dalla console Image Builder, dall'API o SDKs dalla riga di comando. Per ulteriori informazioni su come creare un componente personalizzato con parametri di input e utilizzarlo nelle ricette, vedereTutorial: creare un componente personalizzato con parametri di input.

Le sezioni seguenti mostrano come creare componenti dalla console o da AWS CLI.

Indice

- Crea un componente personalizzato dalla console
- Creare un componente personalizzato dal AWS CLI
- Importa uno script per creare un componente da AWS CLI

Crea un componente personalizzato dalla console

Per creare un componente AWSTOE dell'applicazione dalla console Image Builder, procedi nel seguente modo:

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/

2. Seleziona Componenti dal pannello di navigazione. Quindi seleziona Crea componente.
3. Nella pagina Crea componente, in Dettagli del componente, inserisci quanto segue:

   a. Image Operating System (OS). Specificare il sistema operativo con cui il componente è compatibile.

   b. Categoria di componenti. Dal menu a discesa, seleziona il tipo di componente di build o test che stai creando.

   c. Nome del componente. Immettete un nome per il componente.

   d. Versione del componente. Immettere il numero di versione del componente.

   e. Descrizione. Fornisci una descrizione facoltativa per aiutarti a identificare il componente.

   f. Modificare la descrizione. Fornisci una descrizione facoltativa per aiutarti a comprendere le modifiche apportate a questa versione del componente.
4. Nella sezione Definizione del documento, l'opzione predefinita è Definisci il contenuto del documento. Il documento del componente definisce le azioni che Image Builder esegue sulle istanze di compilazione e test per creare l'immagine.

   Nella casella Contenuto, inserite il contenuto del documento del componente YAML. Per iniziare con un esempio di Hello World per Linux, scegli l'opzione Usa esempio. Per ulteriori informazioni su come creare un documento componente YAML o per copiare e incollare l'esempio UpdateOS da quella pagina, consulta. Creare un documento componente YAML per componenti personalizzati in Image Builder
5. Dopo aver inserito i dettagli del componente, seleziona Crea componente.

   > **Note**
   >
   > Per vedere il nuovo componente quando crei o aggiorni una ricetta, applica il filtro Owned by me all'elenco dei componenti di compilazione o test. Il filtro si trova nella parte superiore dell'elenco dei componenti, accanto alla casella di ricerca.
6. Per eliminare un componente, nella pagina Componenti, seleziona la casella di controllo accanto al componente che desideri eliminare. Dal menu a discesa Azioni, seleziona Elimina componente.

Aggiorna un componente

Per creare una nuova versione del componente, procedi nel seguente modo:

1. A seconda di dove inizi:

   - Dalla pagina dell'elenco dei componenti: seleziona la casella di controllo accanto al nome del componente, quindi seleziona Crea nuova versione dal menu Azioni.
   - Dalla pagina dei dettagli del componente: scegli il pulsante Crea nuova versione nell'angolo in alto a destra dell'intestazione.
2. Le informazioni sul componente sono già compilate con i valori correnti quando viene visualizzata la pagina Crea componente. Segui la procedura di creazione di un componente per aggiornare il componente. Ciò garantisce l'immissione di una versione semantica unica nella versione Component. Per ulteriori informazioni sul controllo delle versioni semantiche per le risorse di Image Builder, vedere. Versionamento semantico in Image Builder

Creare un componente personalizzato dal AWS CLI

In questa sezione, imparerete come configurare e utilizzare i comandi di Image Builder AWS CLI per creare un componente AWSTOE dell'applicazione, come segue.

- Carica il documento del componente YAML in un bucket S3 a cui puoi fare riferimento dalla riga di comando.
- Crea il componente dell' AWSTOE applicazione con il comando. create-component
- Elenca le versioni dei componenti con il list-components comando e un filtro dei nomi per vedere quali versioni esistono già. È possibile utilizzare l'output per determinare quale dovrebbe essere la versione successiva per gli aggiornamenti.

Per creare un componente AWSTOE dell'applicazione da un documento YAML di input, seguite i passaggi corrispondenti alla piattaforma del sistema operativo di immagine.

Linux

Archivia il documento relativo ai componenti dell'applicazione in Amazon S3

Puoi utilizzare un bucket S3 come repository per il documento sorgente dei componenti AWSTOE dell'applicazione. Per archiviare il documento del componente, procedi nel seguente modo:

- Carica il documento su Amazon S3

  Se il documento è più piccolo di 64 KB, puoi saltare questo passaggio. I documenti di dimensioni pari o superiori a 64 KB devono essere archiviati in Amazon S3.

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

Crea un componente dal documento YAML

Per semplificare il create-component comando utilizzato in AWS CLI, create un file JSON che contenga tutti i parametri del componente che desiderate passare al comando. Includi la posizione del *update-linux-os.yaml* documento che hai creato in precedenza. La coppia `uri` chiave-valore contiene il riferimento al file.

> **Note**
>
> La convenzione di denominazione per i valori dei dati nel file JSON segue lo schema specificato per i parametri di richiesta operativa dell'API Image Builder. Per esaminare i parametri di richiesta del comando API, consultate il CreateComponentcomando nell'EC2 Image Builder API Reference.
> Per fornire i valori dei dati come parametri della riga di comando, fate riferimento ai nomi dei parametri specificati nel AWS CLI Command Reference.

1. Creazione di un file JSON di input nella CLI

   Utilizzate uno strumento di modifica dei file per creare un file denominato*create-update-linux-os-component.json*. Includi i seguenti contenuti:

```
{
 "name": "update-linux-os",
 "semanticVersion": "1.1.2",
 "description": "An example component that updates the Linux operating system",
 "changeDescription": "Initial version.",
 "platform": "Linux",
 "uri": "s3://amzn-s3-demo-destination-bucket/my-path/update-linux-os.yaml",
 "kmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/98765432-
b123-456b-7f89-0123456f789c",
 "tags": {
  "MyTagKey-purpose": "security-updates"
 }
}
```

2. Crea il componente

   Utilizzate il seguente comando per creare il componente, facendo riferimento al nome del file JSON creato nel passaggio precedente:

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

   > **Note**
   >
   > - È necessario includere l'annotazione `file://` all'inizio del percorso del file JSON.
   > - Il percorso del file JSON deve seguire la convenzione appropriata per il sistema operativo di base in cui si esegue il comando. Ad esempio, Windows utilizza la barra rovesciata (\) per fare riferimento al percorso della directory, mentre Linux e macOS utilizzano la barra (/).

Windows

Archivia il documento relativo ai componenti dell'applicazione in Amazon S3

Puoi utilizzare un bucket S3 come repository per il documento sorgente dei componenti AWSTOE dell'applicazione. Per archiviare il documento del componente, procedi nel seguente modo:

- Carica il documento su Amazon S3

  Se il documento è più piccolo di 64 KB, puoi saltare questo passaggio. I documenti di dimensioni pari o superiori a 64 KB devono essere archiviati in Amazon S3.

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

Crea un componente dal documento YAML

Per semplificare il create-component comando utilizzato in AWS CLI, create un file JSON che contenga tutti i parametri del componente che desiderate passare al comando. Includi la posizione del *`update-windows-os.yaml`* documento che hai creato in precedenza. La coppia `uri` chiave-valore contiene il riferimento al file.

> **Note**
>
> La convenzione di denominazione per i valori dei dati nel file JSON segue lo schema specificato per i parametri di richiesta operativa dell'API Image Builder. Per esaminare i parametri di richiesta del comando API, consultate il CreateComponentcomando nell'EC2 Image Builder API Reference.
> Per fornire i valori dei dati come parametri della riga di comando, fate riferimento ai nomi dei parametri specificati nel AWS CLI Command Reference.

1. Creazione di un file JSON di input nella CLI

   Utilizzate uno strumento di modifica dei file per creare un file denominato*create-update-windows-os-component.json*. Includi i seguenti contenuti:

```
{
 "name": "update-windows-os",
 "semanticVersion": "1.1.2",
 "description": "An example component that updates the Windows operating
 system.",
 "changeDescription": "Initial version.",
 "platform": "Windows",
 "uri": "s3://amzn-s3-demo-destination-bucket/my-path/update-windows-os.yaml",
 "kmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/98765432-
b123-456b-7f89-0123456f789c",
 "tags": {
  "MyTagKey-purpose": "security-updates"
 }
}
```

   > **Note**
   >
   > - È necessario includere l'annotazione `file://` all'inizio del percorso del file JSON.
   > - Il percorso del file JSON deve seguire la convenzione appropriata per il sistema operativo di base in cui si esegue il comando. Ad esempio, Windows utilizza la barra rovesciata (\) per fare riferimento al percorso della directory, mentre Linux e macOS utilizzano la barra (/).

2. Crea il componente

   Utilizzate il seguente comando per creare il componente, facendo riferimento al nome del file JSON creato nel passaggio precedente:

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

   > **Note**
   >
   > - È necessario includere l'annotazione `file://` all'inizio del percorso del file JSON.
   > - Il percorso del file JSON deve seguire la convenzione appropriata per il sistema operativo di base in cui si esegue il comando. Ad esempio, Windows utilizza la barra rovesciata (\) per fare riferimento al percorso della directory, mentre Linux e macOS utilizzano la barra (/).

macOS

Archivia il documento relativo ai componenti dell'applicazione in Amazon S3

Puoi utilizzare un bucket S3 come repository per il documento sorgente dei componenti AWSTOE dell'applicazione. Per archiviare il documento del componente, procedi nel seguente modo:

- Carica il documento su Amazon S3

  Se il documento è più piccolo di 64 KB, puoi saltare questo passaggio. I documenti di dimensioni pari o superiori a 64 KB devono essere archiviati in Amazon S3.

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

Crea un componente dal documento YAML

Per semplificare il create-component comando utilizzato in AWS CLI, create un file JSON che contenga tutti i parametri del componente che desiderate passare al comando. Includi la posizione del *`wget-macos.yaml`* documento che hai creato in precedenza. La coppia `uri` chiave-valore contiene il riferimento al file.

> **Note**
>
> La convenzione di denominazione per i valori dei dati nel file JSON segue lo schema specificato per i parametri di richiesta operativa dell'API Image Builder. Per esaminare i parametri di richiesta del comando API, consultate il CreateComponentcomando nell'EC2 Image Builder API Reference.
> Per fornire i valori dei dati come parametri della riga di comando, fate riferimento ai nomi dei parametri specificati nel AWS CLI Command Reference.

1. Creazione di un file JSON di input nella CLI

   Utilizzate uno strumento di modifica dei file per creare un file denominato*install-wget-macos-component.json*. Includi i seguenti contenuti:

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

2. Crea il componente

   Utilizzate il seguente comando per creare il componente, facendo riferimento al nome del file JSON creato nel passaggio precedente:

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

> **Note**
>
> - È necessario includere l'annotazione `file://` all'inizio del percorso del file JSON.
> - Il percorso del file JSON deve seguire la convenzione appropriata per il sistema operativo di base in cui si esegue il comando. Ad esempio, Windows utilizza la barra rovesciata (\) per fare riferimento al percorso della directory, mentre Linux e macOS utilizzano la barra (/).

AWSTOE controllo delle versioni dei componenti per gli aggiornamenti da AWS CLI

AWSTOE i nomi e le versioni dei componenti sono incorporati nell'Amazon Resource Name (ARN) del componente, dopo il prefisso del componente. Ogni nuova versione di un componente ha il proprio ARN univoco. I passaggi per creare una nuova versione sono esattamente gli stessi dei passaggi per creare un nuovo componente, purché la versione semantica sia unica per il nome del componente. Per ulteriori informazioni sul controllo delle versioni semantiche per le risorse di Image Builder, vedere. Versionamento semantico in Image Builder

Per assicurarvi di assegnare la versione logica successiva, procuratevi innanzitutto un elenco delle versioni esistenti del componente che desiderate modificare. Utilizzate il list-components comando con AWS CLI, e filtrate in base al nome.

In questo esempio, si filtra in base al nome del componente creato negli esempi precedenti di Linux. Per elencare il componente creato, utilizzate il valore del `name` parametro dal file JSON utilizzato nel create-component comando.

```
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"
        }
    ]
}
```

In base ai risultati, è possibile determinare quale dovrebbe essere la versione successiva.

Importa uno script per creare un componente da AWS CLI

Per alcuni scenari, potrebbe essere più semplice iniziare con uno script preesistente. Per questo scenario, è possibile utilizzare l'esempio seguente.

Questo esempio presuppone che sia presente un file chiamato *import-component.json* (come illustrato). Notate che il file fa riferimento direttamente a uno PowerShell script chiamato `AdminConfig.ps1` che è già stato caricato in*amzn-s3-demo-source-bucket*. Attualmente, `SHELL` è supportato per il componente`format`.

```
{
"name": "MyImportedComponent",
"semanticVersion": "1.0.0",
"description": "An example of how to import a component",
"changeDescription": "First commit message.",
"format": "SHELL",
"platform": "Windows",
"type": "BUILD",
"uri": "s3://amzn-s3-demo-source-bucket/AdminConfig.ps1",
"kmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/60763706-
b131-418b-8f85-3420912f020c"
}
```

Per creare il componente da uno script importato, eseguite il comando seguente.

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

> **Note**
>
> Per evitare addebiti imprevisti, assicuratevi di ripulire le risorse e le pipeline che avete creato sulla base degli esempi di questa guida. Per ulteriori informazioni sull'eliminazione di risorse in Image Builder, vedere. Eliminare risorse Image Builder obsolete o inutilizzate

In che modo Image Builder utilizza l' AWS Task Orchestrator and Executor applicazione per gestire i componenti

EC2 Image Builder utilizza l'applicazione AWS Task Orchestrator and Executor (AWSTOE) per orchestrare flussi di lavoro complessi, modificare le configurazioni di sistema e testare le immagini senza la necessità di script o codice DevOps aggiuntivi. Questa applicazione gestisce ed esegue componenti che implementano lo schema dichiarativo del documento.

AWSTOE è un'applicazione autonoma che Image Builder installa nelle sue istanze di build e test quando si crea un'immagine. È inoltre possibile installarla manualmente sulle EC2 istanze per creare componenti personalizzati. Non richiede alcuna configurazione aggiuntiva e può essere eseguito anche in locale.

Indice

- AWSTOE scarica
- Regioni supportate
- AWSTOE riferimento al comando
- Configurazione manuale per sviluppare componenti personalizzati con AWSTOE
- Usa il framework AWSTOE dei documenti dei componenti per i componenti personalizzati
- Moduli di azione supportati dal gestore AWSTOE dei componenti
- Configurare l'input per il comando AWSTOE run

AWSTOE scarica

Per l'installazione AWSTOE, scegli il link per il download adatto alla tua architettura e piattaforma. Se ti colleghi a un endpoint VPC per il tuo servizio (Image Builder, ad esempio), devi allegare una policy endpoint personalizzata che includa l'accesso al bucket S3 per i download. AWSTOE In caso contrario, le istanze di build e test non saranno in grado di scaricare lo script bootstrap () e installare

l'applicazione. `bootstrap.sh` AWSTOE Per ulteriori informazioni, consulta Crea una policy per gli endpoint VPC per Image Builder.

> **Important**
>
> AWS sta eliminando gradualmente il supporto per le versioni TLS 1.0 e 1.1. Per accedere al bucket S3 per i AWSTOE download, il software client deve utilizzare la versione TLS 1.2 o successiva. Per ulteriori informazioni, consulta questo post sul blog AWS sulla sicurezza.

Architettura	Piattaforma	Collegamento per il download	Esempio
386	AL 2 e 2023 RHEL 7, 8 e 9 Ubuntu 16.04, 18.04, 20.04, 22.04 e 24.04 CentOS 7 e 8 SUSE 12 e 15	`https://awstoe-<region>.s3.com/latest/linux/386/awstoe`	https://awstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/linux/386/awstoe
AMD64	AL 2 e 2023 RHEL 7, 8 e 9 Ubuntu 16.04, 18.04, 20.04, 22.04 e 24.04 CentOS 7 e 8 CentOS Stream 8 SUSE 12 e 15	`https://awstoe-<region>.s3.com/latest/linux/amd64/awstoe`	https://awstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/linux/amd64/awstoe
AMD64	macOS 10.14.x (Mojave), 10.15.x	`https://awstoe-region.s3.com/latest/`	https://awstoe-us-east-1.s3.us-east-1.amazonaws.com/

Architettura	Piattaforma	Collegamento per il download	Esempio
	(Catalina), 11.x (Big Sur), 12.x (Monterey)	`darwin/amd64/awstoe`	latest/darwin/amd64/awstoe
AMD64	Windows Server 2012 R2, 2016, 2019 e 2022	`https://awstoe-<region>.s3.com/latest/windows/amd64/awstoe.exe`	https://awstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/windows/amd64/awstoe.exe
ARM64	AL 2 e 2023 RHEL 7, 8 e 9 Ubuntu 16.04, 18.04, 20.04, 22.04 e 24.04 CentOS 7 e 8 CentOS Stream 8 SUSE 12 e 15	`https://awstoe-<region>.s3.com/latest/linux/arm64/awstoe`	https://awstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/linux/arm64/awstoe

Regioni supportate

AWSTOE è supportata come applicazione autonoma nelle seguenti regioni.

Regione AWS nome	Regione AWS
Stati Uniti orientali (Ohio)	us-east-2
US East (N. Virginia)	us-east-1
AWS GovCloud (Stati Uniti orientali)	us-gov-east-1
AWS GovCloud (Stati Uniti occidentali)	us-gov-west-1

Regione AWS nome	Regione AWS
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
Africa (Cape Town)	af-south-1
Asia Pacifico (Hong Kong)	ap-east-1
Asia Pacifico (Osaka-Locale)	ap-northeast-3
Asia Pacifico (Seoul)	ap-northeast-2
Asia Pacifico (Mumbai)	ap-south-1
Asia Pacific (Hyderabad)	ap-south-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacifico (Giacarta)	ap-southeast-3
Asia Pacifico (Tokyo)	ap-northeast-1
Canada (Central)	ca-central-1
Europa (Francoforte)	eu-central-1
Europa (Zurigo)	eu-central-2
Europa (Stoccolma)	eu-north-1
Europa (Milano)	eu-south-1
Europa (Spagna)	eu-south-2
Europa (Irlanda)	eu-west-1
Europe (London)	eu-west-2

Regione AWS nome	Regione AWS
Europe (Paris)	eu-west-3
Israele (Tel Aviv)	il-central-1
Medio Oriente (Emirati Arabi Uniti)	me-central-1
Medio Oriente (Bahrein)	me-south-1
Sud America (São Paulo)	sa-east-1
China (Beijing)	cn-north-1
China (Ningxia)	cn-northwest-1

AWSTOE riferimento al comando

AWSTOE è un'applicazione di gestione dei componenti a riga di comando che viene eseguita su EC2 istanze Amazon. Quando Image Builder avvia un'istanza di EC2 build o di test, viene installata sull'istanza. AWSTOE Quindi esegue AWSTOE i comandi AWS CLI per installare o convalidare i componenti specificati nell'immagine o nella ricetta del contenitore.

> **Note**
>
> Alcuni moduli di AWSTOE azione richiedono autorizzazioni elevate per essere eseguiti su un server Linux. Per utilizzare autorizzazioni elevate, inserisci come prefisso la sintassi del comando o esegui sudo il sudo su comando una sola volta quando accedi prima di eseguire i comandi collegati di seguito. Per ulteriori informazioni sui moduli di AWSTOE azione, vedere. Moduli di azione supportati dal gestore AWSTOE dei componenti

run

Utilizzate il run comando per eseguire gli script dei documenti YAML per uno o più documenti componenti.

validate

Esegui il validate comando per convalidare la sintassi del documento YAML per uno o più documenti componenti.

comando awstoe run

Questo comando esegue gli script dei documenti dei componenti YAML nell'ordine in cui sono inclusi nel file di configurazione specificato dal `--config` parametro o nell'elenco dei documenti dei componenti specificato dal parametro. `--documents`

> **Note**
>
> È necessario specificare esattamente uno dei seguenti parametri, mai entrambi:
> --config
> --documenti

Sintassi

```
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]
```

Parametri e opzioni

Parametri

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

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

Il file di configurazione (condizionale). Questo parametro contiene la posizione del file JSON che contiene le impostazioni di configurazione per i componenti in esecuzione con questo comando. Se si specificano le impostazioni dei run comandi in un file di configurazione, non è necessario

specificare il `--documents` parametro. Per ulteriori informazioni sulla configurazione degli input, vedereConfigurare l'input per il comando AWSTOE run.

Le località valide includono:

- Un percorso di file locale (*`./config-example.json`*)
- Un URI S3 () `s3://`*`bucket/key`*

--cw-ignore-failures

Forma breve: N/A

Ignora gli errori di registrazione nei registri. CloudWatch

--cw-log-group

Forma breve: N/A

Il `LogGroup` nome dei registri. CloudWatch

--cw-log-region

Forma breve: N/A

La AWS regione che si applica ai registri. CloudWatch

--cw-log-stream

Forma breve: N/A

Il `LogStream` nome dei CloudWatch log, che indica AWSTOE dove trasmettere il file. `console.log`

--document-s3-bucket-owner

Forma breve: N/A

L'ID dell'account del proprietario del bucket per i documenti basati su URI S3.

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

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

I documenti componenti (condizionali). Questo parametro contiene un elenco separato da virgole di posizioni di file per l'esecuzione dei documenti del componente YAML. Se si specificano documenti YAML per il run comando utilizzando il `--documents` parametro, non è necessario specificare il parametro. `--config`

Le posizioni valide includono:

- percorsi di file locali (*./component-doc-example.yaml*).
- S3 URIs (s3://*bucket/key*).
- Versione di build del componente Image Builder ARNs (arn:aws:imagebuilder:us-west--- /2021.12.02/1). *2:123456789012 my-example-component*

> **Note**
>
> Non ci sono spazi tra gli elementi nell'elenco, solo virgole.

--execution-id

Forma breve: -i

Questo è l'ID univoco che si applica all'esecuzione del run comando corrente. Questo ID è incluso nei nomi dei file di output e di registro, per identificare in modo univoco tali file e collegarli all'esecuzione corrente del comando. Se questa impostazione viene omessa, AWSTOE genera un GUID.

--log-directory

Forma breve: -l

La directory di destinazione in cui vengono AWSTOE archiviati tutti i file di registro dell'esecuzione di questo comando. Per impostazione predefinita, questa directory si trova all'interno della seguente directory principale:TOE_<DATETIME>_<EXECUTIONID>. Se non si specifica la directory di registro, AWSTOE utilizza la directory di lavoro corrente (.).

--log-s3-bucket-name

Forma breve: -b

Se i log dei componenti sono archiviati in Amazon S3 (consigliato) AWSTOE , carica i log delle applicazioni dei componenti nel bucket S3 indicato in questo parametro.

--log-s3-bucket-owner

Forma breve: N/A

Se i log dei componenti sono archiviati in Amazon S3 (consigliato), questo è l'ID dell'account proprietario per il bucket AWSTOE in cui vengono scritti i file di registro.

--log-s3-key-prefix

Forma breve: -k

Se i log dei componenti sono archiviati in Amazon S3 (consigliato), questo è il prefisso della chiave oggetto S3 per la posizione del registro nel bucket.

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

Forma breve: N/A

I parametri sono variabili mutabili definite nel documento del componente, con impostazioni che l'applicazione chiamante può fornire in fase di esecuzione.

--phases

Forma breve: -p

Un elenco separato da virgole che specifica quali fasi eseguire dai documenti del componente YAML. Se un documento componente include fasi aggiuntive, queste non verranno eseguite.

--state-directory

Forma breve: -s

Il percorso del file in cui sono archiviati i file di tracciamento dello stato.

--versione

Forma breve: -v

Specifica la versione dell'applicazione del componente.

Opzioni

--help

Forma breve: -h

Visualizza un manuale di aiuto per l'utilizzo delle opzioni dell'applicazione di gestione dei componenti.

--trace

Forma breve: -t

Abilita la registrazione dettagliata sulla console.

comando awstoe validate

Quando si esegue questo comando, convalida la sintassi del documento YAML per ciascuno dei documenti componenti specificati dal parametro. `--documents`

Sintassi

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

Parametri e opzioni

Parametri

--document-s3-bucket-owner

Forma breve: N/A

ID dell'account di origine dei documenti basati sull'URI S3 fornito.

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

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

I documenti che lo compongono (obbligatorio). Questo parametro contiene un elenco separato da virgole delle posizioni dei file in cui eseguire i documenti del componente YAML. Le posizioni valide includono:

- percorsi di file locali (*./component-doc-example.yaml*)
- S3 URIs () `s3://`*bucket/key*
- Versione di build del componente Image Builder ARNs (arn:aws:imagebuilder:us-west-527/ /2021.12.02/1) *2:123456789012 my-example-component*

> **Note**
>
> Non ci sono spazi tra gli elementi nell'elenco, solo virgole.

Opzioni

--help

Forma breve: -h

Visualizza un manuale di aiuto per l'utilizzo delle opzioni dell'applicazione di gestione dei componenti.

--trace

Forma breve: -t

Abilita la registrazione dettagliata sulla console.

Configurazione manuale per sviluppare componenti personalizzati con AWSTOE

L'applicazione AWS Task Orchestrator and Executor (AWSTOE) è un'applicazione autonoma che crea, convalida ed esegue comandi all'interno di un framework di definizione dei componenti. AWS i servizi possono essere utilizzati AWSTOE per orchestrare flussi di lavoro, installare software, modificare le configurazioni di sistema e testare le build di immagini.

Segui questi passaggi per installare manualmente l' AWSTOE applicazione e utilizzarla come applicazione autonoma per sviluppare componenti personalizzati. Image Builder si occupa di questi passaggi per voi, se utilizzate la console AWS CLI o i comandi di Image Builder per creare componenti personalizzati. Per ulteriori informazioni, consulta Crea componenti personalizzati con Image Builder.

Verifica la firma del download AWSTOE di installazione

Questa sezione descrive la procedura consigliata per verificare la validità del download dell'installazione per i sistemi operativi basati AWSTOE su Linux, macOS e Windows.

Argomenti

- Verifica la firma del download dell' AWSTOE installazione su Linux o macOS
- Verifica la firma del download di AWSTOE installazione su Windows

Verifica la firma del download dell' AWSTOE installazione su Linux o macOS

Questo argomento descrive la procedura consigliata per verificare la validità del download dell'installazione per i sistemi operativi basati AWSTOE su Linux o macOS.

Ogni volta che scarichi un'applicazione da Internet, ti consigliamo di autenticare l'identità del produttore del software. Inoltre, verificate che l'applicazione non sia stata alterata o danneggiata

da quando è stata pubblicata. Ciò consente di evitare di installare una versione dell'applicazione contenente un virus o altro malware.

Se, dopo aver eseguito i passaggi descritti in questo argomento, stabilite che il software dell' AWSTOE applicazione è alterato o danneggiato, non eseguite il file di installazione. Contatta invece. Supporto Per ulteriori informazioni sulle opzioni di supporto disponibili, consulta Supporto.

AWSTOE i file per i sistemi operativi basati su Linux e macOS sono firmati utilizzandoGnuPG, un'implementazione open source dello standard Pretty Good Privacy (OpenPGP) per le firme digitali sicure. GnuPG(noto anche comeGPG) fornisce l'autenticazione e il controllo dell'integrità tramite una firma digitale. Amazon EC2 pubblica una chiave pubblica e firme che puoi utilizzare per verificare gli strumenti Amazon CLI EC2 scaricati. Per ulteriori informazioni su PGP e GnuPG (GPG), consulta http://www.gnupg.org.

La prima fase prevede la verifica dell'affidabilità dell'autore del software. Scarica la chiave pubblica dell'autore del software, controlla l'autenticità di tale proprietario e quindi aggiungi la chiave pubblica al tuo keyring. Il keyring è una raccolta di chiavi pubbliche nota. Dopo aver confermato l'autenticità della chiave pubblica, puoi usarla per verificare la firma dell'applicazione.

Argomenti

- Installazione degli strumenti GPG
- Autenticazione e importazione della chiave pubblica
- Verifica della firma del pacchetto

Installazione degli strumenti GPG

Se il tuo sistema operativo è Linux, Unix o macOS, è probabile che gli strumenti GPG siano già installati. Per sapere se gli strumenti sono installati nel sistema, digita gpg al prompt dei comandi. Se gli strumenti GPG sono installati, viene visualizzato un prompt dei comandi GPG. Se gli strumenti GPG non sono installati, viene visualizzato un messaggio di errore che indica che il comando non è stato trovato. Puoi installare il pacchetto GnuPG da un repository.

Per installare gli strumenti GPG, seleziona il sistema operativo che corrisponde alla tua istanza.

Debian-based Linux

Da un terminale, esegui il comando seguente:

```
apt-get install gnupg
```

Red Hat–based Linux

Da un terminale, esegui il comando seguente:

```
yum install gnupg
```

macOS

Da un terminale, esegui il comando seguente:

```
brew install gnupg
```

Autenticazione e importazione della chiave pubblica

La fase successiva del processo consiste nell'autenticare la chiave AWSTOE pubblica e aggiungerla come chiave affidabile nel portachiavi. GPG

Per autenticare e importare la chiave pubblica AWSTOE

1. Recupera una copia della chiave pubblica GPG in uno dei modi seguenti:

   - Scarica la chiave da

     https://awstoe - **<region>** .s3. **<region>**.amazonaws. com/assets/awstoe.gpg. Ad esempio, https://awstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/assets/awstoe.gpg.
   - Copia la chiave dal seguente testo e incollala in un file denominato `awstoe.gpg`. Assicurati di includere quanto segue:

     ```
     -----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. Al prompt dei comandi nella directory in cui hai salvato awstoe.gpg, usa il seguente comando per importare la chiave pubblica nel tuo portachiavi. AWSTOE

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

I risultati restituiti dal comando saranno simili a quanto segue:

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

Annota il valore della chiave. Tale valore verrà usato nel passaggio successivo. Nell'esempio precedente, il valore della chiave è `F5AEBC52`.

3. Verifica l'impronta eseguendo il comando seguente, sostituendo key-value con il valore annotato nella fase precedente:

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

Questo comando restituisce risultati simili ai seguenti:

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

Inoltre, la stringa dell'impronta deve essere uguale al valore `F6DD E01C 869F D639 15E5 5742 DEBD C156 F5AE BC52` riportato nell'esempio precedente. Confronta l'impronta della chiave restituita con quella pubblicata in questa pagina. Le chiavi devono corrispondere. Se non corrispondono, non installate lo script di installazione e contattateci. AWSTOE Supporto

Verifica della firma del pacchetto

Dopo aver installato gli strumenti GPG, avere eseguito l'autenticazione, avere importato la chiave pubblica di AWSTOE e avere verificato che la chiave pubblica sia affidabile, puoi verificare la firma dello script di installazione.

Per verificare la firma dello script di installazione di

1. Al prompt dei comandi, esegui il comando seguente per scaricare il file binario dell'applicazione:

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

   Per esempio:

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

   I valori supportati per **`architecture`** can be `amd64`386, e`arm64`.

2. Al prompt dei comandi, esegui il comando seguente per scaricare il file della firma per il file binario dell'applicazione corrispondente dallo stesso percorso del prefisso della chiave S3:

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

   Per esempio:

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

   I valori supportati per **`architecture`** can be`amd64`, e386. `arm64`

3. Verifica la firma eseguendo il comando seguente al prompt dei comandi nella directory in cui hai salvato `awstoe.sig` e nel file di AWSTOE installazione. Entrambi i file devono essere presenti.

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

   L'output deve essere simile al seguente:

```
gpg: Signature made Mon 20 Jul 2020 08:54:55 AM IST using RSA key ID F5AEBC52
gpg: Good signature from "AWSTOE awstoe@amazon.com" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: F6DD E01C 869F D639 15E5 5742 DEBD C156 F5AE BC52
```

Se l'output contiene la frase `Good signature from "AWSTOE <awstoe@amazon.com>"`, significa che la firma è stata verificata correttamente ed è possibile eseguire lo script di installazione di AWSTOE .

Se l'output include la frase `BAD signature`, controllare di avere eseguito la procedura correttamente. Se continui a ricevere questa risposta, non eseguire il file di installazione scaricato in precedenza e contatta Supporto.

Di seguito sono elencati i dettagli sugli avvisi che potrebbero comparire:

- ATTENZIONE: questa chiave non è certificata con una firma affidabile! Non vi è alcuna indicazione che la firma appartenga al proprietario. Idealmente, visiteresti un AWS ufficio e riceveresti la chiave di persona. Tuttavia, molto probabilmente lo scaricherai da un sito Web. In questo caso, il sito Web è un AWS sito Web.
- gpg: no ultimately trusted keys found. Ciò significa che la chiave specifica non è «in ultima analisi considerata affidabile» da te o da altre persone di cui ti fidi.

Per ulteriori informazioni, consulta http://www.gnupg.org.

Verifica la firma del download di AWSTOE installazione su Windows

In questo argomento viene descritto il processo consigliato per verificare la validità del file di installazione dell' AWS Task Orchestrator and Executor applicazione sui sistemi operativi basati su Windows.

Quando si esegue il download di un'applicazione da Internet, ti consigliamo di autenticare l'identità dell'autore del software e controllare che l'applicazione non risulti modificata o danneggiata rispetto alla versione pubblicata. Ciò consente di evitare di installare una versione dell'applicazione contenente un virus o altro malware.

Se, dopo aver eseguito i passaggi descritti in questo argomento, si determina che il software dell' AWSTOE applicazione è alterato o danneggiato, non eseguite il file di installazione. Contatta Supporto invece.

Per verificare la validità del file binario awstoe scaricato su sistemi operativi basati su Windows, assicurati che l'impronta digitale del relativo certificato di firma Amazon Services LLC sia uguale a questo valore:

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

> **Note**
>
> Durante la finestra di implementazione di un nuovo file binario, il certificato del firmatario potrebbe non corrispondere alla nuova impronta digitale. Se il certificato del firmatario non corrisponde, verifica che il valore dell'impronta digitale sia:
> F8 83 11 EE F0 4A A2 91 E3 79 21 BA 6B FC AF F8 19 92 12 D7

Per verificare questo valore, esegui la procedura seguente:

1. Clicca con il pulsante destro del mouse sul file `awstoe.exe`, e apri la finestra Proprietà.
2. Scegli la scheda Firme digitali.
3. In Elenco firme scegli Amazon Services LLC, quindi Dettagli.
4. Scegli la scheda Generale, se non è già selezionata, quindi scegli Visualizza certificato.
5. Scegli la scheda Dettagli, quindi seleziona Tutto nell'elenco a discesa Mostra, se non è già selezionato.
6. Scorri fino a visualizzare il campo Identificazione personale, quindi scegli Identificazione personale. In questo modo viene visualizzato l'intero valore dell'identificazione personale nella finestra inferiore.

   - Se il valore dell'identificazione personale nella finestra inferiore è identico al valore seguente:

     BA 81 25 EE AC64 2E A9 F3 C5 93 CA6D 3E B7 93 7D 68 75 74

     allora il file AWSTOE binario scaricato è autentico e può essere installato in sicurezza.
   - Se il valore dell'impronta digitale nella finestra dei dettagli inferiore non è identico al valore precedente, non eseguite. `awstoe.exe`

Passaggi introduttivi

- Fase 1: Installazione AWSTOE
- Fase 2: Impostare AWS le credenziali
- Fase 3: Sviluppare localmente i documenti dei componenti
- Fase 4: Convalida dei componenti AWSTOE
- Passaggio 5: Esegui i componenti AWSTOE

Fase 1: Installazione AWSTOE

Per sviluppare componenti localmente, scaricate e installate l' AWSTOE applicazione.

1. Scaricate l' AWSTOE applicazione

   Per l'installazione AWSTOE, scegli il link di download appropriato per la tua architettura e piattaforma. Per l'elenco completo dei link per il download delle applicazioni, consulta AWSTOE scarica

   > **Important**
   >
   > AWS sta eliminando gradualmente il supporto per le versioni TLS 1.0 e 1.1. Per accedere al bucket S3 per i AWSTOE download, il software client deve utilizzare la versione TLS 1.2 o successiva. Per ulteriori informazioni, consulta questo post sul blog AWS sulla sicurezza.

2. Verifica la firma

   I passaggi per la verifica del download dipendono dalla piattaforma server su cui viene eseguita l' AWSTOE applicazione dopo l'installazione. Per verificare il download su un server Linux, consultaVerifica la firma su Linux o macOS. Per verificare il download su un server Windows, consultaVerifica la firma su Windows.

> **Note**
>
> AWSTOE viene richiamato direttamente dalla posizione di download. Non è necessaria una fase di installazione separata. Ciò significa anche che AWSTOE può apportare modifiche all'ambiente locale.

Per assicurarti di isolare le modifiche durante lo sviluppo dei componenti, ti consigliamo di utilizzare un' EC2 istanza per sviluppare e testare AWSTOE i componenti.

Fase 2: Impostare AWS le credenziali

AWSTOE richiede AWS credenziali per connettersi ad altri Servizi AWS, come Amazon S3 e CloudWatch Amazon, durante l'esecuzione di attività come:

- Scaricamento di AWSTOE documenti da un percorso Amazon S3 fornito dall'utente.
- Moduli in esecuzione `S3Download` o di `S3Upload` azione.
- Lo streaming si registra su CloudWatch, se abilitato.

Se stai AWSTOE eseguendo un' EC2 istanza, l'esecuzione AWSTOE utilizza le stesse autorizzazioni del ruolo IAM associato all' EC2 istanza.

Per ulteriori informazioni sui ruoli IAM per EC2, consulta Ruoli IAM per Amazon EC2.

Gli esempi seguenti mostrano come impostare AWS le credenziali utilizzando le variabili di `AWS_SECRET_ACCESS_KEY` ambiente `AWS_ACCESS_KEY_ID` e.

Per impostare queste variabili su Linux, macOS o Unix, usa. `export`

```
export AWS_ACCESS_KEY_ID=your_access_key_id
```

```
export AWS_SECRET_ACCESS_KEY=your_secret_access_key
```

Per impostare queste variabili su Windows utilizzando PowerShell, usa. `$env`

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

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

Per impostare queste variabili su Windows utilizzando il prompt dei comandi, usa`set`.

```
set AWS_ACCESS_KEY_ID=your_access_key_id
```

```
set AWS_SECRET_ACCESS_KEY=your_secret_access_key
```

Fase 3: Sviluppare localmente i documenti dei componenti

I componenti sono creati con documenti YAML in testo semplice. Per ulteriori informazioni sulla sintassi dei documenti, vedere. Usa il framework AWSTOE dei documenti dei componenti per i componenti personalizzati

Di seguito sono riportati alcuni esempi di documenti del componente Hello World per aiutarti a iniziare.

Linux

Alcuni esempi di componenti Linux in questa guida si riferiscono a un file di documento del componente denominato`hello-world-linux.yml`. È possibile utilizzare il seguente documento per iniziare con questi esempi.

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

Windows

Alcuni esempi di componenti Windows in questa guida si riferiscono a un file di documento del componente denominato`hello-world-windows.yml`. È possibile utilizzare il seguente documento per iniziare con questi esempi.

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

macOS

Alcuni esempi di componenti macOS in questa guida si riferiscono a un file di documento del componente denominato. `hello-world-macos.yml` Puoi usare il seguente documento per iniziare con questi esempi.

```
name: Hello World
description: This is hello world testing document for macOS.
schemaVersion: 1.0
phases:
  - name: build
```

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

Fase 4: Convalida dei componenti AWSTOE

È possibile convalidare la sintassi dei AWSTOE componenti localmente con l'applicazione. AWSTOE Gli esempi seguenti mostrano il `validate` comando AWSTOE application per convalidare la sintassi di un componente senza eseguirlo.

> **Note**
>
> L' AWSTOE applicazione può convalidare solo la sintassi del componente per il sistema operativo corrente. Ad esempio, quando è in esecuzione `awstoe.exe` su Windows, non è possibile convalidare la sintassi di un documento Linux che utilizza il modulo di azione. `ExecuteBash`

Linux o macOS:

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

Windows

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

Passaggio 5: Esegui i componenti AWSTOE

L' AWSTOE applicazione può eseguire una o più fasi di documenti specifici utilizzando l'argomento della `--phases` riga di comando. I valori supportati per `--phases` sono `build`, `validate`, `test`. È possibile immettere valori di fase multipli come valori separati da virgole.

Quando si fornisce un elenco di fasi, l' AWSTOE applicazione esegue in sequenza le fasi specificate di ciascun documento. Ad esempio, AWSTOE esegue le `validate` fasi `build` e di`document1.yaml`, seguite dalle `validate` fasi `build` e di`document2.yaml`.

Per garantire che i log siano archiviati in modo sicuro e conservati per la risoluzione dei problemi, consigliamo di configurare lo storage dei log in Amazon S3. In Image Builder, la posizione Amazon S3 per la pubblicazione dei log è specificata nella configurazione dell'infrastruttura. Per ulteriori informazioni sulla configurazione dell'infrastruttura, consulta Gestione della configurazione dell'infrastruttura Image Builder

Se non viene fornito un elenco di fasi, l' AWSTOE applicazione esegue tutte le fasi nell'ordine indicato nel documento YAML.

Per eseguire fasi specifiche in uno o più documenti, utilizzate i seguenti comandi.

Monofase

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

Fasi multiple

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

Esecuzione del documento

Esegui tutte le fasi in un unico documento

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

Esegui tutte le fasi in più documenti

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

Inserisci le informazioni di Amazon S3 per caricare AWSTOE i log da un percorso locale definito dall'utente (consigliato)

```
awstoe run --documents documentName.yaml --log-s3-bucket-name amzn-s3-demo-destination-bucket --log-s3-key-prefix S3KeyPrefix --log-s3-bucket-owner S3BucketOwner --log-directory local_path
```

Esegui tutte le fasi in un unico documento e visualizza tutti i log sulla console

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

Comando della di esempio

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

Esegui il documento con un ID univoco

```
awstoe run --documents documentName.yaml --execution-id user-provided-id --
phases build,test
```

Fatti aiutare con AWSTOE

```
awstoe --help
```

Usa il framework AWSTOE dei documenti dei componenti per i componenti personalizzati

Per creare un componente utilizzando il framework di componenti AWS Task Orchestrator and Executor (AWSTOE), devi fornire un documento basato su YAML che rappresenti le fasi e i passaggi applicabili al componente che crei. Servizi AWS usa il tuo componente quando creano una nuova Amazon Machine Image (AMI) o un'immagine del contenitore.

Argomenti

- Flusso di lavoro relativo ai documenti
- Registrazione dei componenti
- Concatenamento di input e output
- Schema e definizioni del documento

- Esempi di documenti
- Usa le variabili nel tuo documento componente personalizzato
- Usa costrutti condizionali in AWSTOE
- Usa gli operatori di confronto nei documenti AWSTOE dei componenti
- Utilizzo di operatori logici nei documenti AWSTOE dei componenti
- Usa costrutti a ciclo continuo in AWSTOE

Flusso di lavoro relativo ai documenti

Il documento del AWSTOE componente utilizza fasi e passaggi per raggruppare le attività correlate e organizzarle in un flusso di lavoro logico per il componente.

> **Tip**
>
> Il servizio che utilizza il componente per creare un'immagine potrebbe implementare regole su quali fasi utilizzare per il processo di creazione e quando tali fasi possono essere eseguite. Questo è importante da considerare quando si progetta il componente.

Fasi

Le fasi rappresentano la progressione del flusso di lavoro attraverso il processo di creazione dell'immagine. Ad esempio, il servizio Image Builder utilizza `validate` le fasi `build` e le fasi durante la fase di creazione per le immagini che produce. Utilizza `container-host-test` le fasi `test` and durante la fase di test per garantire che l'istantanea dell'immagine o l'immagine del contenitore produca i risultati previsti prima di creare l'AMI finale o distribuire l'immagine del contenitore.

Quando il componente viene eseguito, i comandi associati per ogni fase vengono applicati nell'ordine in cui appaiono nel documento del componente.

Regole per le fasi

- Il nome di ogni fase deve essere univoco all'interno di un documento.
- È possibile definire molte fasi del documento.
- È necessario includere almeno una delle seguenti fasi nel documento:
  - build: per Image Builder, questa fase viene generalmente utilizzata durante la fase di creazione.

  - validare: per Image Builder, questa fase viene generalmente utilizzata durante la fase di creazione.
  - test — per Image Builder, questa fase viene generalmente utilizzata durante la fase di test.
- Le fasi vengono sempre eseguite nell'ordine in cui sono definite nel documento. L'ordine in cui sono specificate per AWSTOE i comandi in non AWS CLI ha effetto.

Fasi

I passaggi sono singole unità di lavoro che definiscono il flusso di lavoro all'interno di ciascuna fase. Le fasi vengono eseguite in ordine sequenziale. Tuttavia, l'input o l'output di una fase possono anche alimentare una fase successiva come input. Questo processo si chiama «concatenamento».

Regole per i passaggi

- Il nome della fase deve essere univoco per la fase.
- La fase deve utilizzare un'azione supportata (modulo di azione) che restituisca un codice di uscita.

  Per un elenco completo dei moduli di azione supportati, del loro funzionamento, input/output dei valori e degli esempi, consultaModuli di azione supportati dal gestore AWSTOE dei componenti.

Registrazione dei componenti

AWSTOE crea una nuova cartella di registro sulle EC2 istanze utilizzate per creare e testare una nuova immagine, ogni volta che viene eseguito il componente. Per le immagini del contenitore, la cartella di registro viene archiviata nel contenitore.

Per facilitare la risoluzione dei problemi in caso di problemi durante il processo di creazione dell'immagine, il documento di input e tutti i file di output AWSTOE creati durante l'esecuzione del componente vengono archiviati nella cartella di registro.

Il nome della cartella di registro è composto dalle seguenti parti:

1. Directory di log: quando un servizio esegue un AWSTOE componente, questo passa nella directory di log, insieme ad altre impostazioni per il comando. Per gli esempi seguenti, mostriamo il formato di file di registro utilizzato da Image Builder.
   - Linux e macOS: `/var/lib/amazon/toe/`
   - Windows: `$env:ProgramFiles\Amazon\TaskOrchestratorAndExecutor\`
2. Prefisso del file: si tratta di un prefisso standard utilizzato per tutti i componenti: "». `TOE_`

3. Ora di esecuzione: si tratta di un timestamp in formato _HH-MM-SS_UTC-0. YYYY-MM-DD
4. ID di esecuzione: è il GUID assegnato quando vengono eseguiti uno o più componenti. AWSTOE

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

AWSTOE memorizza i seguenti file principali nella cartella di registro:

File di input

- document.yaml — Il documento utilizzato come input per il comando. Dopo l'esecuzione del componente, questo file viene memorizzato come artefatto.

File di output

- application.log: il registro dell'applicazione contiene informazioni a livello di debug con data e ora AWSTOE relative a ciò che accade durante l'esecuzione del componente.
- detailedoutput.json: questo file JSON contiene informazioni dettagliate sullo stato di esecuzione, gli input, gli output e gli errori per tutti i documenti, le fasi e i passaggi relativi al componente durante l'esecuzione.
- console.log — Il registro della console contiene tutte le informazioni sullo standard out (stdout) e sugli errori standard (stderr) che vengono scritte sulla console mentre il componente è in esecuzione. AWSTOE
- chaining.json: questo file JSON rappresenta le ottimizzazioni applicate per risolvere le espressioni di concatenamento. AWSTOE

> **Note**
>
> La cartella di registro potrebbe contenere anche altri file temporanei che non sono trattati qui.

Concatenamento di input e output

L'applicazione AWSTOE di gestione della configurazione fornisce una funzionalità per concatenare input e output scrivendo riferimenti nei seguenti formati:

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

oppure

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

La funzione di concatenamento consente di riciclare il codice e migliorare la manutenibilità del documento.

Regole per il concatenamento

- Le espressioni di concatenamento possono essere utilizzate solo nella sezione degli input di ogni passaggio.
- Le dichiarazioni con espressioni concatenate devono essere racchiuse tra virgolette. Per esempio:
  - Espressione non valida: `echo {{ phase.step.inputs.variable }}`
  - Espressione valida: `"echo {{ phase.step.inputs.variable }}"`
  - Espressione valida: `'echo {{ phase.step.inputs.variable }}'`
- Le espressioni concatenate possono fare riferimento a variabili di altri passaggi e fasi dello stesso documento. Tuttavia, il servizio di chiamata potrebbe avere regole che richiedono che le espressioni concatenate funzionino solo nel contesto di una singola fase. Ad esempio, Image Builder non supporta il concatenamento dalla fase di compilazione alla fase di test, poiché esegue ogni fase in modo indipendente.
- Gli indici nelle espressioni concatenate seguono l'indicizzazione a base zero. L'indice inizia con zero (0) per fare riferimento al primo elemento.

Examples (Esempi)

Per fare riferimento alla variabile source nella seconda voce del passaggio di esempio seguente, lo schema di concatenamento è`{{ 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'
```

Per fare riferimento alla variabile di output (uguale a «Hello») del seguente passaggio di esempio, lo schema di concatenamento è. `{{ build.SamplePowerShellStep.outputs.stdout }}`

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

Schema e definizioni del documento

Di seguito è riportato lo schema YAML per un documento.

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

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

Le definizioni dello schema per un documento sono le seguenti.

Campo	Descrizione	Tipo	Richiesto
nome	Nome del documento.	Stringa	No

Campo	Descrizione	Tipo	Richiesto
description	Descrizione del documento.	Stringa	No
schemaVersion	Versione dello schema del documento, attualmente 1.0.	Stringa	Sì
phases	Un elenco di fasi con i relativi passaggi.	Elenco	Sì

Le definizioni dello schema per una fase sono le seguenti.

Campo	Descrizione	Tipo	Richiesto
nome	Nome della fase.	Stringa	Sì
steps	Elenco delle fasi della fase.	Elenco	Sì

Le definizioni dello schema per una fase sono le seguenti.

Campo	Descrizione	Tipo	Richiesto	Valore predefinito
nome	Nome definito dall'utente per il passo.	Stringa		
action	Parola chiave relativa al modulo che esegue il passaggio.	Stringa		

Campo	Descrizione	Tipo	Richiesto	Valore predefinito
timeoutSeconds	Numero di secondi di esecuzione del passaggio prima di fallire o riprovare. Inoltre, supporta il valore -1, che indica un timeout infinito. 0 e altri valori negativi non sono consentiti.	Numero intero	No	7.200 sec (120 minuti)

Campo	Descrizione	Tipo	Richiesto	Valore predefinito
onFailure	Specifica cosa deve fare la fase in caso di errore. I valori validi sono: • Interrompi: fallisce il passaggio dopo il numero massimo di tentativi e interrompe l'esecuzione. Imposta lo stato della fase e del documento su. `Failed` • Continua: fallisce il passaggio dopo il numero massimo di tentativi e continua a eseguire i passaggi rimanenti. Imposta lo stato della fase e del	Stringa	No	Interruzione

Campo	Descrizione	Tipo	Richiesto	Valore predefinito
	documento su`Failed`. • Ignora: imposta il passaggio `IgnoredFailure` successivo al numero massimo di tentativi falliti e continua a eseguire i passaggi rimanenti. Imposta lo stato per la fase e il documento su`SuccessWithIgnoredFailure`.			
maxAttempts	Numero massimo di tentativi consentiti prima di fallire il passaggio.	Numero intero	No	1

Campo	Descrizione	Tipo	Richiesto	Valore predefinito
inputs	Contiene i parametri richiesti dal modulo di azione per eseguire il passaggio.	Dict	Sì	

Esempi di documenti

Gli esempi seguenti mostrano i documenti dei AWSTOE componenti che eseguono attività per il sistema operativo di destinazione.

Linux

Esempio 1: eseguire un file binario personalizzato

Di seguito è riportato un documento di esempio che scarica ed esegue un file binario personalizzato su un'istanza Linux.

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

```
        action: ExecuteBinary
        onFailure: Continue
        inputs:
          path: '{{ build.Download.inputs[0].destination }}'
          arguments:
            - '--install'
      - name: Delete
        action: DeleteFile
        inputs:
          - path: '{{ build.Download.inputs[0].destination }}'
```

Windows

Esempio 1: installare gli aggiornamenti di Windows

Di seguito è riportato un documento di esempio che installa tutti gli aggiornamenti di Windows disponibili, esegue uno script di configurazione, convalida le modifiche prima della creazione dell'AMI e verifica le modifiche dopo la creazione dell'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}}'
```

Esempio 2: installalo AWS CLI su un'istanza di Windows

Di seguito è riportato un documento di esempio che installa AWS CLI su un'istanza di Windows, utilizzando il file di installazione.

```
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 }}'
```

Esempio 3: installare il file AWS CLI con il programma di installazione MSI

Di seguito è riportato un documento di esempio che installa il AWS CLI con il programma di installazione MSI.

```
name: InstallCLIMSI
description: Install &CLI; using the MSI installer
schemaVersion: 1.0
phases:
  - name: build
    steps:
      - name: Download
        action: S3Download
        inputs:
          - source: s3://aws-cli/AWSCLI64PY3.msi
            destination: C:\Windows\temp\AWSCLI64PY3.msi
      - name: Install
        action: ExecuteBinary
        onFailure: Continue
        inputs:
          path: 'C:\Windows\System32\msiexec.exe'
          arguments:
            - '/i'
            - '{{ build.Download.inputs[0].destination }}'
            - '/quiet'
            - '/norestart'
      - name: Delete
        action: DeleteFile
        inputs:
          - path: '{{ build.Download.inputs[0].destination }}'
```

macOS

Esempio 1: eseguire un file binario macOS personalizzato

Di seguito è riportato un documento di esempio che scarica ed esegue un file binario personalizzato su un'istanza macOS.

```
name: macOSBin
```

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

Usa le variabili nel tuo documento componente personalizzato

Le variabili forniscono un modo per etichettare i dati con nomi significativi che possono essere utilizzati in un'applicazione. È possibile definire variabili personalizzate con formati semplici e leggibili per flussi di lavoro complessi e farvi riferimento nel documento del componente dell'applicazione YAML per un componente. AWSTOE

Questa sezione fornisce informazioni per aiutarvi a definire le variabili per il AWSTOE componente nel documento del componente dell'applicazione YAML, tra cui sintassi, vincoli di nome ed esempi.

Costanti

Le costanti sono variabili immutabili che non possono essere modificate o sovrascritte una volta definite. Le costanti possono essere definite utilizzando i valori nella sezione di un documento. `constants` AWSTOE

Regole per i nomi delle costanti

- La lunghezza del nome deve essere compresa tra 3 e 128 caratteri.
- Il nome può contenere solo caratteri alfanumerici (a-z, A-Z, 0-9), trattini (-) o caratteri di sottolineatura (_).
- Il nome deve essere univoco all'interno del documento.
- Il nome deve essere specificato come stringa YAML.

Sintassi

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

Nome chiave	Richiesto	Descrizione
`name`	Sì	Nome della costante. Deve essere univoco per il documento (non deve essere uguale a nessun altro nome di parametro o costante).
`value`	Sì	Valore della costante.
`type`	Sì	Tipo di costante. Il tipo supportato è `string`.

Valori costanti di riferimento in un documento

Puoi fare riferimento alle costanti negli input step o loop all'interno del tuo documento YAML, come segue:

- I riferimenti alle costanti fanno distinzione tra maiuscole e minuscole e il nome deve corrispondere esattamente.
- Il nome deve essere racchiuso tra parentesi doppie arricciate. `{{ MyConstant }}`
- Gli spazi all'interno delle parentesi arricciate sono consentiti e vengono tagliati automaticamente. Ad esempio, tutti i seguenti riferimenti sono validi:

  `{{ MyConstant }}`, `{{ MyConstant}}`, `{{MyConstant }}`, `{{MyConstant}}`
- Il riferimento nel documento YAML deve essere specificato come stringa (racchiusa tra virgolette singole o doppie).

  Ad esempio: non `- {{ MyConstant }}` è valido, in quanto non è identificato come stringa.

  Tuttavia, i seguenti riferimenti sono entrambi validi: `- '{{ MyConstant }}'` e `- "{{ MyConstant }}"`.

Esempi

Costante a cui si fa riferimento negli ingressi Step

```
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'
```

Costante referenziata negli ingressi in loop

```
name: PingHosts
schemaVersion: 1.0
constants:
  - Hosts:
```

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

Parametri

I parametri sono variabili mutabili, con impostazioni che l'applicazione chiamante può fornire in fase di esecuzione. È possibile definire i parametri nella `Parameters` sezione del documento YAML.

Regole per i nomi dei parametri

- La lunghezza del nome deve essere compresa tra 3 e 128 caratteri.
- Il nome può contenere solo caratteri alfanumerici (a-z, A-Z, 0-9), trattini (-) o caratteri di sottolineatura (_).
- Il nome deve essere univoco all'interno del documento.
- Il nome deve essere specificato come stringa YAML.

Sintassi

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

Nome chiave	Richiesto	Descrizione
`name`	Sì	Nome del parametro. Deve essere univoco per il

Nome chiave	Richiesto	Descrizione
		documento (non deve essere uguale a nessun altro nome di parametro o costante).
`type`	Sì	Il tipo di dati del parametro . I tipi supportati includono :`string`.
`default`	No	Il valore predefinito per il parametro.
`description`	No	Descrive il parametro.

Valori dei parametri di riferimento in un documento

Puoi fare riferimento ai parametri negli input step o loop all'interno del tuo documento YAML, come segue:

- I riferimenti ai parametri fanno distinzione tra maiuscole e minuscole e il nome deve corrispondere esattamente.
- Il nome deve essere racchiuso tra parentesi graffe doppie. `{{ MyParameter }}`
- Gli spazi all'interno delle parentesi arricciate sono consentiti e vengono tagliati automaticamente. Ad esempio, tutti i seguenti riferimenti sono validi:

  `{{ MyParameter }}`, `{{ MyParameter}}`, `{{MyParameter }}`, `{{MyParameter}}`
- Il riferimento nel documento YAML deve essere specificato come stringa (racchiusa tra virgolette singole o doppie).

  Ad esempio: non `- {{ MyParameter }}` è valido, in quanto non è identificato come stringa.

  Tuttavia, i seguenti riferimenti sono entrambi validi: `- '{{ MyParameter }}'` e`- "{{ MyParameter }}"`.

Esempi

Gli esempi seguenti mostrano come utilizzare i parametri nel documento YAML:

- Fate riferimento a un parametro in Step Inputs:

```
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'
```

- Fate riferimento a un parametro negli ingressi del loop:

```
name: PingHosts
schemaVersion: 1.0
parameters:
  - Hosts:
      type: string
      default: 127.0.0.1,amazon.com
      description: A comma separated list of hosts to ping.
phases:
  - name: build
    steps:
      - name: Ping
        action: ExecuteBash
        loop:
          forEach:
            list: '{{ Hosts }}'
            delimiter: ','
        inputs:
          commands:
            - ping -c 4 {{ loop.value }}
```

Sostituisci i parametri in fase di esecuzione

È possibile utilizzare l'`--parameters`opzione AWS CLI con una coppia chiave-valore per impostare il valore di un parametro in fase di esecuzione.

- <name><value>Specificate la coppia chiave-valore del parametro come nome e valore, separati da un segno di uguale (=).
- Più parametri devono essere separati da una virgola.
- I nomi dei parametri che non si trovano nel documento del componente YAML vengono ignorati.
- Il nome e il valore del parametro sono entrambi obbligatori.

> **Important**
>
> I parametri dei componenti sono valori di testo semplice e vengono registrati. AWS CloudTrail Ti consigliamo di utilizzare AWS Secrets Manager o il AWS Systems Manager Parameter Store per archiviare i tuoi segreti. Per ulteriori informazioni su Secrets Manager, vedi Cos'è Secrets Manager? nella Guida AWS Secrets Manager per l'utente. Per ulteriori informazioni su AWS Systems Manager Parameter Store, vedere AWS Systems Manager Parameter Store nella Guida AWS Systems Manager per l'utente.

Sintassi

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

Opzione CLI	Richiesto	Descrizione
--parameters *name* =*value*,...	No	Questa opzione accetta l'elenco delle coppie chiave-valore, con il nome del parametro come chiave.

Esempi

Gli esempi seguenti mostrano come utilizzare i parametri nel documento YAML:

- La coppia chiave-valore del parametro specificata in questa `--parameter` opzione non è valida:

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

- Imposta una coppia chiave-valore di parametro con l'opzione in`--parameter`: AWS CLI

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

- Imposta più coppie chiave-valore di parametri con l'opzione in`--parameter`: AWS CLI

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

Usa i parametri del Parameter Store di Systems Manager

È possibile fare riferimento ai parametri AWS Systems Manager Parameter Store (parametri SSM) nei documenti dei componenti aggiungendo alle variabili il prefisso. `aws:ssm` Ad esempio,

`{{ aws:ssm:/my/param }}`si risolve al valore del parametro SSM. `/my/param`

Questa funzionalità supporta i seguenti tipi di parametri SSM:

- Stringa: esegue il mapping al tipo di AWSTOE stringa.
- StringList — Esegue il mapping al AWSTOE `stringList` tipo.
- SecureString — Esegue il mapping al tipo di AWSTOE stringa.

Per ulteriori informazioni sull'archivio dei parametri, vedere AWS Systems Manager Parameter Store nella Guida AWS Systems Manager per l'utente.

Puoi anche fare riferimento ai AWS Secrets Manager segreti utilizzando un parametro `SecureString` SSM. Ad esempio: `{{ aws:ssm:/aws/reference/secretsmanager/test/test-secret }}`. Per ulteriori informazioni, vedere Riferimento ai AWS Secrets Manager segreti dai parametri del Parameter Store.

> **Important**
>
> Image Builder esclude la risoluzione `SecureString` dei parametri dai suoi registri. Tuttavia, l'utente è anche responsabile di garantire che le informazioni riservate non vengano registrate

tramite i comandi immessi nel documento del componente. Ad esempio, se si utilizza il `echo` comando con una stringa sicura, il comando scrive un valore in chiaro nel registro.

Autorizzazioni IAM richieste

Per utilizzare i parametri Systems Manager nei componenti, il ruolo dell'istanza deve disporre dell'`ssm:GetParameter`autorizzazione per la risorsa parametrica ARN. Per esempio:

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

Per accedere ai valori crittografati, sono inoltre necessarie le seguenti autorizzazioni:

- Aggiungi `kms:Decrypt` per `SecureString` parametri o AWS Secrets Manager valori crittografati con un servizio gestito AWS KMS key dal cliente.
- Aggiungi `secretsmanager:GetSecretValue` se fai riferimento a un segreto di Secrets Manager.

Fai riferimento a un parametro SSM in un documento componente

L'esempio seguente mostra come fare riferimento a un parametro Systems Manager Parameter Store dei parametri di Systems Manager in un componente:

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

phases:
  - name: verify
    steps:
```

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

Risoluzione variabile dinamica di runtime per i parametri SSM

AWSTOE fornisce la seguente funzione integrata che è possibile utilizzare all'interno dei riferimenti alle variabili per manipolare o trasformare i valori in fase di esecuzione.

funzione di risoluzione

La `resolve` funzione risolve un riferimento a una variabile all'interno di un altro riferimento a una variabile, consentendo il riferimento dinamico al nome della variabile. Ciò è utile quando si lavora con parametri SSM in cui parte del percorso del parametro può essere variabile e passata come parametro del documento.

La `resolve` funzione supporta solo la risoluzione dinamica della parte relativa al nome di un parametro SSM.

Sintassi

L'`dynamic_variable`esempio seguente rappresenta il nome di un parametro SSM e deve essere uno dei seguenti:

- Un riferimento a un parametro SSM (ad esempio,) `aws:ssm:`*`/my/param`*
- Un riferimento a un parametro del documento componente (ad esempio,*`parameter-name`*)

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

Esempio: risolvi un parametro SSM in fase di esecuzione

L'esempio seguente mostra come utilizzare la `resolve` funzione in un documento del componente YAML:

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

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

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

Usa costrutti condizionali in AWSTOE

I costrutti condizionali eseguono azioni diverse nel documento del componente a seconda che l'espressione condizionale specificata restituisca o meno. `true` `false` È possibile utilizzare il `if` costrutto per controllare il flusso di esecuzione nel documento componente.

if Construct

È possibile utilizzare il `if` costrutto per valutare se un passaggio deve essere eseguito o meno. Per impostazione predefinita, quando l'espressione `if` condizionale restituisce`true`, AWSTOE esegue il passaggio e quando la condizione restituisce a`false`, AWSTOE salta il passaggio. Se un passaggio viene saltato, viene considerato come un passaggio riuscito quando si AWSTOE valuta se la fase e il documento sono stati eseguiti correttamente.

> **Note**
>
> Un'`if`istruzione viene valutata solo una volta, anche se il passaggio attiva un riavvio. Se un passaggio si riavvia, riconosce che l'`if`istruzione è già stata valutata e continua da dove era stata interrotta.

Sintassi

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

```
      [else: <step action>]
```

Nome chiave	Richiesto	Descrizione
espressione condizionale	Sì	L'espressione condizionale può contenere esattamente uno dei seguenti tipi di operatori al livello superiore. • Operatore di confronto: per un elenco degli operatori di confronto e informazioni sul loro funzionamento nei documenti AWSTOE dei componenti, vedereOperatori di confronto. • Operatore logico: gli operatori logici includono `and` `or` `not`, e e operano su uno o più operatori di confronto. Per ulteriori informazioni sul funzionamento degli operatori logici nei documenti dei AWSTOE componenti, vedereOperatori logici. Se l'espressione deve soddisfare più condizioni, utilizzate un operatore logico per specificare le condizioni.
poi	No	

Nome chiave	Richiesto	Descrizione
		Definisce l'azione da intrapren dere se l'espressione condizionale restituisce. `true`
altro	No	Definisce l'azione da intrapren dere se l'espressione condizionale restituisce un risultato. `false`
azione graduale	Condizionale	Quando si utilizza `then` o`else`, è necessario specifica re una delle seguenti azioni di fase: • Interrompi: AWSTOE contrassegna il passaggio come non riuscito. • Esegui: AWSTOE esegue il passaggio. • Ignora: AWSTOE salta il passaggio.

Esempio 1: pacchetto di installazione

I seguenti passaggi di esempio tratti da un documento relativo a un AWSTOE componente utilizzano operatori logici per testare il valore di un parametro ed eseguire i comandi appropriati del gestore di pacchetti per installare un'applicazione se il pacchetto è decompresso.

```
- name: InstallUnzipAptGet
    action: ExecuteBash
    if:
      and:
          - binaryExists: 'apt-get'
```

```
            - not:
                binaryExists: 'unzip'
      inputs:
        commands:
            - sudo apt-get update
            - sudo apt-get install -y unzip

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

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

Esempio 2: saltare un passaggio

L'esempio seguente mostra due modi per saltare un passaggio. Uno utilizza un operatore logico e l'altro utilizza un operatore di confronto con l'azione del `Skip` passaggio.

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

```
      - echo "Hello world" > '/etc/my_config'

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

Usa gli operatori di confronto nei documenti AWSTOE dei componenti

È possibile utilizzare i seguenti operatori di confronto con il modulo di Affermare azione e con le espressioni condizionali che utilizzano ilif Construct. Un operatore di confronto può operare su un singolo valore, ad esempio`stringIsEmpty`, oppure confrontare un valore di base con un secondo valore (valore variabile) per determinare se l'espressione condizionale restituisce o. `true` `false`

Se il confronto opera su due valori, il secondo valore può essere una variabile di concatenamento.

Quando si confrontano valori di tipo diverso, prima del confronto possono verificarsi le seguenti conversioni di valore:

- Per i confronti numerici, se il valore della variabile è una stringa, AWSTOE converte la stringa in un numero prima della valutazione. Se la conversione non è possibile, viene restituito il confronto. `false` Ad esempio, se il valore della variabile è`"1.0"`, la conversione funziona, ma se il valore della variabile è `"a10"` la conversione fallisce.
- Per i confronti tra stringhe, se il valore della variabile è un numero, lo AWSTOE converte in una stringa prima della valutazione.

Confronta le stringhe

I seguenti operatori di confronto utilizzano le stringhe per confrontare valori, verificare la presenza di spazi o una stringa vuota o confrontare un valore di input con un modello regex. I confronti tra stringhe non fanno distinzione tra maiuscole e minuscole e non tagliano gli spazi dall'inizio o dalla fine degli input di stringa.

Operatori di confronto tra stringhe

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

stringIsEmpty

L'`stringIsEmpty`operatore restituisce `true` se la stringa specificata non contiene caratteri. Per esempio:

```
# Evaluates to true
stringIsEmpty: ""

# Evaluates to false
stringIsEmpty: " "

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

stringIsWhitespace

Verifica se la stringa specificata per `stringIsWhitespace` contiene solo spazi. Per esempio:

```
# Evaluates to true
stringIsWhitespace: "   "

# Evaluates to false
stringIsWhitespace: ""

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

StringEquals

Verifica se la stringa specificata per `stringEquals` corrisponde esattamente alla stringa specificata nel parametro. `value` Per esempio:

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

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

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

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

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

stringLessThan

Verifica se la stringa specificata per `stringLessThan` è inferiore alla stringa specificata nel `value` parametro. Per esempio:

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

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

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

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

stringLessThanUguale a

Verifica se la stringa specificata per `stringLessThanEquals` è minore o uguale alla stringa specificata nel `value` parametro. Per esempio:

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

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

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

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

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

stringGreaterThan

Verifica se la stringa specificata per `stringGreaterThan` è maggiore della stringa specificata nel `value` parametro. Per esempio:

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

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

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

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

stringGreaterThanUguale a

Verifica se la stringa specificata per `stringGreaterThanEquals` è maggiore o uguale alla stringa specificata nel `value` parametro. Per esempio:

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

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

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

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

Pattern Matches

Verifica se la stringa specificata nel `value` parametro corrisponde al modello regex specificato per. `patternMatches` Il confronto utilizza il pacchetto Golang regexp, che è conforme alla sintassi. RE2 Per ulteriori informazioni sulle RE2 regole, consulta il repository google/re2 in. GitHub

L'esempio seguente mostra un pattern match che restituisce: `true`

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

Confronta i numeri

I seguenti operatori di confronto funzionano con i numeri. I valori forniti per questi operatori devono essere di uno dei seguenti tipi, in base alla specifica YAML. Il supporto per i confronti numerici utilizza l'operatore di confronto di pacchetti golang big, ad esempio: func (*Float) Cmp.

- Numero intero
- Float (basato su float64, che supporta numeri da -1.7e+308 a +1.7e+308)
- Una stringa che corrisponde al seguente schema regex: `^[-+]?([0-9]+[.])?[0-9]+$`

Operatori di confronto numerico

- numberEquals
- numberLessThan
- numberLessThanEquals
- numberGreaterThan
- numberGreaterThanEquals

Numero uguale a

Verifica se il numero specificato per `numberEquals` è uguale al numero specificato nel parametro. `value` Tutti i seguenti esempi di confronto restituiscono`true`:

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

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

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

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

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

numberLessThan

Verifica se il numero specificato per `numberLessThan` è inferiore al numero specificato nel `value` parametro. Per esempio:

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

# Evaluates to true
numberLessThan: 2
value: 1.9

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

numberLessThanUguale a

Verifica se il numero specificato per `numberLessThanEquals` è inferiore o uguale al numero specificato nel `value` parametro. Per esempio:

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

# Evaluates to true
numberLessThanEquals: 2
value: 1.9

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

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

numberGreaterThan

Verifica se il numero specificato per `numberGreaterThan` è maggiore del numero specificato nel `value` parametro. Per esempio:

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

# Evaluates to true
numberGreaterThan: 1
value: 1.1

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

numberGreaterThanUguale a

Verifica se il numero specificato per `numberGreaterThanEquals` è maggiore o uguale al numero specificato nel `value` parametro. Per esempio:

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

# Evaluates to true
numberGreaterThanEquals: 1
value: 1.1

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

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

Controlla i file

I seguenti operatori di confronto controllano l'hash del file o controllano se esiste un file o una cartella.

Operatori di file e cartelle

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

Il binario esiste

Verifica se un'applicazione è disponibile nel percorso corrente. Per esempio:

```
binaryExists: 'foo'
```

> **Note**
>
> Sui sistemi Linux e macOS, per un'applicazione denominata*foo*, funziona come il seguente comando bash:type ***foo*** >/dev/null 2>&1, dove $? == 0 indica un confronto riuscito.
> Sui sistemi Windows, per un'applicazione denominata*foo*, funziona come il PowerShell comando & C:\Windows\System32\where.exe /Q ***foo*** dove $LASTEXITCODE = 0 indica un confronto riuscito.

Il file esiste

Verifica se un file esiste nel percorso specificato. È possibile fornire un percorso assoluto o relativo. Se la posizione specificata esiste ed è un file, il confronto restituisce. `true` Per esempio:

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

> **Note**
>
> Sui sistemi Linux e macOS, funziona come il seguente comando bash:-d ***/path/to/file***, dove $? == 0 indica un confronto riuscito.

Sui sistemi Windows, funziona come il PowerShell comando. Test-Path -Path '***C:\path\to\file***' -PathType 'Leaf'

La cartella esiste

Verifica se esiste una cartella nel percorso specificato. È possibile fornire un percorso assoluto o relativo. Se la posizione specificata esiste ed è una cartella, il confronto restituisce. `true` Per esempio:

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

> **Note**
>
> Sui sistemi Linux e macOS, funziona come il seguente comando bash:-d ***/path/to/folder***, dove $? == 0 indica un confronto riuscito.
> Sui sistemi Windows, funziona come il PowerShell comando. Test-Path -Path '***C:\path\to\folder***' -PathType 'Container'

file è uguale MD5 a

Verifica se l' MD5 hash di un file è uguale a un valore specificato. Per esempio:

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

file è uguale a SHA1

Verifica se l' SHA1 hash di un file è uguale a un valore specificato. Per esempio:

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

file è uguale a SHA256

Verifica se l' SHA256 hash di un file è uguale a un valore specificato. Per esempio:

```
fileSHA256Equals: '<SHA256Hash>'
```

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

file è uguale a SHA512

Verifica se l' SHA512 hash di un file è uguale a un valore specificato. Per esempio:

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

Utilizzo di operatori logici nei documenti AWSTOE dei componenti

È possibile utilizzare i seguenti operatori logici per aggiungere o modificare espressioni condizionali nel documento componente. AWSTOE valuta le espressioni condizionali nell'ordine in cui sono specificate le condizioni. Per ulteriori informazioni sugli operatori di confronto per i documenti dei componenti, vedere. Usa gli operatori di confronto nei documenti AWSTOE dei componenti

e

Con l'`and`operatore, è possibile valutare due o più confronti come un'unica espressione. L'espressione restituisce `true` quando tutte le condizioni dell'elenco sono vere. In caso contrario, l'espressione restituisce. `false`

Esempi:

L'esempio seguente esegue due confronti: una stringa e un numero. Entrambi i confronti sono veri, quindi l'espressione restituisce vero.

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

L'esempio seguente esegue anche due confronti. Il primo confronto è falso, a quel punto la valutazione si interrompe e il secondo confronto viene saltato. L'espressione restituisce. `false`

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

```
    value: 1
```

oppure

Con l'`or`operatore, è possibile valutare due o più confronti come un'unica espressione. L'espressione restituisce `true` quando uno dei confronti specificati è vero. Se nessuno dei confronti specificati restituisce un risultato positivo`true`, l'espressione restituisce lo stesso risultato. `false`

Esempi:

L'esempio seguente esegue due confronti: una stringa e un numero. Il primo confronto è vero, quindi l'espressione restituisce un risultato `true` e il secondo confronto viene saltato.

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

L'esempio seguente esegue anche due confronti. Il primo confronto è falso e la valutazione continua. Il secondo confronto è vero, quindi l'espressione restituisce come. `true`

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

Nell'ultimo esempio, entrambi i confronti sono falsi, quindi l'espressione restituisce come. `false`

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

non

Con l'`not`operatore, puoi annullare un singolo confronto. L'espressione restituisce `true` se il confronto è falso. Se il confronto è vero, l'espressione restituisce. `false`

Esempi:

L'esempio seguente esegue un confronto tra stringhe. Il confronto è falso, quindi l'espressione restituisce come risultato. `true`

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

L'esempio seguente esegue anche un confronto tra stringhe. Il confronto è vero, quindi l'espressione restituisce. `false`

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

Usa costrutti a ciclo continuo in AWSTOE

Questa sezione fornisce informazioni per aiutarvi a creare costrutti ciclici in. AWSTOE I costrutti di looping definiscono una sequenza ripetuta di istruzioni. È possibile utilizzare i seguenti tipi di costrutti di loop in: AWSTOE

- `for`costrutti: esegue l'iterazione su una sequenza limitata di numeri interi.
- `forEach`costrutti
  - `forEach`loop with input list — Itera su una raccolta finita di stringhe.
  - `forEach`loop con elenco delimitato — Itera su una raccolta finita di stringhe unite da un delimitatore.

> **Note**
>
> I costrutti di looping supportano solo i tipi di dati di tipo stringa.

Argomenti relativi ai costrutti in loop

- Variabili di iterazione di riferimento
- Tipi di costrutti a ciclo continuo

- Campi Step
- Uscite di fase e iterazione

Variabili di iterazione di riferimento

Per fare riferimento all'indice e al valore della variabile di iterazione corrente, l'espressione di riferimento `{{ loop.* }}` deve essere utilizzata all'interno del corpo di input di un passaggio che contiene un costrutto ciclico. Questa espressione non può essere utilizzata per fare riferimento alle variabili di iterazione del costrutto di looping di un altro passaggio.

L'espressione di riferimento è composta dai seguenti membri:

- `{{ loop.index }}`— La posizione ordinale dell'iterazione corrente, indicizzata in. `0`
- `{{ loop.value }}`— Il valore associato alla variabile di iterazione corrente.

Nomi dei loop

Tutti i costrutti in loop hanno un campo nome opzionale per l'identificazione. Se viene fornito un nome di ciclo, può essere utilizzato per fare riferimento alle variabili di iterazione nel corpo di input del passaggio. Per fare riferimento agli indici di iterazione e ai valori di un ciclo denominato, utilizzate `{{ <loop_name>.* }}` with `{{ loop.* }}` nel corpo di input del passaggio. Questa espressione non può essere utilizzata per fare riferimento al costrutto di looping denominato di un altro passaggio.

L'espressione di riferimento è composta dai seguenti membri:

- `{{ <loop_name>.index }}`— La posizione ordinale dell'iterazione corrente del ciclo denominato, indicizzata in. `0`
- `{{ <loop_name>.value }}`— Il valore associato alla variabile di iterazione corrente del ciclo denominato.

Risolve le espressioni di riferimento

AWSTOE Risolve le espressioni di riferimento come segue:

- `{{ <loop_name>.* }}`— AWSTOE risolve questa espressione utilizzando la logica seguente:
  - Se il ciclo del passaggio attualmente in esecuzione corrisponde al `<loop_name>` valore, l'espressione di riferimento si risolve nel costrutto di loop del passaggio attualmente in esecuzione.

- `<loop_name>`si risolve nel costrutto di looping denominato se appare nella fase attualmente in esecuzione.
- `{{ loop.* }}`— AWSTOE risolve l'espressione utilizzando il costrutto di looping definito nella fase attualmente in esecuzione.

Se le espressioni di riferimento vengono utilizzate all'interno di un passaggio che non contiene un ciclo, non AWSTOE risolve le espressioni e vengono visualizzate nel passaggio senza alcuna sostituzione.

> **Note**
>
> Le espressioni di riferimento devono essere racchiuse tra virgolette doppie per essere interpretate correttamente dal compilatore YAML.

Tipi di costrutti a ciclo continuo

Questa sezione fornisce informazioni ed esempi sui tipi di costrutti di looping che possono essere utilizzati in. AWSTOE

Tipi di costrutti a ciclo continuo

- forciclo
- forEachloop con elenco di input
- forEachciclo con elenco delimitato

`for`ciclo

Il `for` ciclo itera su un intervallo di numeri interi specificato all'interno di un limite delineato dall'inizio e dalla fine delle variabili. I valori di iterazione sono inclusi nel set `[start, end]` e includono i valori limite.

AWSTOE verifica i `updateBy` valori `startend`, e per garantire che la combinazione non produca un ciclo infinito.

`for`schema ad anello

```
  - name: "StepName"
```

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

foringresso in loop

Campo	Descrizione	Tipo	Obbligatorio	Predefinita
`name`	Nome univoco del ciclo. Deve essere univoco rispetto ad altri nomi di loop nella stessa fase.	Stringa	No	""
`start`	Valore iniziale dell'iterazione. Non accetta espressioni concatenate.	Numero intero	Sì	N/A
`end`	Valore finale dell'iterazione. Non accetta espressioni concatenate.	Numero intero	Sì	N/A
`updateBy`	Differenza in base alla quale un valore iterativo viene aggiornato tramite addizione	Numero intero	Sì	N/A

Campo	Descrizione	Tipo	Obbligatorio	Predefinita
	. Deve essere un valore negativo o positivo diverso da zero. Non accetta espressioni concatenate.			

`for`esempio di input in loop

```
  - name: "CalculateFileUploadLatencies"
    action: "ExecutePowerShell"
    loop:
      for:
        start: 100000
        end: 1000000
        updateBy: 100000
    inputs:
      commands:
        - |
          $f = new-object System.IO.FileStream c:\temp\test{{ loop.index }}.txt,
 Create, ReadWrite
          $f.SetLength({{ loop.value }}MB)
          $f.Close()
        - c:\users\administrator\downloads\latencyTest.exe --file c:\temp
\test{{ loop.index }}.txt
        - AWS s3 cp c:\users\administrator\downloads\latencyMetrics.json s3://bucket/
latencyMetrics.json
        - |
          Remove-Item -Path c:\temp\test{{ loop.index }}.txt
          Remove-Item -Path c:\users\administrator\downloads\latencyMetrics.json
```

`forEach`loop con elenco di input

Il `forEach` ciclo itera su un elenco esplicito di valori, che può essere costituito da stringhe ed espressioni concatenate.

`forEach`ciclo con schema di elenco di input

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

`forEach`loop con immissione della lista di input

Campo	Descrizione	Tipo	Obbligatorio	Predefinita
`name`	Nome univoco del ciclo. Deve essere univoco rispetto ad altri nomi di loop nella stessa fase.	Stringa	No	""
Elenco di stringhe di loop `forEach`	Elenco di stringhe per l'iterazione. Accetta espressioni concatenate come stringhe nell'elenco. Le espressioni concatenate devono essere racchiuse tra virgolette doppie affinché il compilatore YAML le	Elenco di stringhe	Sì	N/A

Campo	Descrizione	Tipo	Obbligatorio	Predefinita
	interpreti correttamente.			

`forEach`ciclo con elenco di input (esempio 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`ciclo con elenco di input (esempio 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`ciclo con elenco di input (esempio 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`ciclo con elenco delimitato

Il ciclo scorre su una stringa contenente valori separati da un delimitatore. Per iterare sui componenti della stringa, AWSTOE utilizza il delimitatore per dividere la stringa in un array adatto all'iterazione.

`forEach`ciclo con schema a elenco delimitato

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

`forEach`ciclo con inserimento di un elenco delimitato

Campo	Descrizione	Tipo	Obbligatorio	Predefinita
`name`	Nome univoco assegnato al ciclo. Dovrebbe essere unico	Stringa	No	""

Campo	Descrizione	Tipo	Obbligatorio	Predefinita
	se confrontato con altri nomi di loop nella stessa fase.			
`list`	Una stringa composta da stringhe costituenti unite da un carattere delimitatore comune. Accetta anche espressioni concatenate. In caso di espressioni concatenate, assicuratevi che siano racchiuse tra virgolette doppie per una corretta interpretazione da parte del compilatore YAML.	Stringa	Sì	N/A

Campo	Descrizione	Tipo	Obbligatorio	Predefinita
`delimiter`	Carattere usato per separare le stringhe all'interno di un blocco. L'impostazione predefinita è il carattere virgola. È consentito un solo carattere delimitatore dall'elenco fornito: • Punto: `"."` • Virgola: `","` • Punto e virgola: `";"` • Colon: `":"` • Nuova riga: `"\n"` • Scheda: `"\t"` • Spazio: `" "` • Trattino: `"-"` • Sottolineatura: `"_"`	Stringa	No	Virgola: `","`

Campo	Descrizione	Tipo	Obbligatorio	Predefinita
	Le espressioni concatenate non possono essere utilizzate.			

> **Note**
>
> Il valore di `list` viene considerato come una stringa immutabile. Se l'origine di `list` viene modificata durante l'esecuzione, non verrà riflessa durante l'esecuzione.

`forEach`ciclo con elenco delimitato (esempio 1)

Questo esempio utilizza il seguente modello di espressione concatenata per fare riferimento all'output di un altro passaggio:. `<phase_name>.<step_name>.[inputs | outputs].<var_name>`

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

`forEach`ciclo con elenco delimitato (esempio 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 }}"
```

Campi Step

I loop fanno parte di un passaggio. Qualsiasi campo relativo all'esecuzione di un passaggio non viene applicato alle singole iterazioni. I campi della fase si applicano solo a livello di fase, come segue:

- TimeoutSeconds: tutte le iterazioni del ciclo devono essere eseguite entro il periodo di tempo specificato in questo campo. Se l'esecuzione del ciclo scade, AWSTOE esegue la politica di riprova del passaggio e reimposta il parametro di timeout per ogni nuovo tentativo. Se l'esecuzione del ciclo supera il valore di timeout dopo aver raggiunto il numero massimo di tentativi, il messaggio di errore della fase indica che l'esecuzione del ciclo è scaduta.
- OnFailure: la gestione degli errori viene applicata alla fase nel modo seguente:
  - Se OnFailure è impostato su`Abort`, AWSTOE esce dal ciclo e riprova il passaggio in base alla politica di ripetizione dei tentativi. Dopo il numero massimo di tentativi, AWSTOE contrassegna il passaggio corrente come fallito e interrompe l'esecuzione del processo.

    AWSTOE imposta il codice di stato per la fase principale e il documento su`Failed`.

    > **Note**
    >
    > Nessun altro passaggio viene eseguito dopo il passaggio fallito.

  - Se OnFailure è impostato su`Continue`, AWSTOE esce dal ciclo e riprova il passaggio in base alla politica di ripetizione dei tentativi. Dopo il numero massimo di tentativi, AWSTOE contrassegna il passaggio corrente come fallito e continua a eseguire il passaggio successivo.

    AWSTOE imposta il codice di stato per la fase principale e il documento su`Failed`.
  - Se OnFailure è impostato su`Ignore`, AWSTOE esce dal ciclo e riprova il passaggio in base alla politica di ripetizione dei tentativi. Dopo il numero massimo di tentativi, AWSTOE contrassegna il passaggio corrente come `IgnoredFailure` e continua a eseguire il passaggio successivo.

    AWSTOE imposta il codice di stato per la fase principale e il documento su`SuccessWithIgnoredFailure`.

    > **Note**
    >
    > Viene comunque considerata un'esecuzione riuscita, ma include informazioni che indicano che uno o più passaggi non sono riusciti e sono stati ignorati.

- MaxAttempts: per ogni nuovo tentativo, l'intero passaggio e tutte le iterazioni vengono eseguiti dall'inizio.
- status: lo stato generale dell'esecuzione di un passaggio. `status`non rappresenta lo stato delle singole iterazioni. Lo stato di un passaggio con loop è determinato come segue:
  - Se una singola iterazione non viene eseguita, lo stato di un passaggio indica un errore.
  - Se tutte le iterazioni hanno esito positivo, lo stato di un passaggio indica un successo.
- startTime: l'ora di inizio complessiva dell'esecuzione di un passaggio. Non rappresenta l'ora di inizio delle singole iterazioni.
- EndTime: l'ora di fine complessiva dell'esecuzione di un passaggio. Non rappresenta l'ora di fine delle singole iterazioni.
- FailureMessage: include gli indici di iterazione che hanno avuto esito negativo in caso di errori diversi dal timeout. In caso di errori di timeout, il messaggio indica che l'esecuzione del ciclo non è riuscita. Non vengono forniti messaggi di errore individuali per ogni iterazione per ridurre al minimo la dimensione dei messaggi di errore.

Uscite di fase e iterazione

Ogni iterazione contiene un output. Alla fine di un ciclo, AWSTOE consolida tutti gli output di iterazione riusciti in. `detailedOutput.json` Gli output consolidati sono una raccolta di valori che appartengono alle chiavi di output corrispondenti, come definito nello schema di output del modulo di azione. L'esempio seguente mostra come vengono consolidati gli output:

Output di **`ExecuteBash`** per l'iterazione 1

```
{
 "stdout":"Hello"
}
```

Output di **`ExecuteBash`** per l'iterazione 2

```
{
 "stdout":"World"
}
```

Output di **`ExecuteBash`** for Step

```
{
```

```
 "stdout":"Hello\nWorld"
}
```

Ad esempio `ExecuteBash`ExecutePowerShell, e `ExecuteBinary` sono moduli di azione che restituiscono `STDOUT` come output del modulo di azione. `STDOUT`i messaggi vengono uniti al nuovo carattere di riga per produrre l'output complessivo dello step in`detailedOutput.json`.

AWSTOE non consoliderà i risultati delle iterazioni non riuscite.

Moduli di azione supportati dal gestore AWSTOE dei componenti

I servizi di creazione di EC2 immagini, come Image Builder, utilizzano moduli di AWSTOE azione per configurare le EC2 istanze utilizzate per creare e testare immagini di macchine personalizzate. Questa sezione descrive le funzionalità dei moduli di AWSTOE azione di uso comune e come configurarli, inclusi alcuni esempi.

I componenti sono creati con documenti YAML in testo semplice. Per ulteriori informazioni sulla sintassi dei documenti, vedere. Usa il framework AWSTOE dei documenti dei componenti per i componenti personalizzati

> **Note**
>
> Tutti i moduli di azione utilizzano lo stesso account dell'agente Systems Manager quando vengono eseguiti, ovvero `root` su Linux e `NT Authority\SYSTEM` su Windows.

Il seguente riferimento incrociato classifica i moduli di azione in base al tipo di azioni che eseguono.

Esecuzione generale

- Assert (Linux, Windows, macOS)
- ExecuteBash (Linux, macOS)
- ExecuteBinary (Linux, Windows, macOS)
- ExecuteDocument (Linux, Windows, macOS)
- ExecutePowerShell (Windows)

Scaricamento e caricamento di file

- Scarica S3 (Linux, Windows, macOS)
- Caricamento S3 (Linux, Windows, macOS)
- WebDownload (Linux, Windows, macOS)

Operazioni del file system

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

Azioni di installazione del software

- Installa MSI (Windows)
- Disinstalla MSI (Windows)

Azioni di sistema

- Riavvio (Linux, Windows)
- SetRegistry (Windows)
- UpdateOS (Linux, Windows)

Moduli di esecuzione generali

La sezione seguente contiene dettagli sui moduli di azione che eseguono comandi e controllano il flusso di lavoro di esecuzione.

Moduli di azioni generali di esecuzione

- Assert (Linux, Windows, macOS)
- ExecuteBash (Linux, macOS)
- ExecuteBinary (Linux, Windows, macOS)
- ExecuteDocument (Linux, Windows, macOS)
- ExecutePowerShell (Windows)

Assert (Linux, Windows, macOS)

Il modulo di azione Assert esegue confronti di valori utilizzando Operatori di confronto o Operatori logici come input. Il risultato dell'espressione dell'operatore (vero o falso) indica lo stato complessivo di successo o di fallimento della fase.

Se l'espressione dell'operatore logico o di confronto restituisce un risultato positivo`true`, il passaggio viene contrassegnato come`Success`. In caso contrario, il passaggio viene contrassegnato come`Failed`. Se il passaggio fallisce, il `onFailure` parametro decide il risultato del passaggio.

Input

Nome chiave	Descrizione	Tipo	Richiesto
`input`	Contiene un singolo operatore logico o di confronto. Nota,	Questo valore è variabile, a seconda dell'operatore	Sì

Nome chiave	Descrizione	Tipo	Richiesto
	gli operatori logici possono contenere più di un operatore di confronto.		

Esempio di input: un semplice confronto utilizzando l'operatore di **`stringEquals`** confronto

Questo esempio restituisce. `true`

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

Esempio di input: confronti Regex utilizzando l'operatore di confronto **`patternMatches`**

Tutti questi esempi restituiscono. `true`

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

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

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

Esempio di input: confronti annidati con operatori logici e variabili concatenate

L'esempio seguente illustra i confronti annidati con operatori logici che utilizzano confronti con variabili concatenate. Viene `Assert` restituito `true` se una delle seguenti condizioni è vera:

- Il `ApplicationVersion` è maggiore `2.0` e `CPUArchitecture` uguale `arm64` a.
- Gli `CPUArchitecture` uguali. `x86_64`

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

Output:

Il risultato di un `Assert` è il successo o il fallimento della fase.

ExecuteBash (Linux, macOS)

Il modulo ExecuteBashaction consente di eseguire script bash con codice/comandi di shell in linea. Questo modulo supporta Linux.

Tutti i comandi e le istruzioni specificati nel blocco dei comandi vengono convertiti in un file (ad esempio`input.sh`) ed eseguiti con la shell bash. Il risultato dell'esecuzione del file shell è il codice di uscita del passaggio.

Il ExecuteBashmodulo gestisce i riavvii del sistema se lo script esce con un codice di uscita di. `194` Una volta avviata, l'applicazione esegue una delle seguenti azioni:

- L'applicazione consegna il codice di uscita al chiamante se viene eseguita dall'agente Systems Manager. Systems Manager Agent gestisce il riavvio del sistema ed esegue lo stesso passaggio che ha avviato il riavvio, come descritto in Riavvio di un'istanza gestita dagli script.
- L'applicazione salva la versione corrente`executionstate`, configura un trigger di riavvio per rieseguire l'applicazione e riavvia il sistema.

Dopo il riavvio del sistema, l'applicazione esegue lo stesso passaggio che ha avviato il riavvio. Se è necessaria questa funzionalità, è necessario scrivere script idempotenti in grado di gestire più invocazioni dello stesso comando di shell.

Input

Nome chiave	Descrizione	Tipo	Richiesto
`commands`	Contiene un elenco di istruzioni o comandi da eseguire secondo la sintassi di bash. È consentito lo standard YAML multilinea.	Elenco	Sì

Esempio di input: prima e dopo il riavvio

```
name: ExitCode194Example
description: This shows how the exit code can be used to restart a system with
 ExecuteBash
schemaVersion: 1.0
phases:
  - name: build
    steps:
      - name: RestartTrigger
        action: ExecuteBash
        inputs:
          commands:
            - |
              REBOOT_INDICATOR=/var/tmp/reboot-indicator
              if [ -f "${REBOOT_INDICATOR}" ]; then
                echo 'The reboot file exists. Deleting it and exiting with success.'
                rm "${REBOOT_INDICATOR}"
                exit 0
              fi
              echo 'The reboot file does not exist. Creating it and triggering a
 restart.'
              touch "${REBOOT_INDICATOR}"
              exit 194
```

Output

Campo	Descrizione	Tipo
`stdout`	Output standard dell'esecuzione del comando.	string

Se avvii un riavvio e restituisci il codice di uscita `194` come parte del modulo di azione, la compilazione riprenderà dalla stessa fase del modulo di azione che ha avviato il riavvio. Se avvii un riavvio senza il codice di uscita, il processo di compilazione potrebbe non riuscire.

Esempio di output: prima del riavvio (prima lettura del documento)

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

Esempio di output: dopo il riavvio, (seconda volta nel documento)

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

ExecuteBinary (Linux, Windows, macOS)

Il modulo di ExecuteBinaryazione consente di eseguire file binari con un elenco di argomenti della riga di comando.

Il ExecuteBinarymodulo gestisce i riavvii del sistema se il file binario esce con un codice di uscita `194` (Linux) o `3010` (Windows). Quando ciò accade, l'applicazione esegue una delle seguenti azioni:

- L'applicazione consegna il codice di uscita al chiamante se viene eseguita dall'agente Systems Manager. Systems Manager Agent gestisce il riavvio del sistema ed esegue lo stesso passaggio che ha avviato il riavvio, come descritto in Riavvio di un'istanza gestita dagli script.
- L'applicazione salva la versione corrente`executionstate`, configura un trigger di riavvio per rieseguire l'applicazione e riavvia il sistema.

Dopo il riavvio del sistema, l'applicazione esegue lo stesso passaggio che ha avviato il riavvio. Se è necessaria questa funzionalità, è necessario scrivere script idempotenti in grado di gestire più invocazioni dello stesso comando di shell.

Input

Nome chiave	Descrizione	Tipo	Richiesto
`path`	Il percorso del file binario per l'esecuzione.	Stringa	Sì
`arguments`	Contiene un elenco di argomenti della riga di comando da utilizzare durante l'esecuzione del file binario.	Elenco di stringhe	No

Esempio di input: installare.NET

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

Output

Campo	Descrizione	Tipo
`stdout`	Output standard dell'esecuzione del comando.	string

Esempio di output

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

```
}
```

ExecuteDocument (Linux, Windows, macOS)

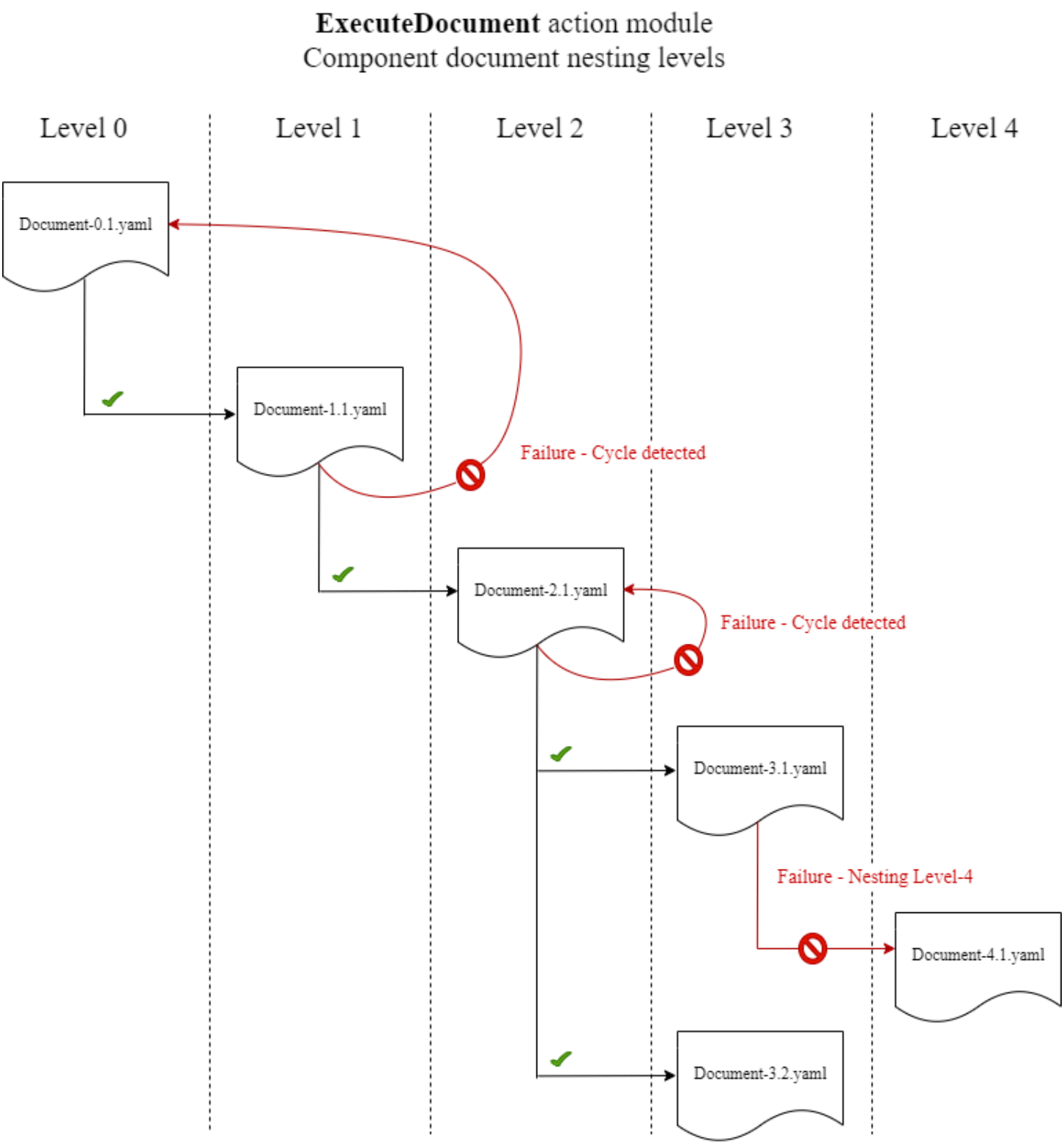
Il modulo di ExecuteDocumentazione aggiunge il supporto per i documenti di componenti annidati, eseguendo più documenti componenti da un unico documento. AWSTOE convalida il documento passato nel parametro di input in fase di esecuzione.

Restrizioni

- Questo modulo di azione viene eseguito una sola volta, senza che siano consentiti nuovi tentativi e senza alcuna opzione per impostare limiti di timeout. ExecuteDocumentimposta i seguenti valori predefiniti e restituisce un errore se si tenta di modificarli.
  - `timeoutSeconds`: -1
  - `maxAttempts`: 1

  > **Note**
  >
  > È possibile lasciare vuoti questi valori e AWSTOE utilizzare i valori predefiniti.

- L'annidamento dei documenti è consentito, fino a tre livelli di profondità, ma non di più. Tre livelli di nidificazione si traducono in quattro livelli di documento, poiché il livello superiore non è nidificato. In questo scenario, il documento di livello più basso non deve richiamare nessun altro documento.
- L'esecuzione ciclica dei documenti dei componenti non è consentita. Qualsiasi documento che richiama se stesso al di fuori di un costrutto a ciclo continuo o che richiama un altro documento più in alto nella catena di esecuzione corrente avvia un ciclo che può generare un ciclo infinito. Quando AWSTOE rileva un'esecuzione ciclica, interrompe l'esecuzione e registra l'errore.



Se un documento componente tenta di eseguirsi da solo o di eseguire uno dei documenti componenti che si trovano più in alto nella catena di esecuzione corrente, l'esecuzione fallisce.

Input

Nome chiave	Descrizione	Tipo	Richiesto
`document`	Percorso del documento componente. Le opzioni valide includono: • Percorsi dei file locali • S3 URIs • EC2 Versione di build del component e Image Builder ARNs	Stringa	Sì
`document-s3-bucket-owner`	L'ID dell'account del proprietario del bucket S3 per il bucket S3 in cui sono archiviat i i documenti dei componenti. (Consigli ato se utilizzi S3 URIs nel documento del componente.)	Stringa	No
`phases`	Fasi da eseguire nel documento del componente, espresse come elenco separato da virgole. Se non viene specificata	Stringa	No

Nome chiave	Descrizione	Tipo	Richiesto
	alcuna fase, vengono eseguite tutte le fasi.		
`parameters`	Parametri di input che vengono passati al documento del componente in fase di esecuzione come coppie chiave-valore.	Elenco delle mappe dei parametri	No

Inserimento della mappa dei parametri

Nome chiave	Descrizione	Tipo	Richiesto
`name`	Il nome del parametro di input da passare al documento del componente che il modulo di ExecuteDocumentazione è in esecuzione.	Stringa	Sì
`value`	Il valore del parametro di input.	Stringa	Sì

Esempi di input

Gli esempi seguenti mostrano le variazioni degli input per il documento del componente, a seconda del percorso di installazione.

Esempio di input: percorso del documento locale

```
# 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
```

Esempio di input: URI S3 come percorso del documento

```
# main.yaml
schemaVersion: 1.0

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

Esempio di input: ARN del componente EC2 Image Builder come percorso del documento

```
# main.yaml
schemaVersion: 1.0

phases:
  - name: build
    steps:
```

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

Utilizzo di un ForEach ciclo per eseguire documenti

```
# 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
```

Utilizzo di un ciclo For per eseguire documenti

```
# main.yaml
schemaVersion: 1.0

phases:
  - name: build
    steps:
```

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

Output

AWSTOE crea un file di output chiamato `detailedoutput.json` ogni volta che viene eseguito. Il file contiene dettagli su ogni fase e passaggio di ogni documento componente che viene richiamato durante l'esecuzione. Per il modulo ExecuteDocumentd'azione, è possibile trovare un breve riepilogo del runtime nel `outputs` campo e dettagli sulle fasi, i passaggi e i documenti in cui viene eseguito. `detailedOutput`

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

L'oggetto di riepilogo dell'output di ogni documento componente contiene i seguenti dettagli, come illustrato di seguito, con valori di esempio:

- executedStepCount«:1
- «ID esecuzione» :"12345a67-89bc-01de-2f34-abcd56789012"
- failedStepCount««:0
- «FailureMessage»: "»
- «Conteggio»: 0 ignoredFailedStep
- «logUrl»: "»

- «status»: «successo»

Esempio di output

L'esempio seguente mostra l'output del modulo di ExecuteDocumentazione quando si verifica un'esecuzione annidata. In questo esempio, il documento `main.yaml` componente esegue correttamente il documento del `Sample-1.yaml` componente.

```
{
    "executionId": "12345a67-89bc-01de-2f34-abcd56789012",
    "status": "success",
    "startTime": "2021-08-26T17:20:31-07:00",
    "endTime": "2021-08-26T17:20:31-07:00",
    "failureMessage": "",
    "documents": [
        {
            "name": "",
            "filePath": "main.yaml",
            "status": "success",
            "description": "",
            "startTime": "2021-08-26T17:20:31-07:00",
            "endTime": "2021-08-26T17:20:31-07:00",
            "failureMessage": "",
            "phases": [
                {
                    "name": "build",
                    "status": "success",
                    "startTime": "2021-08-26T17:20:31-07:00",
                    "endTime": "2021-08-26T17:20:31-07:00",
                    "failureMessage": "",
                    "steps": [
                        {
                            "name": "ExecuteNestedDocument",
                            "status": "success",
                            "failureMessage": "",
                            "timeoutSeconds": -1,
                            "onFailure": "Abort",
                            "maxAttempts": 1,
                            "action": "ExecuteDocument",
                            "startTime": "2021-08-26T17:20:31-07:00",
                            "endTime": "2021-08-26T17:20:31-07:00",
                            "inputs": "[{\"document\":\"Sample-1.yaml\",\"document-s3-
bucket-owner\":\"\",\"phases\":\"\",\"parameters\":null}]",
```

```
                            "outputs": "[{\"executedStepCount\":1,\"executionId\":
\"98765f43-21ed-09cb-8a76-fedc54321098\",\"failedStepCount\":0,\"failureMessage\":\"\",
\"ignoredFailedStepCount\":0,\"logUrl\":\"\",\"status\":\"success\"}]",
                            "loop": null,
                            "detailedOutput": [
                                {
                                    "executionId": "98765f43-21ed-09cb-8a76-
fedc54321098",
                                    "status": "success",
                                    "startTime": "2021-08-26T17:20:31-07:00",
                                    "endTime": "2021-08-26T17:20:31-07:00",
                                    "failureMessage": "",
                                    "documents": [
                                        {
                                            "name": "",
                                            "filePath": "Sample-1.yaml",
                                            "status": "success",
                                            "description": "",
                                            "startTime": "2021-08-26T17:20:31-07:00",
                                            "endTime": "2021-08-26T17:20:31-07:00",
                                            "failureMessage": "",
                                            "phases": [
                                                {
                                                    "name": "build",
                                                    "status": "success",
                                                    "startTime":
 "2021-08-26T17:20:31-07:00",
                                                    "endTime":
 "2021-08-26T17:20:31-07:00",
                                                    "failureMessage": "",
                                                    "steps": [
                                                        {
                                                            "name": "ExecuteBashStep",
                                                            "status": "success",
                                                            "failureMessage": "",
                                                            "timeoutSeconds": 7200,
                                                            "onFailure": "Abort",
                                                            "maxAttempts": 1,
                                                            "action": "ExecuteBash",
                                                            "startTime":
 "2021-08-26T17:20:31-07:00",
                                                            "endTime":
 "2021-08-26T17:20:31-07:00",
```

```
                                                                "inputs": "[{\"commands\":
[\"echo \\\"Hello World!\\\"\"]}]",
                                                                "outputs": "[{\"stdout\":
\"Hello World!\"}]",
                                                                "loop": null,
                                                                "detailedOutput": null
                                                            }]
                                                        }]
                                                    }]
                                                }]
                                            }]

                                    }]
                                }]
}
```

ExecutePowerShell (Windows)

Il modulo di ExecutePowerShellazione consente di eseguire PowerShell script con codice/comandi di shell in linea. Questo modulo supporta la piattaforma Windows e Windows. PowerShell

Tutto ciò che è commands/instructions specificato nel blocco dei comandi viene convertito in un file di script (ad esempio`input.ps1`) ed eseguito utilizzando WindowsPowerShell. Il risultato dell'esecuzione del file shell è il codice di uscita.

Il ExecutePowerShellmodulo gestisce i riavvii del sistema se il comando shell esce con un codice di uscita di. `3010` Una volta avviata, l'applicazione esegue una delle seguenti azioni:

- Fornisce il codice di uscita al chiamante se eseguito dall'agente Systems Manager. Systems Manager Agent gestisce il riavvio del sistema ed esegue lo stesso passaggio che ha avviato il riavvio, come descritto in Riavvio di un'istanza gestita dagli script.
- Salva la versione corrente`executionstate`, configura un trigger di riavvio per rieseguire l'applicazione e riavvia il sistema.

Dopo il riavvio del sistema, l'applicazione esegue lo stesso passaggio che ha avviato il riavvio. Se è necessaria questa funzionalità, è necessario scrivere script idempotenti in grado di gestire più invocazioni dello stesso comando di shell.

Input

Nome chiave	Descrizione	Tipo	Richiesto
`commands`	Contiene un elenco di istruzioni o comandi da eseguire secondo la sintassi. PowerShell È consentito il formato YAML multilinea.	Elenco di stringhe	Sì. È necessario specificare `commands` o `file` non entrambi.
`file`	Contiene il percorso di un file di PowerShell script. PowerShell verrà eseguito su questo file utilizzando l'argomento della riga di `-file` comando. Il percorso deve puntare a un `.ps1` file.	Stringa	Sì. È necessario specificare `commands` o `file` non entrambi.

Esempio di input: prima e dopo il riavvio

```
name: ExitCode3010Example
description: This shows how the exit code can be used to restart a system with
 ExecutePowerShell
schemaVersion: 1.0
phases:
  - name: build
    steps:
      - name: RestartTrigger
        action: ExecutePowerShell
        inputs:
          commands:
            - |
              $rebootIndicator = Join-Path -Path $env:SystemDrive -ChildPath 'reboot-
indicator'
```

```
            if (Test-Path -Path $rebootIndicator) {
              Write-Host 'The reboot file exists. Deleting it and exiting with
success.'
              Remove-Item -Path $rebootIndicator -Force | Out-Null
              [System.Environment]::Exit(0)
            }
            Write-Host 'The reboot file does not exist. Creating it and triggering a
restart.'
            New-Item -Path $rebootIndicator -ItemType File | Out-Null
            [System.Environment]::Exit(3010)
```

Output

Campo	Descrizione	Tipo
`stdout`	Output standard dell'esecuzione del comando.	string

Se esegui un riavvio e restituisci il codice di uscita `3010` come parte del modulo di azione, la compilazione riprenderà dalla stessa fase del modulo di azione che ha avviato il riavvio. Se esegui un riavvio senza il codice di uscita, il processo di compilazione potrebbe non riuscire.

Esempio di output: prima del riavvio (prima lettura del documento)

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

Esempio di output: dopo il riavvio, (seconda volta nel documento)

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

Moduli per il download e il caricamento dei file

La sezione seguente contiene dettagli sui moduli di azione che caricano o scaricano file.

Scarica e carica i moduli d'azione

- Scarica S3 (Linux, Windows, macOS)

- Caricamento S3 (Linux, Windows, macOS)
- WebDownload (Linux, Windows, macOS)

Scarica S3 (Linux, Windows, macOS)

Con il modulo di `S3Download` azione, puoi scaricare un oggetto Amazon S3, o un set di oggetti, in un file o una cartella locale che specifichi con il `destination` percorso. Se esiste già un file nella posizione specificata e il `overwrite` flag è impostato su true, `S3Download` sovrascrive il file.

La tua `source` posizione può puntare a un oggetto specifico in Amazon S3 oppure puoi utilizzare un prefisso chiave con un asterisco wildcard * () per scaricare un set di oggetti che corrispondono al percorso del prefisso chiave. Quando specificate un prefisso chiave nella vostra `source` posizione, il modulo di `S3Download` azione scarica tutto ciò che corrisponde al prefisso (file e cartelle inclusi). Assicurati che il prefisso della chiave termini con una barra, seguita da un asterisco (`/*`), in modo da scaricare tutto ciò che corrisponde al prefisso. Ad esempio: *`s3://my-bucket/my-folder/*`*.

Se l'`S3Download`azione per uno specifico key prefix fallisce durante un download, il contenuto della cartella non viene ripristinato allo stato precedente all'errore. La cartella di destinazione rimane com'era al momento dell'errore.

Casi di utilizzo supportati

Il modulo di `S3Download` azione supporta i seguenti casi d'uso:

- L'oggetto Amazon S3 viene scaricato in una cartella locale, come specificato nel percorso di download.
- Gli oggetti Amazon S3 (con un prefisso chiave nel percorso del file Amazon S3) vengono scaricati nella cartella locale specificata, che copia in modo ricorsivo tutti gli oggetti Amazon S3 che corrispondono al prefisso della chiave nella cartella locale.

Requisiti IAM

Il ruolo IAM associato al profilo dell'istanza deve disporre delle autorizzazioni per eseguire il modulo di `S3Download` azione. Le seguenti politiche IAM devono essere associate al ruolo IAM associato al profilo dell'istanza:

- File singolo: `s3:GetObject` contro bucket/object (ad esempio,`arn:aws:s3:::`**`BucketName`**`/*`).
- File multipli: `s3:ListBucket` contro bucket/object (ad esempio,`arn:aws:s3:::`**`BucketName`**) e `s3:GetObject` contro bucket/object (ad esempio,`arn:aws:s3:::`**`BucketName`**`/*`).

Input

Chiave	Descrizione	Tipo	Obbligatorio	Predefinita
`source`	Il bucket Amazon S3 che è la fonte per il download. Potete specifica re un percorso per un oggetto specifico o utilizzare un prefisso chiave che termina con una barra, seguita da un asterisco wildcard (`/*`), per scaricare un set di oggetti che corrispondono al prefisso della chiave.	Stringa	Sì	N/D
`destination`	Il percorso locale in cui vengono scaricati gli oggetti Amazon S3. Per scaricare un singolo file, devi specifica re il nome del file come parte del percorso.	Stringa	Sì	N/D

Chiave	Descrizione	Tipo	Obbligatorio	Predefinita
	Ad esempio, */myfolder/package.zip* .			
`expectedBucketOwner`	ID dell'account proprietario previsto del bucket fornito nel `source` percorso. Ti consigliamo di verificare la proprietà del bucket Amazon S3 specificato nel codice sorgente.	Stringa	No	N/D

Chiave	Descrizione	Tipo	Obbligatorio	Predefinita
`overwrite`	Se impostato su true, se un file con lo stesso nome esiste già nella cartella di destinazione per il percorso locale specifica to, il file scaricato sovrascrive il file locale. Se impostato su false, il file esistente nel sistema locale è protetto dalla sovrascrittura e il modulo di azione ha esito negativo e restituisce un errore di download. Ad esempio, `Error: S3Download: File already exists and "overwrit e" property for "destinat ion" file`	Boolean	No	true

Chiave	Descrizione	Tipo	Obbligatorio	Predefinita
	is set to false. Cannot download.			

> **Note**
>
> Per gli esempi seguenti, il percorso della cartella Windows può essere sostituito con un percorso Linux. Ad esempio, *C:\myfolder\package.zip* può essere sostituito con*/myfolder/package.zip*.

Esempio di input: copia un oggetto Amazon S3 in un file locale

L'esempio seguente mostra come copiare un oggetto Amazon S3 in un file locale.

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

Esempio di input: copia tutti gli oggetti Amazon S3 in un bucket Amazon S3 con prefisso chiave in una cartella locale

L'esempio seguente mostra come copiare tutti gli oggetti Amazon S3 in un bucket Amazon S3 con il prefisso chiave in una cartella locale. Amazon S3 non ha il concetto di cartella, pertanto tutti gli

oggetti che corrispondono al prefisso del key vengono copiati. Il numero massimo di oggetti che possono essere scaricati è 1000.

```
- name: MyS3DownloadKeyprefix
  action: S3Download
  maxAttempts: 3
  inputs:
    - source: s3://amzn-s3-demo-source-bucket/path/to/*
      destination: C:\myfolder\
      expectedBucketOwner: 123456789022
      overwrite: false
    - source: s3://amzn-s3-demo-source-bucket/path/to/*
      destination: C:\myfolder\
      expectedBucketOwner: 123456789022
      overwrite: true
    - source: s3://amzn-s3-demo-source-bucket/path/to/*
      destination: C:\myfolder\
      expectedBucketOwner: 123456789022
```

Output

Nessuna.

Caricamento S3 (Linux, Windows, macOS)

Con il modulo di azione S3Upload, puoi caricare un file da un file o una cartella di origine in una posizione Amazon S3. Puoi utilizzare un carattere jolly (*) nel percorso specificato per la posizione di origine per caricare tutti i file il cui percorso corrisponde al modello dei caratteri jolly.

Se l'azione ricorsiva S3Upload fallisce, tutti i file che sono già stati caricati rimarranno nel bucket Amazon S3 di destinazione.

Casi di utilizzo supportati

- File locale su oggetto Amazon S3.
- File locali nella cartella (con wildcard) con prefisso chiave Amazon S3.
- Copia la cartella locale (deve essere `recurse` impostata su`true`) nel prefisso chiave di Amazon S3.

Requisiti IAM

Il ruolo IAM associato al profilo dell'istanza deve disporre delle autorizzazioni per eseguire il modulo di `S3Upload` azione. La seguente policy IAM deve essere associata al ruolo IAM associato al profilo dell'istanza. La policy deve concedere `s3:PutObject` le autorizzazioni al bucket Amazon S3 di destinazione. Ad esempio, `arn:aws:s3:::BucketName/*`.

Input

Chiave	Descrizione	Tipo	Obbligatorio	Predefinita
`source`	Il percorso locale da cui proviene l'origine. files/folders `source`Supporta un asterisco wildcard (`*`).	Stringa	Sì	N/D
`destination`	Il percorso del bucket Amazon S3 di destinazione in cui vengono caricati i file/le cartelle di origine.	Stringa	Sì	N/D
`recurse`	Se impostato su, esegue S3Upload in modo **`true`** ricorsivo.	Stringa	No	`false`
`expectedBucketOwner`	L'ID dell'account proprietario previsto per il bucket Amazon	Stringa	No	N/D

Chiave	Descrizione	Tipo	Obbligatorio	Predefinita
	S3 specificato nel percorso di destinazione. Ti consigliamo di verificare la proprietà del bucket Amazon S3 specificato nella destinazione.			

Esempio di input: copia un file locale su un oggetto Amazon S3

L'esempio seguente mostra come copiare un file locale su un oggetto Amazon S3.

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

Esempio di input: copia tutti i file in una cartella locale in un bucket Amazon S3 con prefisso chiave

L'esempio seguente mostra come copiare tutti i file nella cartella locale in un bucket Amazon S3 con key prefix. Questo esempio non copia le sottocartelle o il loro contenuto perché non `recurse` è specificato e l'impostazione predefinita è. `false`

```
- name: MyS3UploadMultipleFiles
  action: S3Upload
  onFailure: Abort
  maxAttempts: 3
  inputs:
    - source: C:\myfolder\*
      destination: s3://amzn-s3-demo-destination-bucket/path/to/
```

```
        expectedBucketOwner: 123456789022
```

Esempio di input: copia ricorsivamente tutti i file e le cartelle da una cartella locale a un bucket Amazon S3

L'esempio seguente mostra come copiare tutti i file e le cartelle in modo ricorsivo da una cartella locale a un bucket Amazon S3 con prefisso chiave.

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

Output

Nessuna.

WebDownload (Linux, Windows, macOS)

Il modulo di WebDownloadazione consente di scaricare file e risorse da una postazione remota tramite il HTTP/HTTPS protocollo (si consiglia HTTPS). Non ci sono limiti al numero o alla dimensione dei download. Questo modulo gestisce la logica dei tentativi e del backoff esponenziale.

A ogni operazione di download vengono assegnati un massimo di 5 tentativi di successo in base agli input dell'utente. Questi tentativi sono diversi da quelli specificati nel `maxAttempts` campo del documento`steps`, che sono correlati agli errori del modulo di azione.

Questo modulo di azione gestisce implicitamente i reindirizzamenti. Tutti i codici di stato HTTP, ad eccezione di`200`, generano un errore.

Input

Nome chiave	Descrizione	Tipo	Obbligatorio	Predefinita
`source`	L' HTTP/HTTPS URL valido (si consiglia	Stringa	Sì	N/D

Nome chiave	Descrizione	Tipo	Obbligatorio	Predefinita
	HTTPS), che segue lo standard RFC 3986. Le espressioni concatenate sono consentite.			
`destination`	Un percorso di file o cartella assoluto o relativo nel sistema locale. I percorsi delle cartelle devono terminare con`/`. Se non terminano con`/`, verranno trattati come percorsi di file. Il modulo crea qualsiasi file o cartella necessario per il corretto download. Le espressioni concatenate sono consentite.	Stringa	Sì	N/D

Nome chiave	Descrizione	Tipo	Obbligatorio	Predefinita
`overwrite`	Se abilitata, sovrascrive tutti i file esistenti sul sistema locale con il file o la risorsa scaricati. Se non è abilitata, i file esistenti sul sistema locale non vengono sovrascritti e il modulo di azione si interrompe e restituisce un errore. Quando la sovrascri ttura è abilitata e vengono specificati il checksum e l'algoritmo, il modulo di azione scarica il file solo se il checksum e l'hash dei file preesistenti non corrispondono.	Boolean	No	`true`

Nome chiave	Descrizione	Tipo	Obbligatorio	Predefinita
`checksum`	Quando si specifica il checksum, questo viene confrontato con l'hash del file scaricato generato con l'algoritmo fornito. Affinché la verifica dei file sia abilitata, è necessario fornire sia il checksum che l'algoritmo. Le espressioni concatenate sono consentite.	Stringa	No	N/D

Nome chiave	Descrizione	Tipo	Obbligatorio	Predefinita
`algorithm`	L'algoritmo utilizzato per calcolare il checksum. Le opzioni sonoMD5, SHA1 SHA256, e SHA512. Per abilitare la verifica dei file, è necessari o fornire sia il checksum che l'algoritmo. Le espressioni concatenate sono consentite.	Stringa	No	N/D
`ignoreCertificateErrors`	La convalida del certificato SSL viene ignorata se abilitata.	Boolean	No	`false`

Output

Nome chiave	Descrizione	Tipo
`destination`	Stringa di nuova riga delimitata da caratteri che specifica il percorso di	Stringa

Nome chiave	Descrizione	Tipo				
	destinazione in cui sono archiviati i file o le risorse scaricati.					

Esempio di input: scarica il file remoto nella destinazione 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"
}
```

Esempio di input: scaricare più di un file remoto su più di una destinazione locale

```
  - 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"
}
```

Esempio di input: scarica un file remoto senza sovrascrivere la destinazione locale e scarica un altro file remoto con la verifica del file

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

Output:

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

Esempio di input: scarica il file remoto e ignora la convalida della certificazione 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"
}
```

Moduli operativi del file system

La sezione seguente contiene dettagli sui moduli di azione che eseguono operazioni sul file system.

Moduli di azione per il funzionamento del file system

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

AppendFile (Linux, Windows, macOS)

Il modulo di AppendFileazione aggiunge il contenuto specificato al contenuto preesistente di un file.

Se il valore di codifica del file è diverso dal valore di codifica predefinito (`utf-8`), è possibile specificare il valore di codifica del file utilizzando l'opzione. `encoding` Per impostazione predefinita, `utf-16` si presume che utilizzino la `utf-32` codifica little-endian.

Il modulo di azione restituisce un errore quando si verifica quanto segue:

- Il file specificato non esiste in fase di esecuzione.
- Non disponi delle autorizzazioni di scrittura per modificare il contenuto del file.
- Il modulo rileva un errore durante l'operazione sul file.

Input

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
`path`	Il percorso del file.	Stringa	Sì	N/D	N/D	Sì
`content`	Il contenuto da aggiungere al file.	Stringa	No	Stringa vuota	N/D	Sì
`encoding`	Lo standard di codifica.	Stringa	No	`utf8`	`utf8`,`utf-8-LE`, `utf-16-LE` `utf16-BE`utf-16-BE ,`utf32`,`utf-LE`, `utf-32-LE` utf32-BE, e `utf-32-BE` . Il valore dell'opzione di	Sì

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
					codifica non fa distinzione tra maiuscole e minuscole.	

Esempio di input: aggiungi file senza codifica (Linux)

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

Esempio di input: aggiungi file senza codifica (Windows)

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

Esempio di input: aggiungi file con codifica (Linux)

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

Esempio di input: aggiungi file con codifica (Windows)

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

Esempio di input: aggiungi un file con una stringa vuota (Linux)

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

Esempio di input: aggiungi un file con una stringa vuota (Windows)

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

Output

Nessuna.

CopyFile (Linux, Windows, macOS)

Il modulo di CopyFileazione copia i file dall'origine specificata alla destinazione specificata. Per impostazione predefinita, il modulo crea in modo ricorsivo la cartella di destinazione se non esiste in fase di esecuzione.

Se un file con il nome specificato esiste già nella cartella specificata, il modulo di azione, per impostazione predefinita, sovrascrive il file esistente. È possibile ignorare questo comportamento predefinito impostando l'opzione di sovrascrittura su. `false` Quando l'opzione di sovrascrittura è impostata su e nella posizione specificata è già presente un file con il nome specificato, il modulo di azione restituirà un errore. `false` Questa opzione funziona come il `cp` comando in Linux, che per impostazione predefinita sovrascrive.

Il nome del file sorgente può includere un wildcard ()*. I caratteri jolly sono accettati solo dopo l'ultimo separatore di percorso del file (/o). \ Se nel nome del file di origine sono inclusi caratteri jolly, tutti i

file che corrispondono ai caratteri jolly vengono copiati nella cartella di destinazione. Se desiderate spostare più di un file utilizzando un carattere jolly, l'input dell'`destination`opzione deve terminare con un separatore del percorso del file (`/o\`), che indica che l'input di destinazione è una cartella.

Se il nome del file di destinazione è diverso dal nome del file di origine, è possibile specificare il nome del file di destinazione utilizzando l'`destination`opzione. Se non si specifica un nome di file di destinazione, il nome del file di origine viene utilizzato per creare il file di destinazione. Qualsiasi testo che segue il separatore (`/o\`) dell'ultimo percorso del file viene considerato come nome del file. Se si desidera utilizzare lo stesso nome di file del file di origine, l'input dell'`destination`opzione deve terminare con un separatore di percorso del file (`/o\`).

Il modulo di azione restituisce un errore quando si verifica quanto segue:

- Non sei autorizzato a creare un file nella cartella specificata.
- I file di origine non esistono in fase di esecuzione.
- Esiste già una cartella con il nome di file specificato e l'`overwrite`opzione è impostata su`false`.
- Il modulo di azione rileva un errore durante l'esecuzione dell'operazione.

Input

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
`source`	Il percorso del file sorgente.	Stringa	Sì	N/D	N/D	Sì
`destination`	Il percorso del file di destinazione.	Stringa	Sì	N/D	N/D	Sì
`overwrite`	Se impostato su false, i file di destinazi	Boolean	No	`true`	N/D	Sì

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
	one non verranno sostituiti quando nella posizione specificata è già presente un file con il nome specificato.					

Esempio di input: copia un file (Linux)

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

Esempio di input: copiare un file (Windows)

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

Esempio di input: copia un file usando il nome del file sorgente (Linux)

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

```
      destination: /MyFolder/
```

Esempio di input: copiare un file utilizzando il nome del file sorgente (Windows)

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

Esempio di input: copia un file usando il carattere jolly (Linux)

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

Esempio di input: copia un file usando il carattere jolly (Windows)

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

Esempio di input: copiare un file senza sovrascriverlo (Linux)

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

Esempio di input: copiare un file senza sovrascriverlo (Windows)

```
- name: CopyingFilesWithoutOverwriteWindows
  action: CopyFile
  inputs:
```

```
      - source: C:\Sample\MyFolder\Sample.txt
        destination: C:\MyFolder\destinationFile.txt
        overwrite: false
```

Output

Nessuna.

CopyFolder (Linux, Windows, macOS)

Il modulo di CopyFolderazione copia una cartella dall'origine specificata alla destinazione specificata. L'input per l'`source`opzione è la cartella da copiare e l'input per l'`destination`opzione è la cartella in cui viene copiato il contenuto della cartella di origine. Per impostazione predefinita, il modulo crea in modo ricorsivo la cartella di destinazione se non esiste in fase di esecuzione.

Se una cartella con il nome specificato esiste già nella cartella specificata, il modulo di azione, per impostazione predefinita, sovrascrive la cartella esistente. È possibile ignorare questo comportamento predefinito impostando l'opzione di sovrascrittura su. `false` Quando l'opzione di sovrascrittura è impostata su e nella posizione specificata è già presente una cartella con il nome specificato, il modulo di azione restituirà un errore. `false`

Il nome della cartella di origine può includere un carattere jolly ()*. I caratteri jolly sono accettati solo dopo l'ultimo separatore di percorso del file (/o). \ Se nel nome della cartella di origine sono inclusi caratteri jolly, tutte le cartelle che corrispondono al carattere jolly vengono copiate nella cartella di destinazione. Se desiderate copiare più di una cartella utilizzando un carattere jolly, l'input dell'`destination`opzione deve terminare con un separatore del percorso del file (/o\), che indica che l'input di destinazione è una cartella.

Se il nome della cartella di destinazione è diverso dal nome della cartella di origine, è possibile specificare il nome della cartella di destinazione utilizzando l'`destination`opzione. Se non si specifica un nome per la cartella di destinazione, per creare la cartella di destinazione viene utilizzato il nome della cartella di origine. Qualsiasi testo che segue il separatore (/o\) dell'ultimo percorso del file viene considerato come nome della cartella. Se si desidera utilizzare lo stesso nome di cartella della cartella di origine, l'immissione dell'`destination`opzione deve terminare con un separatore del percorso del file (/o\).

Il modulo di azione restituisce un errore quando si verifica quanto segue:

- Non sei autorizzato a creare una cartella nella cartella specificata.
- Le cartelle di origine non esistono in fase di esecuzione.

- Esiste già una cartella con il nome specificato e l'`overwrite`opzione è impostata su`false`.
- Il modulo di azione rileva un errore durante l'esecuzione dell'operazione.

Input

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
`source`	Il percorso della cartella di origine.	Stringa	Sì	N/D	N/D	Sì
`destination`	Il percorso della cartella di destinazione.	Stringa	Sì	N/D	N/D	Sì
`overwrite`	Se impostato su false, le cartelle di destinazione non verranno sostituite se nella posizione specificata è già presente una cartella con il nome specificato.	Boolean	No	`true`	N/D	Sì

Esempio di input: copia una cartella (Linux)

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

Esempio di input: copiare una cartella (Windows)

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

Esempio di input: copia una cartella usando il nome della cartella di origine (Linux)

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

Esempio di input: copia una cartella usando il nome della cartella di origine (Windows)

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

Esempio di input: copia una cartella usando il carattere jolly (Linux)

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

Esempio di input: copia una cartella usando il carattere jolly (Windows)

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

Esempio di input: copia una cartella senza sovrascriverla (Linux)

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

Esempio di input: copiare una cartella senza sovrascriverla (Windows)

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

Output

Nessuna.

CreateFile (Linux, Windows, macOS)

Il modulo di CreateFileazione crea un file in una posizione specificata. Per impostazione predefinita, se necessario, il modulo crea anche in modo ricorsivo le cartelle principali.

Se il file esiste già nella cartella specificata, il modulo di azione, per impostazione predefinita, tronca o sovrascrive il file esistente. È possibile ignorare questo comportamento predefinito impostando l'opzione di sovrascrittura su. `false` Quando l'opzione di sovrascrittura è impostata su e nella posizione specificata è già presente un file con il nome specificato, il modulo di azione restituirà un errore. `false`

Se il valore di codifica del file è diverso dal valore di codifica predefinito (`utf-8`), è possibile specificare il valore di codifica del file utilizzando l'opzione. `encoding` Per impostazione predefinita, `utf-16` si presume che utilizzino la `utf-32` codifica little-endian.

`owner`, e sono input `group` opzionali. `permissions` L'input per `permissions` deve essere un valore di stringa. I file vengono creati con valori predefiniti quando non vengono forniti. Queste opzioni non sono supportate sulle piattaforme Windows. Questo modulo di azione convalida e restituisce un errore se le `permissions` opzioni `ownergroup`, e vengono utilizzate sulle piattaforme Windows.

Questo modulo di azione può creare un file con autorizzazioni definite dal `umask` valore predefinito del sistema operativo. È necessario impostare il `umask` valore se si desidera sovrascrivere il valore predefinito.

Il modulo di azione restituisce un errore quando si verifica quanto segue:

- Non sei autorizzato a creare un file o una cartella nella cartella principale specificata.
- Il modulo di azione rileva un errore durante l'esecuzione dell'operazione.

Input

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme	
`path`	Il percorso del file.	Stringa	Sì	N/D	N/D	Sì	
`content`	Il contenuto testuale del file.	Stringa	No	N/D	N/D	Sì	
`encoding`	Lo standard di codifica.	Stringa	No	`utf8`	`utf8`,`utf-8`, `utf-16-LE`, `utf16-BE`utf-16-	Sì	6,utf1

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
					BE ,utf32,utfLE, utf-32-LE utf32-BE, e utf-32-BE . Il valore dell'opzione di codifica non fa distinzione tra maiuscole e minuscole.	
owner	Il nome utente o l'ID.	Stringa	No	N/D	N/D	Non supportato su Windows.
group	Il nome o l'ID del gruppo.	Stringa	No	L'utente corrente.	N/D	Non supportato su Windows.

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
`permissions`	I permessi dei file.	Stringa	No	`0666`	N/D	Non supportato su Windows.
`overwrite`	Se il nome del file specificato esiste già, impostando questo valore per `false` evitare che il file venga troncato o sovrascritto per impostazione predefinita.	Boolean	No	`true`	N/D	Sì

Esempio di input: crea un file senza sovrascrivere (Linux)

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

Esempio di input: crea un file senza sovrascrivere (Windows)

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

Esempio di input: creare un file con le proprietà del file

```
- name: CreatingFileWithFileProperties
  action: CreateFile
  inputs:
    - path: SampleFolder/Sample.txt
      content: The text content of the sample file.
      encoding: UTF-16
      owner: Ubuntu
      group: UbuntuGroup
      permissions: 0777
   - path: SampleFolder/SampleFile.txt
      permissions: 755
    - path: SampleFolder/TextFile.txt
      encoding: UTF-16
      owner: root
      group: rootUserGroup
```

Esempio di input: creare un file senza proprietà del file

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

Esempio di input: crea un file vuoto per saltare una sezione dello script di pulizia di Linux

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

Per ulteriori informazioni, consulta Sostituisci lo script di pulizia di Linux

Output

Nessuna.

CreateFolder (Linux, Windows, macOS)

Il modulo di CreateFolderazione crea una cartella in una posizione specificata. Per impostazione predefinita, se necessario, il modulo crea anche in modo ricorsivo le cartelle principali.

Se la cartella esiste già nella cartella specificata, il modulo di azione, per impostazione predefinita, tronca o sovrascrive la cartella esistente. È possibile ignorare questo comportamento predefinito impostando l'opzione di sovrascrittura su. `false` Quando l'opzione di sovrascrittura è impostata su e nella posizione specificata è già presente una cartella con il nome specificato, il modulo di azione restituirà un errore. `false`

`ownergroup`, e `permissions` sono input opzionali. L'input per `permissions` deve essere un valore di stringa. Queste opzioni non sono supportate sulle piattaforme Windows. Questo modulo di azione convalida e restituisce un errore se le `permissions` opzioni `ownergroup`, e vengono utilizzate sulle piattaforme Windows.

Questo modulo di azione può creare una cartella con autorizzazioni definite dal `umask` valore predefinito del sistema operativo. È necessario impostare il `umask` valore se si desidera sovrascrivere il valore predefinito.

Il modulo di azione restituisce un errore quando si verifica quanto segue:

- Non sei autorizzato a creare una cartella nella posizione specificata.
- Il modulo di azione rileva un errore durante l'esecuzione dell'operazione.

Input

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
`path`	Il percorso della cartella.	Stringa	Sì	N/D	N/D	Sì

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
`owner`	Il nome utente o l'ID.	Stringa	No	L'utente corrente.	N/D	Non supportato su Windows.
`group`	Il nome o l'ID del gruppo.	Stringa	No	Il gruppo dell'utente corrente.	N/D	Non supportato su Windows.
`permissions`	Le autorizzazioni della cartella.	Stringa	No	`0777`	N/D	Non supportato su Windows.
`overwrite`	Se il nome del file specificato esiste già, impostando questo valore per `false` evitare che il file venga troncato o sovrascritto per impostazione predefinita.	Boolean	No	`true`	N/D	Sì

Esempio di input: creare una cartella (Linux)

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

Esempio di input: creare una cartella (Windows)

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

Esempio di input: creare una cartella specificando le proprietà della cartella

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

Esempio di input: crea una cartella che sovrascriva la cartella esistente, se presente.

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

Output

Nessuna.

CreateSymlink (Linux, Windows, macOS)

Il modulo di CreateSymlinkazione crea collegamenti simbolici o file che contengono un riferimento a un altro file. Questo modulo non è supportato sulle piattaforme Windows.

L'input per le `target` opzioni `path` and può essere un percorso assoluto o relativo. Se l'input per l'`path`opzione è un percorso relativo, viene sostituito con il percorso assoluto al momento della creazione del collegamento.

Per impostazione predefinita, quando un link con il nome specificato esiste già nella cartella specificata, il modulo di azione restituisce un errore. È possibile ignorare questo comportamento predefinito impostando l'`force`opzione su`true`. Quando l'`force`opzione è impostata su`true`, il modulo sovrascriverà il link esistente.

Se non esiste una cartella principale, il modulo di azione crea la cartella in modo ricorsivo, per impostazione predefinita.

Il modulo di azione restituisce un errore quando si verifica quanto segue:

- Il file di destinazione non esiste in fase di esecuzione.
- Esiste già un file di collegamento non simbolico con il nome specificato.
- Il modulo di azione rileva un errore durante l'esecuzione dell'operazione.

Input

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
`path`	Il percorso del file.	Stringa	Sì	N/D	N/D	Non supportato su Windows.
`target`	Il percorso del file di destinazione a cui punta il link simbolico.	Stringa	Sì	N/D	N/D	Non supportato in Windows.
`force`	Forza la creazione	Boolean	No	`false`	N/D	Non supportat

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
	di un collegamento quando esiste già un collegamento con lo stesso nome.					o in Windows.

Esempio di input: crea un link simbolico che forza la creazione di un link

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

Esempio di input: crea un link simbolico che non imponga la creazione di un link

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

Output

Nessuna.

DeleteFile (Linux, Windows, macOS)

Il modulo di DeleteFileazione elimina uno o più file in una posizione specificata.

L'input di `path` deve essere un percorso di file valido o un percorso di file con un carattere wild card (*) nel nome del file. Quando nel nome del file vengono specificati caratteri jolly, tutti i file all'interno della stessa cartella che corrispondono al carattere jolly verranno eliminati.

Il modulo di azione restituisce un errore quando si verifica quanto segue:

- Non sei autorizzato a eseguire operazioni di eliminazione.
- Il modulo di azione rileva un errore durante l'esecuzione dell'operazione.

Input

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
`path`	Il percorso del file.	Stringa	Sì	N/D	N/D	Sì

Esempio di input: eliminare un singolo file (Linux)

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

Esempio di input: eliminare un singolo file (Windows)

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

Esempio di input: elimina un file che termina con «log» (Linux)

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

Esempio di input: eliminare un file che termina con «log» (Windows)

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

Esempio di input: elimina tutti i file in una cartella specificata (Linux)

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

Esempio di input: elimina tutti i file in una cartella specificata (Windows)

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

Output

Nessuna.

DeleteFolder (Linux, Windows, macOS)

Il modulo di DeleteFolderazione elimina le cartelle.

Se la cartella non è vuota, è necessario impostare l'`force`opzione per `true` rimuovere la cartella e il suo contenuto. Se non si imposta l'`force`opzione su e la cartella che si sta tentando di eliminare non è vuota, il modulo di azione restituisce un errore. `true` Il valore predefinito dell'`force`opzione è`false`.

Il modulo di azione restituisce un errore quando si verifica quanto segue:

- Non sei autorizzato a eseguire operazioni di eliminazione.
- Il modulo di azione rileva un errore durante l'esecuzione dell'operazione.

Input

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
`path`	Il percorso della cartella.	Stringa	Sì	N/D	N/D	Sì
`force`	Rimuove la cartella indipendentemente dal fatto che sia vuota o meno.	Boolean	No	`false`	N/D	Sì

Esempio di input: elimina una cartella che non è vuota usando l'**`force`**opzione (Linux)

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

Esempio di input: elimina una cartella che non è vuota usando l'**`force`**opzione (Windows)

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

Esempio di input: eliminare una cartella (Linux)

```
- name: DeletingFolderWithOutForceLinux
  action: DeleteFolder
```

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

Esempio di input: eliminare una cartella (Windows)

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

Output

Nessuna.

ListFiles (Linux, Windows, macOS)

Il modulo di ListFilesazione elenca i file in una cartella specificata. Quando l'opzione ricorsiva è impostata su`true`, elenca i file nelle sottocartelle. Per impostazione predefinita, questo modulo non elenca i file nelle sottocartelle.

Per elencare tutti i file con nomi che corrispondono a uno schema specificato, utilizzate l'`fileNamePattern`opzione per fornire il modello. L'`fileNamePattern`opzione accetta il valore wildcard (`*`). Quando `fileNamePattern` viene fornito, vengono restituiti tutti i file che corrispondono al formato del nome file specificato.

Il modulo di azione restituisce un errore quando si verifica quanto segue:

- La cartella specificata non esiste in fase di esecuzione.
- Non sei autorizzato a creare un file o una cartella nella cartella principale specificata.
- Il modulo di azione rileva un errore durante l'esecuzione dell'operazione.

Input

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
`path`	Il percorso della cartella.	Stringa	Sì	N/D	N/D	Sì

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
`fileNamePattern`	Lo schema a cui abbinare elenca tutti i file con nomi che corrispondono al modello.	Stringa	No	N/D	N/D	Sì
`recursive`	Elenca i file nella cartella in modo ricorsivo.	Boolean	No	`false`	N/D	Sì

Esempio di input: elenca i file nella cartella specificata (Linux)

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

Esempio di input: elenca i file nella cartella specificata (Windows)

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

Esempio di input: elenca i file che terminano con «log» (Linux)

```
- name: ListingFilesWithEndingWithLogLinux
  action: ListFiles
```

```
    inputs:
      - path: /Sample/MyFolder/
        fileNamePattern: *log
```

Esempio di input: elenca i file che terminano con «log» (Windows)

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

Esempio di input: elenca i file in modo ricorsivo

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

Output

Nome chiave	Descrizione	Tipo
`files`	L'elenco dei file.	Stringa

Esempio di output

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

MoveFile (Linux, Windows, macOS)

Il modulo di MoveFileazione sposta i file dall'origine specificata alla destinazione specificata.

Se il file esiste già nella cartella specificata, il modulo di azione, per impostazione predefinita, sovrascrive il file esistente. È possibile ignorare questo comportamento predefinito impostando

l'opzione di sovrascrittura su. `false` Quando l'opzione di sovrascrittura è impostata su e nella posizione specificata è già presente un file con il nome specificato, il modulo di azione restituirà un errore. `false` Questa opzione funziona come il `mv` comando in Linux, che per impostazione predefinita sovrascrive.

Il nome del file sorgente può includere un wildcard ()`*`. I caratteri jolly sono accettati solo dopo l'ultimo separatore di percorso del file (/o). \ Se nel nome del file di origine sono inclusi caratteri jolly, tutti i file che corrispondono ai caratteri jolly vengono copiati nella cartella di destinazione. Se desiderate spostare più di un file utilizzando un carattere jolly, l'input dell'`destination`opzione deve terminare con un separatore del percorso del file (/o\), che indica che l'input di destinazione è una cartella.

Se il nome del file di destinazione è diverso dal nome del file di origine, è possibile specificare il nome del file di destinazione utilizzando l'`destination`opzione. Se non si specifica un nome di file di destinazione, il nome del file di origine viene utilizzato per creare il file di destinazione. Qualsiasi testo che segue il separatore (/o\) dell'ultimo percorso del file viene considerato come nome del file. Se si desidera utilizzare lo stesso nome di file del file di origine, l'input dell'`destination`opzione deve terminare con un separatore di percorso del file (/o\).

Il modulo di azione restituisce un errore quando si verifica quanto segue:

- Non sei autorizzato a creare un file nella cartella specificata.
- I file di origine non esistono in fase di esecuzione.
- Esiste già una cartella con il nome di file specificato e l'`overwrite`opzione è impostata su`false`.
- Il modulo di azione rileva un errore durante l'esecuzione dell'operazione.

Input

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
`source`	Il percorso del file sorgente.	Stringa	Sì	N/D	N/D	Sì
`destination`	Il percorso del file di	Stringa	Sì	N/D	N/D	Sì

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
	destinazione.					
`overwrite`	Se impostato su false, i file di destinazione non verranno sostituiti quando nella posizione specificata è già presente un file con il nome specificato.	Boolean	No	`true`	N/D	Sì

Esempio di input: sposta un file (Linux)

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

Esempio di input: sposta un file (Windows)

```
- name: MovingAFileWindows
  action: MoveFile
  inputs:
```

```
      - source: C:\Sample\MyFolder\Sample.txt
        destination: C:\MyFolder\destinationFile.txt
```

Esempio di input: sposta un file usando il nome del file sorgente (Linux)

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

Esempio di input: sposta un file usando il nome del file sorgente (Windows)

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

Esempio di input: sposta un file usando un carattere jolly (Linux)

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

Esempio di input: sposta un file usando un carattere jolly (Windows)

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

Esempio di input: sposta un file senza sovrascriverlo (Linux)

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

```
        destination: /MyFolder/destinationFile.txt
        overwrite: false
```

Esempio di input: sposta un file senza sovrascriverlo (Windows)

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

Output

Nessuna.

MoveFolder (Linux, Windows, macOS)

Il modulo di MoveFolderazione sposta le cartelle dall'origine specificata alla destinazione specificata. L'input dell'`source`opzione è la cartella da spostare e l'input dell'`destination`opzione è la cartella in cui viene spostato il contenuto delle cartelle di origine.

Se la cartella principale di destinazione o l'input dell'`destination`opzione non esiste in fase di esecuzione, il comportamento predefinito del modulo consiste nel creare ricorsivamente la cartella nella destinazione specificata.

Se una cartella con la stessa cartella di origine esiste già nella cartella di destinazione, il modulo di azione, per impostazione predefinita, sovrascrive la cartella esistente. È possibile ignorare questo comportamento predefinito impostando l'opzione di sovrascrittura su. `false` Quando l'opzione di sovrascrittura è impostata su e nella posizione specificata è già presente una cartella con il nome specificato, il modulo di azione restituirà un errore. `false`

Il nome della cartella di origine può includere un carattere jolly ()`*`. I caratteri jolly sono accettati solo dopo l'ultimo separatore di percorso del file (`/`o). \ Se nel nome della cartella di origine sono inclusi caratteri jolly, tutte le cartelle che corrispondono al carattere jolly vengono copiate nella cartella di destinazione. Se desiderate spostare più di una cartella utilizzando un carattere jolly, l'input dell'`destination`opzione deve terminare con un separatore del percorso del file (`/`o`\`), che indica che l'input di destinazione è una cartella.

Se il nome della cartella di destinazione è diverso dal nome della cartella di origine, è possibile specificare il nome della cartella di destinazione utilizzando l'`destination`opzione. Se non si

specifica un nome per la cartella di destinazione, per creare la cartella di destinazione viene utilizzato il nome della cartella di origine. Qualsiasi testo che segue il separatore (/o\) dell'ultimo percorso del file viene considerato come nome della cartella. Se si desidera utilizzare lo stesso nome di cartella della cartella di origine, l'immissione dell'`destination`opzione deve terminare con un separatore del percorso del file (/o\).

Il modulo di azione restituisce un errore quando si verifica quanto segue:

- Non sei autorizzato a creare una cartella nella cartella di destinazione.
- Le cartelle di origine non esistono in fase di esecuzione.
- Esiste già una cartella con il nome specificato e l'`overwrite`opzione è impostata su`false`.
- Il modulo di azione rileva un errore durante l'esecuzione dell'operazione.

Input

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
`source`	Il percorso della cartella di origine.	Stringa	Sì	N/D	N/D	Sì
`destination`	Il percorso della cartella di destinazione.	Stringa	Sì	N/D	N/D	Sì
`overwrite`	Se impostato su false, le cartelle di destinazione non verranno sostituite	Boolean	No	`true`	N/D	Sì

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
	se nella posizione specificata è già presente una cartella con il nome specificato.					

Esempio di input: sposta una cartella (Linux)

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

Esempio di input: sposta una cartella (Windows)

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

Esempio di input: sposta una cartella usando il nome della cartella di origine (Linux)

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

Esempio di input: sposta una cartella usando il nome della cartella di origine (Windows)

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

Esempio di input: sposta una cartella usando un carattere jolly (Linux)

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

Esempio di input: sposta una cartella usando un carattere jolly (Windows)

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

Esempio di input: sposta una cartella senza sovrascriverla (Linux)

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

Esempio di input: sposta una cartella senza sovrascriverla (Windows)

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

Output

Nessuna.

ReadFile (Linux, Windows, macOS)

Il modulo di ReadFileazione legge il contenuto di un file di testo di tipo string. Questo modulo può essere utilizzato per leggere il contenuto di un file da utilizzare nei passaggi successivi tramite concatenamento o per leggere i dati nel `console.log` file. Se il percorso specificato è un collegamento simbolico, questo modulo restituisce il contenuto del file di destinazione. Questo modulo supporta solo file di testo.

Se il valore di codifica del file è diverso dal valore di codifica predefinito (`utf-8`), è possibile specificare il valore di codifica del file utilizzando l'opzione. `encoding` Per impostazione predefinita, `utf-16` si presume che utilizzino la `utf-32` codifica little-endian.

Per impostazione predefinita, questo modulo non può stampare il contenuto del file sul file. `console.log` È possibile ignorare questa impostazione impostando la `printFileContent` proprietà su`true`.

Questo modulo può restituire solo il contenuto di un file. Non può analizzare file, come file Excel o JSON.

Il modulo di azione restituisce un errore quando si verifica quanto segue:

- Il file non esiste in fase di esecuzione.
- Il modulo di azione rileva un errore durante l'esecuzione dell'operazione.

Input

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
`path`	Il percorso del file.	Stringa	Sì	N/D	N/D	Sì
`encoding`	Lo standard di codifica.	Stringa	No	`utf8`	`utf8`,`utf-8-LE`, `utf-16-LE`	Sì

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
					`utf16-BEutf-16-BE` ,`utf32`,`utfLE`, `utf-32-LE` `utf32-BE`, e `utf-32-BE` . Il valore dell'opzione di codifica non fa distinzione tra maiuscole e minuscole.	
`printFileContent`	Stampa il contenuto del file sul file. `console.log`	Boolean	No	false	N/D	Sì.

Esempio di input: lettura di un file (Linux)

```
- name: ReadingFileLinux
  action: ReadFile
```

```
    inputs:
      - path: /home/UserName/SampleFile.txt
```

Esempio di input: leggere un file (Windows)

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

Esempio di input: leggere un file e specificare lo standard di codifica

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

Esempio di input: leggere un file e stampare sul **`console.log`** file

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

Output

Campo	Descrizione	Tipo
`content`	Il contenuto del file.	string

Esempio di output

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

SetFileEncoding (Linux, Windows, macOS)

Il modulo di SetFileEncodingazione modifica la proprietà di codifica di un file esistente. Questo modulo può convertire la codifica dei file da uno standard `utf-8` di codifica specificato. Per impostazione predefinita, `utf-16` si presume che sia una codifica little-endian. `utf-32`

Il modulo di azione restituisce un errore quando si verifica quanto segue:

- Non sei autorizzato a eseguire la modifica specificata.
- Il file non esiste in fase di esecuzione.
- Il modulo di azione rileva un errore durante l'esecuzione dell'operazione.

Input

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme	
`path`	Il percorso del file.	Stringa	Sì	N/D	N/D	Sì	
`encoding`	Lo standard di codifica.	Stringa	No	`utf8`	`utf8`,`utf-8`, `utf-16-LE`, `utf16-BE`,`utf-16-BE`, `utf32`,`utf-32-LE`, `utf32-BE`, e `utf-32-BE`. Il valore dell'opzi	Sì	`6`,`utf1`

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
					one di codifica non fa distinzione tra maiuscole e minuscole.	

Esempio di input: imposta la proprietà di codifica del file

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

Output

Nessuna.

SetFileOwner (Linux, Windows, macOS)

Il modulo di SetFileOwnerazione modifica le proprietà `owner` e il `group` proprietario di un file esistente. Se il file specificato è un collegamento simbolico, il modulo modifica la `owner` proprietà del file sorgente. Questo modulo non è supportato sulle piattaforme Windows.

Questo modulo accetta nomi di utenti e gruppi come input. Se il nome del gruppo non viene fornito, il modulo assegna il proprietario del gruppo del file al gruppo a cui appartiene l'utente.

Il modulo di azione restituisce un errore quando si verifica quanto segue:

- Non sei autorizzato a eseguire la modifica specificata.
- Il nome utente o gruppo specificato non esiste in fase di esecuzione.
- Il file non esiste in fase di esecuzione.

- Il modulo di azione rileva un errore durante l'esecuzione dell'operazione.

Input

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
`path`	Il percorso del file.	Stringa	Sì	N/D	N/D	Non supportato su Windows.
`owner`	Il nome utente.	string	Sì	N/D	N/D	Non supportato su Windows.
`group`	Il nome del gruppo di utenti.	Stringa	No	Il nome del gruppo a cui appartiene l'utente.	N/D	Non supportato su Windows.

Esempio di input: imposta la proprietà del proprietario del file senza specificare il nome del gruppo di utenti

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

Esempio di input: imposta la proprietà del proprietario del file specificando il proprietario e il gruppo di utenti

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

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

Output

Nessuna.

SetFolderOwner (Linux, Windows, macOS)

Il modulo di SetFolderOwnerazione modifica in modo ricorsivo le proprietà `owner` e il `group` proprietario di una cartella esistente. Per impostazione predefinita, il modulo può modificare la proprietà di tutti i contenuti di una cartella. È possibile impostare l'`recursive`opzione per `false` ignorare questo comportamento. Questo modulo non è supportato sulle piattaforme Windows.

Questo modulo accetta nomi di utenti e gruppi come input. Se il nome del gruppo non viene fornito, il modulo assegna il proprietario del gruppo del file al gruppo a cui appartiene l'utente.

Il modulo di azione restituisce un errore quando si verifica quanto segue:

- Non sei autorizzato a eseguire la modifica specificata.
- Il nome utente o gruppo specificato non esiste in fase di esecuzione.
- La cartella non esiste in fase di esecuzione.
- Il modulo di azione rileva un errore durante l'esecuzione dell'operazione.

Input

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
`path`	Il percorso della cartella.	Stringa	Sì	N/D	N/D	Non supportato su Windows.
`owner`	Il nome utente.	string	Sì	N/D	N/D	Non supportat

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
						o su Windows.
`group`	Il nome del gruppo di utenti.	Stringa	No	Il nome del gruppo a cui appartiene l'utente.	N/D	Non supportato su Windows.
`recursive`	Sostituisce il comportamento predefinito di modifica della proprietà di tutto il contenuto di una cartella quando è impostato su. `false`	Boolean	No	`true`	N/D	Non supportato in Windows.

Esempio di input: imposta la proprietà del proprietario della cartella senza specificare il nome del gruppo di utenti

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

Esempio di input: imposta la proprietà del proprietario della cartella senza sovrascrivere la proprietà di tutti i contenuti di una cartella

```
- name: SettingFolderPropertyWithOutRecursively
  action: SetFolderOwner
  inputs:
    - path: /SampleFolder/
      owner: LinuxUser
      recursive: false
```

Esempio di input: imposta la proprietà di proprietà del file specificando il nome del gruppo di utenti

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

Output

Nessuna.

SetFilePermissions (Linux, Windows, macOS)

Il modulo di SetFilePermissionsazione modifica un `permissions` file esistente. Questo modulo non è supportato sulle piattaforme Windows.

L'input per `permissions` deve essere un valore di stringa.

Questo modulo di azione può creare un file con i permessi definiti dal valore umask predefinito del sistema operativo. È necessario impostare il `umask` valore se si desidera sovrascrivere il valore predefinito.

Il modulo di azione restituisce un errore quando si verifica quanto segue:

- Non sei autorizzato a eseguire la modifica specificata.
- Il file non esiste in fase di esecuzione.
- Il modulo di azione rileva un errore durante l'esecuzione dell'operazione.

Input

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
`path`	Il percorso del file.	Stringa	Sì	N/D	N/D	Non supportato su Windows.
`permissions`	I permessi dei file.	Stringa	Sì	N/D	N/D	Non supportato su Windows.

Esempio di input: modifica delle autorizzazioni dei file

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

Output

Nessuna.

SetFolderPermissions (Linux, Windows, macOS)

Il modulo di SetFolderPermissionsazione modifica in modo ricorsivo una cartella esistente e tutti i relativi sottofile e sottocartelle. `permissions` Per impostazione predefinita, questo modulo può modificare le autorizzazioni per tutti i contenuti della cartella specificata. È possibile impostare l'`recursive`opzione per `false` ignorare questo comportamento. Questo modulo non è supportato sulle piattaforme Windows.

L'input per `permissions` deve essere un valore di stringa.

Questo modulo di azione può modificare le autorizzazioni in base al valore umask predefinito del sistema operativo. È necessario impostare il `umask` valore se si desidera sovrascrivere il valore predefinito.

Il modulo di azione restituisce un errore quando si verifica quanto segue:

- Non sei autorizzato a eseguire la modifica specificata.
- La cartella non esiste in fase di esecuzione.
- Il modulo di azione rileva un errore durante l'esecuzione dell'operazione.

Input

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
`path`	Il percorso della cartella.	Stringa	Sì	N/D	N/D	Non supportato su Windows.
`permissions`	Le autorizzazioni della cartella.	Stringa	Sì	N/D	N/D	Non supportato su Windows.
`recursive`	Sostituisce il comportamento predefinito di modifica delle autorizzazioni per tutto il contenuto di una	Boolean	No	`true`	N/D	Non supportato in Windows.

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili	Supportato su tutte le piattaforme
	cartella quando è impostato su. `false`					

Esempio di input: imposta le autorizzazioni per le cartelle

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

Esempio di input: imposta le autorizzazioni per le cartelle senza modificare le autorizzazioni per tutto il contenuto di una cartella

```
- name: SettingFolderPermissionsNoRecursive
  action: SetFolderPermissions
  inputs:
    - path: /home/UserName/SampleFolder/
      permissions: 777
      recursive: false
```

Output

Nessuna.

Azioni di installazione del software

La sezione seguente descrive i moduli di azione che installano o disinstallano il software.

Requisiti IAM

Se il percorso di download dell'installazione è un URI S3, il ruolo IAM associato al profilo dell'istanza deve disporre dell'autorizzazione per eseguire il modulo di `S3Download` azione. Per concedere

l'autorizzazione richiesta, collega la policy `S3:GetObject` IAM al ruolo IAM associato al profilo dell'istanza e specifica il percorso del bucket. Ad esempio, *`arn:aws:s3:::BucketName/*`*.

Ingressi MSI complessi

Se le stringhe di input contengono virgolette doppie ("), è necessario utilizzare uno dei seguenti metodi per garantire che vengano interpretate correttamente:

- È possibile utilizzare virgolette singole (') all'esterno della stringa, per contenerla, e virgolette doppie («) all'interno della stringa, come illustrato nell'esempio seguente.

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

  In questo caso, se devi usare un apostrofo all'interno della tua stringa, devi evitarlo. Ciò significa usare un'altra virgoletta singola (') prima dell'apostrofo.

- Puoi usare le virgolette doppie («) all'esterno della stringa per contenerla. E puoi evitare le virgolette doppie all'interno della stringa, usando il carattere barra rovesciata (\), come mostrato nell'esempio seguente.

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

Entrambi questi metodi passano il valore `COMPANYNAME="Acme ""Widgets"" and ""Gizmos."""` al msiexec comando.

Moduli di azione per l'installazione del software

- Installa MSI (Windows)
- Disinstalla MSI (Windows)

Installa MSI (Windows)

Il modulo di `InstallMSI` azione installa un'applicazione Windows utilizzando un file MSI. È possibile specificare il file MSI utilizzando un percorso locale, un URI di oggetto S3 o un URL Web. L'opzione di riavvio configura il comportamento di riavvio del sistema.

AWSTOE genera il msiexec comando in base ai parametri di input per il modulo di azione. I valori per i parametri di input `path` (posizione del file MSI) e `logFile` (posizione del file di registro) devono essere racchiusi tra virgolette («).

I seguenti codici di uscita MSI sono considerati corretti:

- 0 (Successo)
- 1614 (ERROR_PRODUCT_UNINSTALLED)
- 1641 (riavvio avviato)
- 3010 (riavvio richiesto)

Input

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili
`path`	Specificate la posizione del file MSI utilizzando uno dei seguenti metodi: • Il percorso del file locale. Il percorso può essere assoluto o relativo • Un URI di oggetto S3 valido. •	Stringa	Sì	N/D	N/D

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili
	Un HTTP/HTTPS URL Web valido (si consiglia HTTPS) che segua lo standard RFC 3986. Le espressioni concatenate sono consentite.				

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili
`reboot`	Configura il comportamento di riavvio del sistema dopo l'esecuzione corretta del modulo di azione. Impostazioni: • `Force`— Avvia il riavvio del sistema dopo la corretta esecuzione del msiexec comando. • `Allow`— Avvia un riavvio del sistema se il msiexec comando restituisce un codice di uscita che indica	Stringa	No	`Allow`	`Allow`, `Force`, `Skip`

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili
	che è necessario riavviare il sistema. • `Skip`— Registra un messaggio informativo nel `console.log` file che indica che il riavvio è stato saltato. Questa opzione impedisce il riavvio, anche se il msiexec comando restituisce un codice di uscita che indica che è necessario un riavvio.				

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili
`logOptions`	Specificare le opzioni da utilizzare per la registrazione dell'installazione MSI. I flag specificati vengono passati al programma di installazione MSI, insieme al parametro della riga di /L comando per abilitare la registrazione. Se non viene specificato alcun flag, AWSTOE utilizza il valore predefinito. Per ulteriori informazioni sulle opzioni di registro per MSI, vedere Opzioni	Stringa	No	`*VX`	`i,w,e,a,r,u,c,m,o,p,v,x,+,!,*`

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili
	della riga di comando nella documentazione del prodotto Microsoft Windows Installer.				
`logFile`	Un percorso assoluto o relativo alla posizione del file di registro. Se il percorso del file di registro non esiste, viene creato. Se il percorso del file di registro non viene fornito, AWSTOE non memorizza il registro di installazione MSI.	Stringa	No	N/D	N/D

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili
`properties`	Coppie chiave-valore della proprietà di registrazione MSI, ad esempio: `TARGETDIR: "C:\target\location"` Nota: non è consentita la modifica delle seguenti proprietà: • `REBOOT="ReallySupress"` • `REINSTALLMODE="ecmus"` • `REINSTALL="ALL"`	Mappa [Stringa] Stringa	No	N/D	N/D

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili
`ignoreAuthenticodeSignatureErrors`	Contrassegna per ignorare gli errori di convalida della firma authenticode per l'installatore specificato nel percorso. Il Get-AuthenticodeSignature comando viene utilizzato per convalidare gli installatori. Impostazioni: • `true`— Gli errori di convalida vengono ignorati e il programma di installazione viene eseguito. •	Boolean	No	`false`	`true`, `false`

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili
	`false`— Gli errori di convalida non vengono ignorati. Il programma di installazione viene eseguito solo quando la convalida ha esito positivo. Questo è il comportamento che segue di default.				

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili
`allowUnsignedInstaller`	Contrassegno per consentire l'esecuzione del programma di installazione non firmato specificato nel percorso. Il Get-AuthenticodeSignature comando viene utilizzato per convalidare gli installatori. Impostazioni: • `true`— Ignora lo `NotSigned` stato restituito dal Get-AuthenticodeSignature comando ed esegue il	Boolean	No	`false`	`true`, `false`

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili
	programma di installazione. • `false`— Richiede la firma del programma di installazione. Gli installatori non firmati non verranno eseguiti. Questo è il comportamento che segue di default.				

Esempi

Gli esempi seguenti mostrano le variazioni della sezione di input per il documento del componente, a seconda del percorso di installazione.

Esempio di input: installazione del percorso del documento locale

```
- 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
```

Esempio di input: installazione del percorso 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
```

Esempio di input: installazione di percorsi web

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

Output

Di seguito è riportato un esempio dell'output del modulo di `InstallMSI` azione.

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

Disinstalla MSI (Windows)

Il modulo di `UninstallMSI` azione consente di rimuovere un'applicazione Windows utilizzando un file MSI. È possibile specificare la posizione del file MSI utilizzando un percorso di file locale, un URI di oggetto S3 o un URL Web. L'opzione di riavvio configura il comportamento di riavvio del sistema.

AWSTOE genera il msiexec comando in base ai parametri di input per il modulo di azione. La posizione del file MSI (`path`) e la posizione del file di registro (`logFile`) sono esplicitamente racchiuse tra virgolette doppie («) durante la generazione del comando. msiexec

I seguenti codici di uscita MSI sono considerati validi:

- 0 (Successo)
- 1605 (ERROR_UNKNOWN_PRODUCT)
- 1614 (ERROR_PRODUCT_UNINSTALLED)
- 1641 (riavvio avviato)
- 3010 (riavvio richiesto)

Input

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili
`path`	Specificate la posizione del file MSI utilizzando uno dei seguenti metodi: • Il percorso del file locale. Il percorso può essere	Stringa	Sì	N/D	N/D

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili
	assoluto o relativo. • Un URI di oggetto S3 valido. • Un HTTP/HTTPS URL Web valido (si consiglia HTTPS) che segua lo standard RFC 3986. Le espressioni concatenate sono consentite.				

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili
`reboot`	Configura il comportamento di riavvio del sistema dopo l'esecuzione corretta del modulo di azione. Impostazioni: • `Force`— Avvia il riavvio del sistema dopo la corretta esecuzione del msiexec comando. • `Allow`— Avvia un riavvio del sistema se il msiexec comando restituisce un codice di uscita che indica	Stringa	No	`Allow`	`Allow`, `Force`, `Skip`

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili
	che è necessario riavviare il sistema. • `Skip`— Registra un messaggio informativo nel `console.log` file che indica che il riavvio è stato saltato. Questa opzione impedisce il riavvio, anche se il msiexec comando restituisce un codice di uscita che indica che è necessario un riavvio.				

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili
`logOptions`	Specificare le opzioni da utilizzare per la registrazione dell'installazione MSI. I flag specificati vengono passati al programma di installazione MSI, insieme al parametro della riga di /L comando per abilitare la registrazione. Se non viene specificato alcun flag, AWSTOE utilizza il valore predefinito. Per ulteriori informazioni sulle opzioni di registro per MSI, vedere Opzioni	Stringa	No	`*VX`	`i,w,e,a,r,u,c,m,o,p,v,x,+,!,*`

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili
	della riga di comando nella documentazione del prodotto Microsoft Windows Installer.				
`logFile`	Un percorso assoluto o relativo alla posizione del file di registro. Se il percorso del file di registro non esiste, viene creato. Se il percorso del file di registro non viene fornito, AWSTOE non memorizza il registro di installazione MSI.	Stringa	No	N/D	N/D

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili
`properties`	Coppie chiave-valore della proprietà di registrazione MSI, ad esempio: `TARGETDIR: "C:\target\location"` Nota: non è consentita la modifica delle seguenti proprietà: • `REBOOT="ReallySupress"` • `REINSTALLMODE="ecmus"` • `REINSTALL="ALL"`	Mappa [Stringa] Stringa	No	N/D	N/D

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili
`ignoreAuthenticodeSignatureErrors`	Contrassegna per ignorare gli errori di convalida della firma authenticode per l'installatore specificato nel percorso. Il Get-AuthenticodeSignature comando viene utilizzato per convalidare gli installatori. Impostazioni: • `true`— Gli errori di convalida vengono ignorati e il programma di installazione viene eseguito. •	Boolean	No	`false`	`true`, `false`

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili
	`false`— Gli errori di convalida non vengono ignorati. Il programma di installazione viene eseguito solo quando la convalida ha esito positivo. Questo è il comportamento che segue di default.				

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili
`allowUnsignedInstaller`	Contrassegno per consentire l'esecuzione del programma di installazione non firmato specificato nel percorso. Il Get-AuthenticodeSignature comando viene utilizzato per convalidare gli installatori. Impostazioni: • `true`— Ignora lo `NotSigned` stato restituito dal Get-AuthenticodeSignature comando ed esegue il	Boolean	No	`false`	`true`, `false`

Nome chiave	Descrizione	Tipo	Richiesto	Valore predefinito	Valori accettabili
	programma di installazione. • `false`— Richiede la firma del programma di installazione. Gli installatori non firmati non verranno eseguiti. Questo è il comportamento che segue di default.				

Esempi

Gli esempi seguenti mostrano le variazioni della sezione di input per il documento del componente, a seconda del percorso di installazione.

Esempio di input: rimuovere l'installazione del percorso del documento locale

```
- 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
```

Esempio di input: rimuovere l'installazione del percorso 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
```

Esempio di input: rimuovere l'installazione del percorso web

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

Output

Di seguito è riportato un esempio dell'output del modulo di `UninstallMSI` azione.

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

Moduli di azione del sistema

La sezione seguente descrive i moduli di azione che eseguono azioni di sistema o aggiornano le impostazioni di sistema.

Moduli di azione del sistema

- Riavvio (Linux, Windows)
- SetRegistry (Windows)
- UpdateOS (Linux, Windows)

Riavvio (Linux, Windows)

Il modulo di azione Reboot riavvia l'istanza. Ha un'opzione configurabile per ritardare l'inizio del riavvio. Per impostazione predefinita, `delaySeconds` è impostato su`0`, il che significa che non vi è alcun ritardo. Il timeout degli step non è supportato per il modulo di azione Reboot, in quanto non si applica al riavvio dell'istanza.

Se l'applicazione viene richiamata dall'agente Systems Manager, consegna il codice di uscita (`3010`per Windows, `194` per Linux) all'agente Systems Manager. Systems Manager Agent gestisce il riavvio del sistema come descritto in Riavvio di un'istanza gestita dagli script.

Se l'applicazione viene richiamata sull'host come processo autonomo, salva lo stato di esecuzione corrente, configura un trigger di esecuzione automatica post-riavvio per rieseguire l'applicazione dopo il riavvio e quindi riavvia il sistema.

Trigger di esecuzione automatica dopo il riavvio:

- Windows. AWSTOE crea una voce di Windows Task Scheduler con un trigger che viene eseguito automaticamente su `SystemStartup`
- Linux. AWSTOE aggiunge un job in crontab che viene eseguito automaticamente dopo il riavvio del sistema.

```
@reboot /download/path/awstoe run --document s3://bucket/key/doc.yaml
```

Questo trigger viene eliminato all'avvio dell'applicazione.

Tentativi

Per impostazione predefinita, il numero massimo di tentativi è impostato su Systems Manager`CommandRetryLimit`. Se il numero di riavvii supera il limite di tentativi, l'automazione fallisce. È possibile modificare il limite modificando il file di configurazione dell'agente Systems Manager (`Mds.CommandRetryLimit`). Vedere Runtime Configuration nell'agente Systems Manager open source.

Per utilizzare il modulo di azione Reboot, per le fasi che contengono il riavvio `exitcode` (ad esempio,`3010`), è necessario eseguire il file binario dell'applicazione come. `sudo user`

Input

Nome chiave	Descrizione	Tipo	Obbligatorio	Predefinita
`delaySeconds`	Ritarda un determinato periodo di tempo prima di iniziare il riavvio.	Numero intero	No	`0`

Esempio di input: fase di riavvio

```
- name: RebootStep
  action: Reboot
  onFailure: Abort
  maxAttempts: 2
  inputs:
    delaySeconds: 60
```

Output

Nessuna.

Una volta completato il modulo Reboot, Image Builder passa alla fase successiva della compilazione.

SetRegistry (Windows)

Il modulo di SetRegistryazione accetta un elenco di input e consente di impostare il valore per la chiave di registro specificata. Se una chiave di registro non esiste, viene creata nel percorso definito. Questa funzionalità si applica solo a Windows.

Input

Nome chiave	Descrizione	Tipo	Richiesto
`path`	Percorso della chiave di registro.	Stringa	Sì
`name`	Nome della chiave di registro.	Stringa	Sì
`value`	Valore della chiave di registro.	String/Number/Array	Sì
`type`	Tipo di valore della chiave di registro.	Stringa	Sì

Prefissi di percorso supportati

- `HKEY_CLASSES_ROOT / HKCR:`
- `HKEY_USERS / HKU:`
- `HKEY_LOCAL_MACHINE / HKLM:`
- `HKEY_CURRENT_CONFIG / HKCC:`
- `HKEY_CURRENT_USER / HKCU:`

Tipi supportati

- `BINARY`
- `DWORD`
- `QWORD`
- `SZ`
- `EXPAND_SZ`

- MULTI_SZ

Esempio di input: imposta i valori delle chiavi di registro

```
- name: SetRegistryKeyValues
  action: SetRegistry
  maxAttempts: 3
  inputs:
    - path: HKLM:\SOFTWARE\MySoftWare
      name: MyName
      value: FirstVersionSoftware
      type: SZ
    - path: HKEY_CURRENT_USER\Software\Test
      name: Version
      value: 1.1
      type: DWORD
```

Output

Nessuna.

UpdateOS (Linux, Windows)

Il modulo di azione UpdateOS aggiunge il supporto per l'installazione degli aggiornamenti di Windows e Linux. Installa tutti gli aggiornamenti disponibili per impostazione predefinita. In alternativa, è possibile configurare un elenco di uno o più aggiornamenti specifici per l'installazione del modulo di azione. È inoltre possibile specificare gli aggiornamenti da escludere dall'installazione.

Se vengono forniti entrambi gli elenchi «includi» ed «escludi», l'elenco di aggiornamenti risultante può includere solo quelli elencati nell'elenco «include» che non sono elencati nell'elenco «escludi».

> **Note**
>
> UpdateOS non supporta Amazon Linux 2023 (AL2023). Ti consigliamo di aggiornare l'AMI di base alla nuova versione fornita con ogni versione. Per altre alternative, consulta Controllare gli aggiornamenti ricevuti dalle versioni principali e secondarie nella Guida per l'utente di Amazon Linux 2023.

- Windows. Gli aggiornamenti vengono installati dalla fonte degli aggiornamenti configurata sul computer di destinazione.

- Linux. L'applicazione verifica la presenza del gestore di pacchetti supportato nella piattaforma Linux e utilizza uno dei due `yum` o il gestore di `apt-get` pacchetti. Se nessuno dei due è supportato, viene restituito un errore. È necessario disporre `sudo` delle autorizzazioni per eseguire il modulo di azione UpdateOS. Se non si dispone delle `sudo` autorizzazioni, viene restituito. `error.Input`

Input

Nome chiave	Descrizione	Tipo	Richiesto
`include`	Per Windows, è possibile specificare quanto segue: • Uno o più articoli della Microsoft Knowledge Base (KB) IDs da includere nell'elenco degli aggiornamenti che possono essere installati. I formati validi sono `KB1234567` o `1234567`. • Un nome di aggiornamento che utilizza un valore jolly (*). I formati validi sono `Security*` o `*Security*` .	Elenco di stringhe	No

Nome chiave	Descrizione	Tipo	Richiesto
	Per Linux, è possibile specificare uno o più pacchetti da includere nell'elenco degli aggiornamenti per l'installazione.		

Nome chiave	Descrizione	Tipo	Richiesto
`exclude`	Per Windows, è possibile specificare quanto segue: • Uno o più articoli IDs della Microsoft Knowledge Base (KB) da includere nell'elenco degli aggiornamenti da escludere dall'installazione. I formati validi sono `KB1234567` o `1234567`. • Un nome di aggiornamento che utilizza un valore wildcard (*). I formati validi sono: `Security*` o `*Security*` . Per Linux, è possibile specificare uno o più pacchetti da escludere dall'elenco degli aggiornamenti per l'installazione.	Elenco di stringhe	No

Esempio di input: aggiungi il supporto per l'installazione degli aggiornamenti Linux

```
- name: UpdateMyLinux
  action: UpdateOS
  onFailure: Abort
  maxAttempts: 3
  inputs:
    exclude:
      - ec2-hibinit-agent
```

Esempio di input: aggiungere il supporto per l'installazione degli aggiornamenti di Windows

```
- name: UpdateWindowsOperatingSystem
  action: UpdateOS
  onFailure: Abort
  maxAttempts: 3
  inputs:
    include:
      - KB1234567
      - '*Security*'
```

Output

Nessuna.

Configurare l'input per il comando AWSTOE run

Per semplificare l'input da riga di comando per il AWSTOE run comando, è possibile includere le impostazioni per i parametri e le opzioni del comando in un file di configurazione di input in formato JSON con estensione di file. `.json` AWSTOE può leggere il file da una delle seguenti posizioni:

- Un percorso di file locale (*`./config.json`*).
- Un bucket S3 ()*`s3://<bucket-path>/<bucket-name>/config.json`*.

Quando si immette il run comando, è possibile specificare il file di configurazione di input utilizzando il --config parametro. Per esempio:

```
awstoe run --config <file-path>/config.json
```

File di configurazione di input

Il file JSON di configurazione di input include coppie chiave-valore per tutte le impostazioni che è possibile fornire direttamente tramite parametri e opzioni di run comando. Se specificate un'impostazione sia nel file di configurazione di input che nel run comando, come parametro o opzione, si applicano le seguenti regole di precedenza:

Regole di precedenza

1. Un'impostazione fornita direttamente al run comando in AWS CLI, tramite un parametro o un'opzione, sovrascrive qualsiasi valore definito nel file di configurazione di input per la stessa impostazione.
2. Un'impostazione nel file di configurazione di input sovrascrive il valore predefinito di un componente.
3. Se non vengono passate altre impostazioni nel documento del componente, è possibile applicare un valore predefinito, se esistente.

Esistono due eccezioni a questa regola: documenti e parametri. Queste impostazioni funzionano in modo diverso nella configurazione di input e come parametri di comando. Se utilizzate il file di configurazione di input, non dovete specificare questi parametri direttamente nel run comando. Questa operazione genererà un errore.

Impostazioni dei componenti

Il file di configurazione di input contiene le seguenti impostazioni. Per semplificare il file, puoi omettere tutte le impostazioni opzionali che non sono necessarie. Tutte le impostazioni sono opzionali se non diversamente specificato.

- cwIgnoreFailures(Boolean) — Ignora gli errori di registrazione nei log. CloudWatch
- cwLogGroup(String) — Il nome dei log. `LogGroup` CloudWatch
- cwLogRegion(String) — La AWS regione che si applica ai CloudWatch registri.
- cwLogStream(String) — Il `LogStream` nome dei CloudWatch log, che indica AWSTOE dove trasmettere il file. `console.log`
- Documents3 BucketOwner (String): l'ID dell'account del proprietario del bucket per i documenti basati su URI S3.
- documents (array di oggetti, obbligatorio) — Un array di oggetti JSON che rappresentano i documenti del componente YAML su cui il comando è in esecuzione. AWSTOE run È necessario specificare almeno un documento componente.

Ogni oggetto è composto dai seguenti campi:

- path (String, required) — La posizione del file del documento del componente YAML. Deve essere uno dei seguenti:
  - Un percorso di file locale (`./component-doc-example.yaml`).
  - Un URI S3 (`s3://bucket/key`).
  - Una versione di build del componente Image Builder ARN (arn:aws:imagebuilder:us-west-th//2021.12.02/1). `2:123456789012 my-example-component`
- parameters (array di oggetti) — Una matrice di oggetti a coppie chiave-valore, ciascuno dei quali rappresenta un parametro specifico del componente che il comando trasmette quando esegue il documento del componente. run I parametri sono facoltativi per i componenti. Il documento del componente può avere o meno dei parametri definiti.

  Ogni oggetto è composto dai seguenti campi:

  - name (String, required) — Il nome del parametro del componente.
  - value (String, required) — Il valore da passare al documento del componente per il parametro denominato.

  Per ulteriori informazioni sui parametri del componente, consulta la sezione Parametri della Usa le variabili nel tuo documento componente personalizzato pagina.

- ExecutonID (String) — Questo è l'ID univoco che si applica all'esecuzione del comando correnterun. Questo ID è incluso nei nomi dei file di output e di registro, per identificare in modo univoco tali file e collegarli all'esecuzione corrente del comando. Se questa impostazione viene omessa, AWSTOE genera un GUID.
- LogDirectory (String) — La directory di destinazione in cui vengono AWSTOE archiviati tutti i file di registro dell'esecuzione di questo comando. Per impostazione predefinita, questa directory si trova all'interno della seguente directory principale:`TOE_<DATETIME>_<EXECUTIONID>`. Se non si specifica la directory di registro, AWSTOE utilizza la directory di lavoro corrente (`.`).
- LogS3 BucketName (String) — Se i log dei componenti sono archiviati in Amazon S3 (consigliato), AWSTOE carica i log delle applicazioni dei componenti nel bucket S3 denominato in questo parametro.
- LogS3 BucketOwner (String): se i log dei componenti sono archiviati in Amazon S3 (consigliato), questo è l'ID dell'account proprietario per il bucket in cui vengono scritti i file di registro. AWSTOE
- LogS3 KeyPrefix (String): se i log dei componenti sono archiviati in Amazon S3 (consigliato), questo è il prefisso della chiave dell'oggetto S3 per la posizione del registro nel bucket.

- parameters (array di oggetti) — Un array di oggetti a coppie chiave-valore che rappresentano parametri che si applicano globalmente a tutti i componenti inclusi nell'esecuzione corrente del comando. run
  - name (String, required) — Il nome del parametro globale.
  - value (String, required) — Il valore da passare a tutti i documenti componenti per il parametro denominato.
- phases (String) — Un elenco separato da virgole che specifica quali fasi eseguire dai documenti dei componenti YAML. Se un documento componente include fasi aggiuntive, queste non verranno eseguite.
- stateDirectory (String) — Il percorso del file in cui sono archiviati i file di tracciamento dello stato.
- trace (Boolean) — Abilita la registrazione dettagliata sulla console.

Esempi

L'esempio seguente mostra un file di configurazione di input che esegue le `test` fasi `build` and per due documenti componenti:, e. `sampledoc.yaml` `conversation-intro.yaml` Ogni documento componente ha un parametro che si applica solo a se stesso ed entrambi utilizzano un parametro condiviso. Il `project` parametro si applica a entrambi i documenti componenti.

```
{
   "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>"
}
```

Risorse di immagini di output di Image Builder

Dopo aver creato risorse di immagini per AMI o immagini di container con Image Builder, è possibile gestirle utilizzando la console Image Builder, tramite l'API Image Builder o con i comandi in. imagebuilder AWS CLI

> **Tip**
>
> Quando hai più risorse dello stesso tipo, l'etichettatura ti aiuta a identificare una risorsa specifica in base ai tag che le hai assegnato. Per ulteriori informazioni sull'etichettatura delle risorse utilizzando i comandi Image Builder in, AWS CLI consultate Aggiunta di tag alle risorse la sezione di questa guida.

Questa sezione spiega come elencare, visualizzare e creare immagini. Per informazioni sui flussi di lavoro relativi alle immagini e su come gestirli, consultaGestisci i flussi di lavoro di creazione e test per le immagini di Image Builder.

Indice

- Elenca le immagini e crea le versioni
- Visualizza i dettagli delle risorse di immagini
- Crea immagini personalizzate con Image Builder
- Importazione ed esportazione di immagini di macchine virtuali con Image Builder
- Importa immagini disco ISO di Windows verificate con Image Builder
- Gestione dei risultati di sicurezza per le immagini di Image Builder
- Pulisci le risorse di Image Builder

Elenca le immagini e crea le versioni

Nella pagina Immagini della console Image Builder, puoi visualizzare gli elenchi di tutte le risorse di immagini di Image Builder che possiedi, che sono condivise con te e a cui hai accesso. I risultati dell'elenco includono alcuni dettagli chiave su tali risorse.

Puoi anche vedere tutte le immagini del tuo account che hanno azioni del flusso di lavoro in sospeso.

Indice

- Elenca immagini
- Elenca le immagini in attesa di azione
- Elenca le versioni di build dell'immagine

Elenca immagini

Questa sezione descrive i diversi modi in cui puoi elencare le informazioni sulle tue immagini.

È possibile utilizzare uno dei seguenti metodi per elencare le risorse di immagini di Image Builder a cui si ha accesso. Per l'azione API, vedere ListImagesnel riferimento all'API EC2 Image Builder. Per la richiesta SDK associata, fai riferimento al link Vedi anche nella stessa pagina.

Indice

- Elenca le immagini nella console
- Elenca le immagini con i comandi AWS CLI

Elenca le immagini nella console

Per aprire la pagina con l'elenco delle immagini nella console, procedi nel seguente modo:

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Scegli Immagini dal pannello di navigazione.

La pagina Immagini nella console è suddivisa in schede, in base alla proprietà dell'immagine o alle azioni del flusso di lavoro in sospeso. Questa sezione copre le prime tre schede che mostrano le immagini di cui sei proprietario o a cui hai accesso.

Scheda Console: Owned by me

Nella scheda Owned by me, puoi utilizzare i seguenti filtri per semplificare i risultati dell'elenco delle immagini.

- Puoi cercare tutto o parte del nome nella barra di ricerca.
- Puoi filtrare le immagini in base alla piattaforma del sistema operativo (Windows, Linux o macOS).
- Puoi filtrare le immagini in base al tipo di output che producono (AMI o immagine contenitore).

- È possibile utilizzare Filter source per trovare immagini importate da una macchina virtuale (VMIE) o da un'immagine disco ISO.

Dopo i controlli del filtro, la scheda Owned by me mostra un elenco di immagini di Image Builder create, con i seguenti dettagli per le risorse elencate:

Nome/versione

I nomi delle risorse di immagine di Image Builder iniziano con il nome e la versione della ricetta da cui sono state create. Seleziona il link per vedere tutte le versioni di build delle immagini correlate.

Tipo

Il tipo di immagine di output che Image Builder crea per questa risorsa di immagine (un AMI o un'immagine contenitore).

Platform (Piattaforma)

La piattaforma del sistema operativo della risorsa immagine, ad esempio «Linux», «Windows» o «macOS».

Fonte dell'immagine

L'origine dell'immagine di base utilizzata da Image Builder per creare questa risorsa di immagini. Viene utilizzato principalmente per filtrare i risultati delle immagini importate da una macchina virtuale (VMIE).

Creation time

La data e l'ora in cui Image Builder ha creato la versione corrente della risorsa immagine.

ARN

L'Amazon Resource Name (ARN) della versione corrente della risorsa immagine.

Scheda Console: condivisa con me

Nella scheda Condivisi con me, puoi utilizzare i seguenti filtri per semplificare i risultati dell'elenco delle immagini.

- Puoi cercare tutto o parte del nome nella barra di ricerca.
- Puoi filtrare le immagini in base alla piattaforma del sistema operativo (Windows, Linux o macOS).
- Puoi filtrare le immagini in base al tipo di output che producono (AMI o immagine contenitore).

- È possibile utilizzare Filter source per trovare immagini importate da una macchina virtuale (VMIE) o da un'immagine disco ISO.

Dopo i controlli del filtro, la scheda Condivisi con me mostra un elenco di immagini di Image Builder che sono state condivise con te, con i seguenti dettagli per le risorse elencate:

Nome dell'immagine

Il nome della risorsa di immagini che è stata condivisa con te. Per utilizzare un'immagine condivisa in una ricetta, seleziona l'opzione Seleziona immagini gestite e modifica l'origine dell'immagine in Immagini condivise con me.

Tipo

Il tipo di immagine di output che Image Builder crea per questa risorsa di immagine (un AMI o un'immagine contenitore).

Versione

<major>La versione della piattaforma del sistema operativo per la risorsa immagine, in genere un campo numerico nel seguente formato:. <minor>. <patch>.

Fonte dell'immagine

L'origine dell'immagine di base utilizzata da Image Builder per creare questa risorsa di immagini, se applicabile. Viene utilizzato principalmente per filtrare i risultati delle immagini importate da una macchina virtuale (VMIE).

Platform (Piattaforma)

La piattaforma del sistema operativo della risorsa immagine, ad esempio «Linux», «Windows» o «macOS».

Creation time

La data e l'ora in cui Image Builder ha creato la versione della risorsa immagine condivisa con te.

Proprietario

Il proprietario della risorsa di immagine condivisa.

ARN

L'Amazon Resource Name (ARN) della versione della risorsa immagine che è stata condivisa con te.

Scheda Console: gestita da Amazon

Nella scheda Gestito da Amazon, puoi utilizzare i seguenti filtri per semplificare i risultati dell'elenco di immagini.

- Puoi cercare tutto o parte del nome nella barra di ricerca.
- Puoi filtrare le immagini in base alla piattaforma del sistema operativo (Windows, Linux o macOS).
- Puoi filtrare le immagini in base al tipo di output che producono (AMI o immagine contenitore).
- È possibile utilizzare Filter source per trovare immagini importate da una macchina virtuale (VMIE) o da un'immagine disco ISO.

Dopo i controlli del filtro, la scheda Managed by Amazon mostra un elenco di immagini gestite da Amazon Image Builder che puoi usare come immagini di base per le tue ricette. Image Builder visualizza i seguenti dettagli per le risorse elencate:

Nome dell'immagine

Il nome dell'immagine gestita. Quando crei una ricetta, l'impostazione predefinita per l'immagine di base è Quick start (gestito da Amazon). Le immagini elencate in questa scheda compongono l'elenco dei nomi delle immagini associato alla piattaforma del sistema operativo scelta per l'immagine di base quando crei una ricetta.

Tipo

Il tipo di immagine di output che Image Builder crea per questa risorsa di immagine (un AMI o un'immagine contenitore).

Versione

<major>La versione della piattaforma del sistema operativo per la risorsa immagine, in genere un campo numerico nel seguente formato:. <minor>. <patch>.

Platform (Piattaforma)

La piattaforma del sistema operativo della risorsa immagine, ad esempio «Linux», «Windows» o «macOS».

Creation time

La data e l'ora in cui Image Builder ha creato la versione della risorsa immagine condivisa con te.

Proprietario

Amazon è proprietaria delle immagini gestite.

ARN

L'Amazon Resource Name (ARN) della versione della risorsa immagine che è stata condivisa con te.

Elenca le immagini con i comandi AWS CLI

Quando esegui il list-images comando in AWS CLI, puoi ottenere un elenco di immagini che possiedi o a cui hai accesso.

Il seguente esempio di comando mostra come utilizzare il list-images comando senza filtri per elencare tutte le risorse di immagini di Image Builder di cui si è proprietari.

Esempio: elenca tutte le immagini

```
aws imagebuilder list-images
```

Output:

```
{
 "requestId": "1abcd234-e567-8fa9-0123-4567b890cd12",
 "imageVersionList": [
  {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/image-recipe-name/1.0.0",
    "name": "image-recipe-name",
    "type": "AMI",
    "version": "1.0.0",
    "platform": "Linux",
    "owner": "123456789012",
    "dateCreated": "2022-04-28T01:38:23.286Z"
  },
  {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/image-recipe-win/1.0.1",
    "name": "image-recipe-win",
    "type": "AMI",
    "version": "1.0.1",
    "platform": "Windows",
```

```
    "owner": "123456789012",
    "dateCreated": "2022-04-28T01:38:23.286Z"
  },
  {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/image-recipe-
macos/1.1.1",
    "name": "image-recipe-macos",
    "type": "AMI",
    "version": "1.1.1",
    "platform": "macOS",
    "owner": "123456789012",
    "dateCreated": "2022-04-28T01:38:23.286Z"
  }
 ]
}
```

Quando si esegue il list-images comando, è possibile applicare filtri per semplificare i risultati, come illustrato nell'esempio seguente. Per ulteriori informazioni su come filtrare i risultati, vedete il comando list-images nel Command Reference.AWS CLI

Esempio: filtro per immagini Linux

```
aws imagebuilder list-images --filters name="platform",values="Linux"
```

Output:

```
{
 "requestId": "1abcd234-e567-8fa9-0123-4567b890cd12",
 "imageVersionList": [
  {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/image-recipe-name/1.0.0",
    "name": "image-recipe-name",
    "type": "AMI",
    "version": "1.0.0",
    "platform": "Linux",
    "owner": "123456789012",
    "dateCreated": "2022-04-28T01:38:23.286Z"
  }
 ]
}
```

Elenca le immagini in attesa di azione

Quando si utilizza l'azione `WaitForAction` graduale nel flusso di lavoro delle immagini, il flusso di lavoro viene sospeso finché non viene inviato un segnale di ripresa dell'elaborazione o l'interruzione del flusso di lavoro. È possibile utilizzare questa operazione se è necessario eseguire un processo esterno prima di continuare. È quindi possibile utilizzare il `SendWorkflowStepAction` per inviare un segnale al passaggio in pausa a `RESUME` o`STOP`. Puoi anche interrompere o riprendere il flusso di lavoro dalla console.

Le seguenti schede mostrano come ottenere un elenco di tutte le risorse di immagini presenti nel tuo account con le fasi del flusso di lavoro attualmente sospese per attendere la ripresa o l'interruzione di un segnale. Le schede illustrano i passaggi della console e il comando. AWS CLI

Puoi anche utilizzare l'API o un SDK per ottenere un elenco di passaggi del flusso di lavoro in attesa di intervento. Per l'azione API, vedere ListWaitingWorkflowStepsnel riferimento all'API EC2 Image Builder. Per la richiesta SDK associata, fai riferimento al link Vedi anche nella stessa pagina.

Console

Per accedere alla scheda In attesa di azione nella console, segui questi passaggi:

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Scegli Immagini dal pannello di navigazione. Si apre la pagina con l'elenco delle immagini.
3. Seleziona la scheda In attesa di azione dalla pagina dell'elenco.
4. (facoltativo) Per interrompere o riprendere un passaggio, seleziona la casella di controllo accanto al nome, quindi scegli Arresta passaggio o Riprendi passaggio. È possibile selezionare più di una casella di controllo per eseguire la stessa azione per tutti i passaggi selezionati.

Dettagli della fase del flusso di lavoro in sospeso

I dettagli del flusso di lavoro per la fase in sospeso includono quanto segue:

- Nome immagine: il nome della risorsa immagine con il passaggio in sospeso. È possibile selezionare il collegamento al nome per visualizzare la pagina di dettaglio dell'immagine.
- Nome della fase in sospeso: il nome della fase del flusso di lavoro in attesa di azione.

- ID di esecuzione del passaggio: identifica in modo univoco l'istanza di runtime della fase del flusso di lavoro. È possibile selezionare l'ID collegato per visualizzare i dettagli di runtime per la fase.
- Inizio fase: data e ora in cui è iniziata l'istanza di runtime della fase del flusso di lavoro.
- ARN del flusso di lavoro: Amazon Resource Name (ARN) del flusso di lavoro con il passaggio in sospeso.
- Azioni: l'azione graduale in stato di attesa.

AWS CLI

Quando esegui il list-waiting-workflow-steps comando in AWS CLI, otterrai un elenco di tutte le immagini del tuo account per le quali sono presenti fasi del flusso di lavoro in attesa di essere eseguite prima di completare il processo di creazione dell'immagine.

L'esempio di comando seguente mostra come utilizzare il list-waiting-workflow-steps comando per elencare tutte le immagini dell'account con le fasi del flusso di lavoro che sono in attesa di essere eseguite.

Esempio: elenca le immagini presenti nel tuo account con le fasi del flusso di lavoro in attesa

```
aws imagebuilder list-waiting-workflow-steps
```

Output:

L'output di questo esempio mostra un'immagine dell'account con un passaggio in attesa di azione.

```
{
    "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"
        }
    ]
```

```
}
```

Elenca le versioni di build dell'immagine

Nella pagina Image build versions della console Image Builder, puoi visualizzare un elenco di versioni di build e dettagli aggiuntivi per una risorsa di immagini di tua proprietà. Puoi anche utilizzare comandi o azioni con l'API Image Builder o AWS CLI elencare SDKs le versioni di build di immagini.

È possibile utilizzare uno dei seguenti metodi per elencare le versioni di build di immagini per le risorse di immagini di cui si è proprietari. Per l'azione API, vedere ListImageBuildVersionsnel riferimento all'API EC2 Image Builder. Per la richiesta SDK associata, fai riferimento al link Vedi anche nella stessa pagina.

Console

Dettagli della versione

I dettagli nella pagina delle versioni di Image build nella console Image Builder includono quanto segue:

- Versione: la versione build della risorsa immagine. Nella console Image Builder, la versione si collega a una pagina di dettaglio dell'immagine.
- Tipo: il tipo di output distribuito da Image Builder quando ha creato questa risorsa immagine (un AMI o un'immagine contenitore).
- Data di creazione: la data e l'ora in cui Image Builder ha creato la versione di creazione dell'immagine.
- Stato dell'immagine: lo stato attuale della versione di generazione dell'immagine. Lo stato può riferirsi alla costruzione o alla disposizione dell'immagine. Ad esempio, durante il processo di creazione, potresti visualizzare lo stato di `Building` o`Distributing`. Per quanto riguarda la disposizione dell'immagine, potresti visualizzare lo stato di `Deprecated` o`Deleted`.
- Motivo dell'errore: il motivo dello stato dell'immagine. La console Image Builder mostra il motivo solo quando la compilazione fallisce (lo stato dell'immagine è ugualе`Failed`).
- Risultati di sicurezza: i risultati aggregati della scansione delle immagini per la versione di generazione dell'immagine di riferimento.
- ARN: Amazon Resource Name (ARN) per la versione di riferimento della risorsa immagine.
- Stream di log: un collegamento ai dettagli del flusso di log per la versione di build dell'immagine di riferimento.

Elenca le versioni

Per elencare le versioni di build delle immagini nella console Image Builder, effettuate le seguenti operazioni:

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Scegli Immagini dal pannello di navigazione. Per impostazione predefinita, l'elenco delle immagini mostra la versione corrente di ciascuna delle immagini che possiedi.
3. Per visualizzare un elenco di tutte le versioni di un'immagine, scegli il link della versione corrente. Il collegamento apre la pagina delle versioni di build dell'immagine che elenca tutte le versioni di build per un'immagine specifica.

AWS CLI

Quando esegui il list-image-build-versions comando in AWS CLI, otterrai un elenco completo delle versioni di build per la risorsa di immagine specificata. È necessario possedere l'immagine per eseguire questo comando.

Il seguente esempio di comando mostra come utilizzare il list-image-build-versions comando per elencare tutte le versioni di build per l'immagine specificata.

Esempio: elenca le versioni di build per un'immagine specifica

```
aws imagebuilder list-image-build-versions --image-version-arn
 arn:aws:imagebuilder:us-west-2:123456789012:image/image-recipe-name/1.0.0
```

Output:

L'output di questo esempio include due versioni di build per la ricetta di immagine specificata.

```
{
  "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**
>
> L'output del list-image-build-versions comando non include i risultati di sicurezza o i flussi di log in questo momento.

Visualizza i dettagli delle risorse di immagini

Nella pagina dei dettagli dell'immagine nella console Image Builder, è possibile visualizzare i dettagli di una risorsa di immagini specifica di cui si è proprietari. Puoi anche utilizzare comandi o azioni con l'API Image Builder o AWS CLI per ottenere dettagli sull'immagine. SDKs

Per ulteriori informazioni sulle risorse che un altro utente Account AWS ha condiviso con te tramite una condivisione di risorse AWS Resource Access Manager (AWS RAM), consulta Accedere alle AWS risorse condivise con te nella Guida per l'AWS RAM utente.

Indice

- Visualizza i dettagli dell'immagine nella console Image Builder
- Per informazioni dettagliate sulla politica relativa alle immagini, consulta il AWS CLI

Visualizza i dettagli dell'immagine nella console Image Builder

La pagina dei dettagli dell'immagine nella console Image Builder include una sezione di riepilogo, con informazioni aggiuntive raggruppate in schede. L'intestazione della pagina è il nome e la versione di build della ricetta che ha creato l'immagine. Se una scheda non si applica all'immagine, la scheda è inattiva e non mostra dati.

Sezioni e schede con i dettagli della console

- Sezione riepilogo
- Scheda Risorse di output
- Scheda di configurazione dell'infrastruttura
- Scheda delle impostazioni di distribuzione

- Scheda Workflow
- Scheda Risultati di sicurezza
- Scheda Tag

Sezione riepilogo

La sezione di riepilogo copre l'intera larghezza della pagina e include i seguenti dettagli. Questi dettagli vengono sempre visualizzati.

Ricetta

Il nome e la versione della ricetta che non includono la versione build. Ad esempio, se la versione build è`sample-linux-recipe | 1.0.1/2`, la ricetta lo è `sample-linux-recipe | 1.0.1` e la versione build è`2`.

Date created (Data di creazione)

La data e l'ora in cui Image Builder ha creato la versione di creazione dell'immagine.

Stato dell'immagine

Lo stato attuale della versione di creazione dell'immagine. Lo stato può riferirsi alla struttura o alla disposizione dell'immagine. Ad esempio, durante il processo di creazione, potresti visualizzare lo stato di `Building` o`Distributing`. Per quanto riguarda la disposizione dell'immagine, potresti visualizzare lo stato di `Deprecated` o`Deleted`.

Motivo del fallimento

Il motivo dello stato dell'immagine. La console Image Builder mostra il motivo solo quando la compilazione fallisce (lo stato dell'immagine è ugualе`Failed`).

Scheda Risorse di output

La scheda Risorse di output elenca i dettagli di output e distribuzione per la risorsa immagine attualmente visualizzata. Le informazioni visualizzate da Image Builder dipendono dal tipo di ricetta utilizzata dalla pipeline per creare l'immagine, come segue.

Ricetta dell'immagine

- Regione: la regione di distribuzione per l'output Amazon Machine Image (AMI) specificata nella colonna Image.

- Immagine: l'ID dell'AMI che Image Builder ha distribuito alla destinazione. Questo ID è collegato alla pagina Amazon Machine Images (AMIs) nella EC2 console Amazon.

> **Note**
>
> Image Builder crea l'AMI dopo aver creato la risorsa dell'immagine di output e prima di distribuirla alla destinazione.

- Nome: il nome dell'AMI che Image Builder ha distribuito alla destinazione.
- Descrizione: la descrizione facoltativa della ricetta dell'immagine utilizzata dalla pipeline per creare la risorsa dell'immagine di output.
- Account: Account AWS il proprietario della risorsa immagine Image Builder attualmente visualizzata.

Ricetta del contenitore

Image Builder visualizza i seguenti dettagli per l'output creato da una ricetta contenitore.

- Regione: la regione di distribuzione per l'immagine del contenitore specificata nella colonna URI dell'immagine.
- URI dell'immagine: l'URI dell'immagine del contenitore di output che Image Builder ha distribuito all'archivio ECR nella regione di destinazione.

> **Note**
>
> Image Builder visualizza una riga per destinazione. L'immagine di output ha sempre almeno una voce da distribuire all'account che ha creato l'immagine. Le destinazioni aggiuntive possono includere distribuzioni tra regioni o AWS Organizations. Account AWS Per ulteriori informazioni, consulta Gestione delle impostazioni di distribuzione di Image Builder.

Scheda di configurazione dell'infrastruttura

La scheda di configurazione dell'infrastruttura mostra le impostazioni EC2 dell'infrastruttura Amazon utilizzate da Image Builder per creare e testare l'immagine attualmente visualizzata. Image Builder visualizza sempre il nome della risorsa di configurazione dell'infrastruttura (Configuration name) e

il relativo Amazon Resource Name (ARN). Se la configurazione dell'infrastruttura imposta i valori, i dettagli aggiuntivi dell'infrastruttura possono includere quanto segue

- Tipi di istanza
- Un profilo di istanza
- Infrastruttura di rete
- Impostazioni dei gruppi di sicurezza
- Una posizione Amazon S3 in cui Image Builder archivia i log delle applicazioni
- Una coppia di EC2 chiavi Amazon per la risoluzione dei problemi
- Un argomento di Amazon SNS per le notifiche degli eventi

Per ulteriori informazioni, consulta Gestione della configurazione dell'infrastruttura Image Builder.

Scheda delle impostazioni di distribuzione

La scheda Impostazioni di distribuzione mostra le impostazioni utilizzate da Image Builder per distribuire le immagini di output. Image Builder visualizza sempre il nome della risorsa di configurazione della distribuzione (Configuration name) e il relativo Amazon Resource Name (ARN). I dettagli di distribuzione aggiuntivi dipendono dal tipo di ricetta utilizzata dalla pipeline di Image Builder per creare l'immagine, come segue:

Ricetta dell'immagine

Se la risorsa di configurazione della distribuzione imposta i valori, i dettagli di distribuzione aggiuntivi possono includere quanto segue:

- Regione: la regione di distribuzione per l'output Amazon Machine Image (AMI).
- Nome AMI di output: il nome dell'AMI che Image Builder ha distribuito alla destinazione.
- Crittografia (chiave KMS): se configurata, utilizzata da Image Builder per crittografare l'immagine per la distribuzione nella regione di destinazione. AWS KMS key
- Account di destinazione per la distribuzione: se hai configurato la distribuzione tra account, questa colonna visualizza un elenco separato da virgole con cui Account AWS condividere l'immagine di output nella regione di destinazione.
- Responsabili con autorizzazione condivisa: un elenco separato da virgole dei responsabili autorizzati a AWS lanciare l'immagine, ad esempio, Account AWS o di gruppi o unità organizzative (). AWS Organizations OUs

> **Note**
>
> Quando concedi il permesso ad altri destinatari di lanciare la tua immagine, l'immagine resta di tua proprietà. AWS fattura sul tuo account tutte le istanze che Amazon EC2 avvia dalla tua immagine.

- Target account per una configurazione di avvio più rapida: è Account AWS qui che EC2 Fast Launch distribuisce istantanee preimpostate per il lancio.
- Configurazioni di licenza associate: la configurazione ARNs della licenza License Manager associata all'AMI nella regione specificata.
- Configurazione del modello di avvio: identifica un modello di EC2 lancio Amazon da utilizzare per un account specifico.
- Imposta la versione predefinita del modello di lancio: imposta il modello di EC2 lancio Amazon specificato come modello di lancio predefinito per il modello di lancio specificato Account AWS.

Ricetta del contenitore

Le distribuzioni di container includono sempre i seguenti dettagli:

- Regione: la regione di distribuzione per l'immagine del contenitore specificata nella colonna URI dell'immagine.
- URI dell'immagine: l'URI dell'immagine del contenitore di output che Image Builder ha distribuito al repository Amazon ECR nella regione di destinazione.

> **Note**
>
> Image Builder visualizza una riga per destinazione. L'immagine di output ha sempre almeno una voce da distribuire all'account che ha creato l'immagine. Le destinazioni aggiuntive possono includere distribuzioni tra regioni o AWS Organizations. Account AWS Per ulteriori informazioni, consulta Gestione delle impostazioni di distribuzione di Image Builder.

Scheda Workflow

I flussi di lavoro definiscono la sequenza di passaggi che Image Builder esegue quando crea una nuova immagine. Tutte le immagini hanno flussi di lavoro di compilazione e test. I contenitori

dispongono di un flusso di lavoro aggiuntivo per la distribuzione. La scheda Workflow mostra i flussi di lavoro applicabili eseguiti da Image Builder per l'immagine.

Filtra i tipi di workflow

Per impostazione predefinita, Image Builder visualizza inizialmente il riepilogo del flusso di lavoro di creazione o importazione e le fasi del flusso di lavoro. Tuttavia, il filtro Workflow mostra tutti i flussi di lavoro in corso o completati per l'immagine. Per visualizzare un flusso di lavoro diverso, selezionalo dall'elenco.

I flussi di lavoro di immagini che producono output AMI possono avere flussi di lavoro di compilazione, importazione o test. I flussi di lavoro dei container che producono output dai container possono avere flussi di lavoro di compilazione, test o distribuzione.

> **Note**
>
> Se il flusso di lavoro non è ancora iniziato, non viene visualizzato nell'elenco. Ad esempio, se la build dell'immagine con flussi di lavoro configurati sia di compilazione che di test è appena stata avviata, il flusso di lavoro di compilazione è l'unico tipo di flusso di lavoro visualizzato nell'elenco. Quando inizia il flusso di lavoro di test, Image Builder lo aggiunge all'elenco.

Dopo il filtro Workflow, il workflow selezionato mostra un riepilogo del runtime che include i seguenti dettagli per ogni tipo di workflow:

Stato del flusso di lavoro

Lo stato di esecuzione corrente di questo flusso di lavoro. I valori possono includere quanto segue:

- In attesa
- Saltato
- In esecuzione
- Completato
- Non riuscito
- R ollback-in-progress
- Rollback completato

ID di esecuzione

Un identificatore univoco assegnato da Image Builder per tenere traccia delle risorse di runtime ogni volta che esegue un flusso di lavoro.

Start (Avvio)

Il timestamp di avvio dell'istanza di runtime di questo flusso di lavoro.

Fine

Il timestamp al termine di questa istanza di runtime del flusso di lavoro.

Passaggi totali

Il numero totale di passaggi del flusso di lavoro. Dovrebbe essere uguale alla somma dei conteggi dei passaggi completati, ignorati e non riusciti.

I passaggi sono riusciti

Un conteggio in fase di esecuzione per il numero di passaggi del flusso di lavoro eseguiti correttamente.

Passaggi non riusciti

Un conteggio in fase di esecuzione per il numero di passaggi del flusso di lavoro non riusciti.

Passaggi saltati

Un conteggio in fase di esecuzione per il numero di passaggi del flusso di lavoro che sono stati saltati.

I dettagli nell'elenco seguente riportano lo stato corrente di tutti i passaggi di questa istanza di runtime del flusso di lavoro. Image Builder visualizza gli stessi dettagli per tutti i tipi di immagini.

Fase #

Un numero che rappresenta l'ordine in cui Image Builder esegue le fasi del flusso di lavoro.

ID del passaggio

Un identificatore univoco per la fase del flusso di lavoro, assegnato in fase di esecuzione.

Stato della fase

Lo stato di esecuzione corrente della fase del flusso di lavoro specificata.

Stato di rollback

Lo stato di rollback corrente se questa istanza di runtime del flusso di lavoro non è riuscita.

Nome della fase

Il nome della fase del flusso di lavoro specificata.

Start (Avvio)

Il timestamp in cui è iniziato il passaggio specificato per questa istanza di runtime del flusso di lavoro.

Fine

Il timestamp al termine del passaggio specificato per questa istanza di runtime del flusso di lavoro.

Scheda Risultati di sicurezza

Se hai attivato la scansione, la scheda Risultati di sicurezza mostra i risultati delle vulnerabilità e delle esposizioni comuni (CVE). Amazon Inspector ha identificato questi risultati sull'istanza di test lanciata da Image Builder per creare la tua nuova immagine. Per garantire che Image Builder acquisisca i risultati dell'immagine, è necessario configurare la scansione come segue:

1. Attiva le scansioni di Amazon Inspector per il tuo account. Per ulteriori informazioni, consulta la sezione Guida introduttiva ad Amazon Inspector nella Amazon Inspector User Guide.
2. Attiva i risultati di sicurezza per la pipeline che crea questa immagine. Quando si attivano i risultati di sicurezza per la pipeline, Image Builder salva un'istantanea dei risultati prima di terminare l'istanza di test. Per ulteriori informazioni, consulta Configurare le scansioni di sicurezza per le immagini di Image Builder in AWS Management Console

La scheda Risultati di sicurezza include i seguenti dettagli per ogni vulnerabilità identificata da Amazon Inspector per la tua immagine.

Gravità

Il livello di gravità del risultato CVE. I valori sono i seguenti:

- Non classificato
- Messaggio informativo
- Bassa

- Media
- Elevata
- Critico

ID risultato

L'identificatore univoco del risultato CVE rilevato da Amazon Inspector per la tua immagine durante la scansione dell'istanza di test. L'ID è collegato alla pagina Risultati di sicurezza > Per vulnerabilità. Per ulteriori informazioni, consulta Gestisci i risultati di sicurezza per le immagini di Image Builder in AWS Management Console.

Origine

La fonte delle informazioni sulla vulnerabilità per il risultato CVE.

Età

Il numero di giorni trascorsi dalla prima osservazione del ritrovamento della tua immagine.

Punteggio Inspector

Il punteggio assegnato da Amazon Inspector per il risultato CVE.

Scheda Tag

La scheda Tag mostra tutti i tag che hai definito per l'immagine.

Per informazioni dettagliate sulla politica relativa alle immagini, consulta il AWS CLI

L'esempio seguente mostra come ottenere i dettagli di una policy relativa alle immagini con il relativo Amazon Resource Name (ARN).

```
aws imagebuilder get-image-policy --image-arn arn:aws:imagebuilder:us-
west-2:123456789012:image/example-image/2019.12.02
```

Crea immagini personalizzate con Image Builder

Esistono diversi modi per creare una nuova immagine di Image Builder. Ad esempio, è possibile utilizzare uno dei seguenti metodi per creare un'immagine con AWS Management Console o AWS

CLI. Puoi anche utilizzare l'azione CreateImageAPI o eseguire una pipeline di compilazione per creare l'immagine. Per la richiesta SDK associata all'azione API, puoi fare riferimento al link Vedi anche per quel comando nel riferimento all'API EC2 Image Builder.

AWS Management Console

Per creare una nuova immagine da una pipeline esistente, puoi eseguire manualmente la pipeline, come segue. È inoltre possibile utilizzare la procedura guidata per la pipeline per creare una nuova immagine partendo da zero. Vedi Procedura guidata per la pipeline: crea AMI oPipeline Wizard: crea un'immagine del contenitore, a seconda del tipo di immagine che desideri creare.

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Scegliete Image pipelines dal pannello di navigazione.
3. Seleziona la casella di controllo accanto al nome della pipeline che desideri eseguire.
4. Per creare l'immagine, selezionate Esegui pipeline dal menu Azioni. Questo avvia la pipeline.

Puoi anche specificare una pianificazione per eseguire la pipeline o utilizzare Amazon EventBridge per eseguire la pipeline in base a regole che configuri.

AWS CLI

Prima di eseguire il create-image comando in AWS CLI, devi creare le seguenti risorse se non esistono già:

Risorse obbligatorie

- Ricetta: è necessario specificare esattamente una ricetta per l'immagine, come segue:

  Immagine (ricetta)

  Specificate l'Amazon Resource Name (ARN) per la risorsa di ricette di immagini con il `--image-recipe-arn` parametro.

  Ricetta del contenitore

  Specificate l'ARN per la risorsa della ricetta del contenitore con il `--container-recipe-arn` parametro.
- Configurazione dell'infrastruttura: specifica l'ARN per la risorsa di configurazione dell'infrastruttura con il `--infrastructure-configuration-arn` parametro.

È inoltre possibile specificare una delle seguenti risorse richieste dall'immagine:

Risorse e configurazione opzionali

- Configurazione della distribuzione: per impostazione predefinita, Image Builder distribuisce la risorsa dell'immagine di output all'account dell'utente nella regione in cui viene eseguito il comando. create-image Per fornire destinazioni o configurazioni aggiuntive per la tua distribuzione, specifica l'ARN per la risorsa di configurazione della distribuzione con il `--distribution-configuration-arn` parametro.
- Scansione delle immagini: per configurare le istantanee dei risultati di Amazon Inspector sull'istanza di test dell'immagine o del contenitore, utilizza `--image-scanning-configuration` il parametro. Per le immagini dei contenitori, specifica anche l'archivio ECR che Amazon Inspector utilizza per le sue scansioni.
- Test delle immagini — Per sopprimere la fase di test di Image Builder, utilizzate `--image-tests-configuration` il parametro. In alternativa, è possibile impostare un timeout per quanto tempo può durare.
- Tag di immagine: utilizzate il `--tags` parametro per aggiungere tag alla risorsa di immagini di output.
- Flussi di lavoro di immagini: se non specificate alcun flusso di lavoro di compilazione o test, Image Builder crea l'immagine con il relativo flusso di lavoro predefinito. Per specificare i flussi di lavoro che hai creato, utilizza il parametro. `--workflows`

> **Note**
>
> Se specifichi flussi di lavoro basati su immagini, devi anche fornire il nome o l'ARN del ruolo IAM utilizzato da Image Builder per eseguire le azioni del flusso di lavoro nel parametro. `--execution-role`

L'esempio seguente mostra come creare un'immagine con il comando create-image. AWS CLI Per ulteriori informazioni, consulta la sezione relativa alle informazioni di riferimento ai comandi di AWS CLI .

Esempio: creare un'immagine di base con distribuzione predefinita

```
aws imagebuilder create-image --image-recipe-arn arn:aws:imagebuilder:us-
west-2:123456789012:image-recipe/simple-recipe-linux/1.0.0 --infrastructure-
```

```
configuration-arn arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-
configuration/simple-infra-config-linux
```

Output:

```
{
"requestId": "1abcd234-e567-8fa9-0123-4567b890cd12",
"imageVersionList": [
    {
        "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/simple-recipe-
linux/1.0.0",
        "name": "simple-recipe-linux",
        ...
    }
]
}
```

Annulla la creazione dell'immagine da AWS CLI

Per annullare la creazione di un'immagine in corso, utilizzate il cancel-image-creation comando seguente:

```
aws imagebuilder cancel-image-creation --image-build-version-arn
 arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-recipe/2019.12.03/1
```

Importazione ed esportazione di immagini di macchine virtuali con Image Builder

Quando si esporta una macchina virtuale dal relativo ambiente di virtualizzazione, tale processo crea un set di uno o più file contenitore su disco che fungono da istantanee dell'ambiente, delle impostazioni e dei dati della macchina virtuale. È possibile utilizzare questi file per importare la macchina virtuale e utilizzarla come immagine di base per le ricette di immagini. Per l'esportazione, è possibile creare file su disco VM come output della creazione dell'immagine personalizzata e distribuire i file.

Image Builder supporta i seguenti formati di file per i contenitori di dischi VM:

- Open Virtualization Archive (OVA)

- Disco della macchina virtuale (VMDK)
- Disco rigido virtuale (VHD/VHDX)
- Raw

L'importazione utilizza i dischi per creare una risorsa di immagine Amazon Machine Image (AMI) e una risorsa di immagine Image Builder, entrambe le quali possono fungere da immagine di base per la tua ricetta di immagine personalizzata. I dischi VM devono essere archiviati in bucket S3 per l'importazione. In alternativa, puoi importare da uno snapshot EBS esistente.

Nella console Image Builder, è possibile importare direttamente l'immagine e quindi utilizzare l'immagine di output o l'AMI nelle ricette, oppure è possibile specificare i parametri di importazione durante la creazione della ricetta o della versione della ricetta. Per ulteriori informazioni sull'importazione come parte della ricetta dell'immagine, consulta. Configurazione di importazione delle macchine virtuali

Importazione di una macchina virtuale in Image Builder

Image Builder si integra con l' Import/Export API Amazon EC2 VM per consentire l'esecuzione asincrona del processo di importazione in background. Image Builder fa riferimento all'ID dell'attività dall'importazione della macchina virtuale per monitorarne l'avanzamento e crea una risorsa immagine Image Builder come output. Ciò consente di fare riferimento alla risorsa immagine Image Builder nelle ricette prima del termine dell'importazione della macchina virtuale.

Console

Per importare una macchina virtuale con la console Image Builder, segui questi passaggi:

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Scegli Immagini dal pannello di navigazione.
3. Per aprire la finestra di dialogo di importazione, scegli Importa immagine.
4. Immettete le seguenti informazioni generali:
   - Specificate un nome univoco per l'immagine.
   - Specificate una versione per l'immagine di base. Utilizza il seguente formato: *major.minor.patch*.
5. Scegli il tipo di importazione: importazione VM.

6. Fornisci i dettagli per ciascuna delle seguenti sezioni nella pagina Importa immagine. Quindi scegli Importa immagine quando hai finito.

Sistema operativo con immagine di base

1. Seleziona l'opzione Image Operating System (OS) che corrisponde alla tua piattaforma VM OS.
2. Seleziona la versione del sistema operativo che corrisponde alla versione della tua macchina virtuale dall'elenco.

Configurazione di importazione VM

1. Quando si esporta la macchina virtuale dal relativo ambiente di virtualizzazione, tale processo crea un set di uno o più file contenitore del disco. Questi agiscono come istantanee dell'ambiente, delle impostazioni e dei dati della macchina virtuale. È possibile utilizzare questi file per importare la macchina virtuale come immagine di base per la ricetta dell'immagine. Per ulteriori informazioni sull'importazione VMs in Image Builder, vedere. Importa ed esporta immagini VM

   Per specificare la posizione della fonte di importazione, procedi nel seguente modo:

   Fonte di importazione

   Specificare la fonte per il primo contenitore o istantanea di immagini VM da importare nella sezione Contenitore disco 1.

   a. Fonte: può essere un bucket S3 o un'istantanea EBS.
   b. Seleziona la posizione S3 del disco: inserisci la posizione in Amazon S3 in cui sono archiviate le immagini del disco. Per cercare la posizione, scegli Browse S3.
   c. Per aggiungere un contenitore per dischi, scegli Aggiungi contenitore disco.

2. Ruolo IAM

Per associare un ruolo IAM alla configurazione di importazione della macchina virtuale, seleziona il ruolo dall'elenco a discesa dei ruoli IAM o scegli Crea nuovo ruolo per crearne uno nuovo. Se crei un nuovo ruolo, la pagina della console IAM Roles si apre in una scheda separata.

3. Impostazioni avanzate: facoltative

   Le seguenti impostazioni sono opzionali. Con queste impostazioni, è possibile configurare la crittografia, le licenze, i tag e altro per l'immagine di base creata dall'importazione.

   Architettura dell'immagine di base

   Per specificare l'architettura della sorgente di importazione delle macchine virtuali, selezionare un valore dall'elenco Architettura.

   Crittografia

   Se le immagini del disco della macchina virtuale sono crittografate, è necessario fornire una chiave da utilizzare per il processo di importazione. Per specificare una chiave KMS per l'importazione, selezionare un valore dall'elenco Encryption (chiave KMS). L'elenco contiene le chiavi KMS a cui il tuo account ha accesso nella regione corrente.

   Gestione delle licenze

   Quando si importa una macchina virtuale, il processo di importazione rileva automaticamente il sistema operativo della macchina virtuale e applica la licenza appropriata all'immagine di base. A seconda della piattaforma del sistema operativo, i tipi di licenza sono i seguenti:

   - Licenza inclusa: all'immagine di base viene applicata una AWS licenza appropriata per la piattaforma in uso.
   - Bring your own license (BYOL): conserva la licenza della macchina virtuale, se applicabile.

   Per allegare le configurazioni di licenza create con AWS License Manager all'immagine di base, selezionatele dall'elenco dei nomi di configurazione della licenza. Per ulteriori informazioni su License Manager, vedere Working with AWS License Manager

> **Note**
>
> - Le configurazioni delle licenze contengono regole di licenza basate sui termini dei contratti aziendali.
> - Linux supporta solo le licenze BYOL.

Tag (immagine di base)

I tag utilizzano coppie chiave-valore per assegnare testo ricercabile alla risorsa Image Builder. Per specificare i tag per l'immagine di base importata, inserite le coppie chiave-valore utilizzando le caselle Chiave e Valore.

Per aggiungere un tag, scegli Add tag (Aggiungi tag). Per rimuovere un tag, scegli Remove (Rimuovi).

AWS CLI

Per importare una macchina virtuale dai dischi in un'AMI e creare una risorsa immagine Image Builder a cui puoi fare riferimento subito, segui questi passaggi dal: AWS CLI

1. Avvia un'importazione di macchine virtuali, con il Import/Export import-image comando Amazon EC2 VM in. AWS CLI Prendi nota dell'ID dell'attività restituito nella risposta al comando. Servirà per la fase successiva. Per ulteriori informazioni, vedere Importazione di una macchina virtuale come immagine utilizzando VM Import/Export nella Guida per l'utente della macchina virtuale. Import/Export

2. Creazione di un file JSON di input nella CLI

   Per semplificare il comando Image import-vm-image Builder utilizzato in AWS CLI, creiamo un file JSON che contiene tutta la configurazione di importazione che vogliamo passare al comando.

> **Note**
>
> La convenzione di denominazione per i valori dei dati nel file JSON segue lo schema specificato per i parametri di richiesta operativa dell'API Image Builder. Per esaminare i parametri di richiesta dell'operazione API, consulta l'ImportVmImageoperazione nell'EC2 Image Builder API Reference.
> Per fornire i valori dei dati come parametri della riga di comando, fare riferimento ai nomi dei parametri specificati nel AWS CLI Command Reference. Al comando Image Builder come import-vm-image opzioni.

Ecco un riepilogo dei parametri specificati in questo esempio:

- name (string, required) — Il nome della risorsa immagine Image Builder da creare come output dall'importazione.
- semanticVersion <major>(string, required) — La versione semantica dell'immagine di output che specifica la versione nel seguente formato, con valori numerici in ogni posizione per indicare una versione specifica:. <minor>. <patch>. Ad esempio, `1.0.0`. Per ulteriori informazioni sul controllo delle versioni semantiche per le risorse di Image Builder, vedere. Versionamento semantico in Image Builder
- description (string) — La descrizione della ricetta dell'immagine.
- platform (string, required) — La piattaforma del sistema operativo per la macchina virtuale importata.
- vmImportTaskId (stringa, obbligatorio) — Il `ImportTaskId` (AWS CLI) dal processo di importazione di Amazon EC2 VM. Image Builder monitora il processo di importazione per inserire l'AMI che crea e crea una risorsa di immagini Image Builder che può essere utilizzata immediatamente nelle ricette.
- tags (mappa di stringhe): i tag sono coppie chiave-valore allegate alle risorse di importazione. Sono consentite fino a 50 coppie chiave-valore.

Salvare il file con nome`import-vm-image.json`, da utilizzare nel comando Image Builderimport-vm-image.

```
{
    "name": "example-request",
```

```
    "semanticVersion": "1.0.0",
    "description": "vm-import-test",
    "platform": "Linux",
    "vmImportTaskId": "import-ami-01ab234567890cd1e",
    "tags": {
     "Usage": "VMIE"
    }
}
```

3. Importa l'immagine

   Esegui il import-vm-image comando, con il file che hai creato come input:

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

   > **Note**
   >
   > - È necessario includere l'annotazione `file://` all'inizio del percorso del file JSON.
   > - Il percorso del file JSON deve seguire la convenzione appropriata per il sistema operativo di base in cui si esegue il comando. Ad esempio, Windows utilizza la barra rovesciata (\) per fare riferimento al percorso della directory, mentre Linux e macOS utilizzano la barra (/).

Distribuisci i dischi VM dalla tua immagine creata da AWS CLI

È possibile impostare la distribuzione dei file in formato disco VM supportati nei bucket S3 nelle regioni di destinazione come parte del normale processo di creazione delle immagini, utilizzando le configurazioni di distribuzione di Image Builder in. AWS CLI Per ulteriori informazioni, consulta Creare impostazioni di distribuzione per i dischi VM in uscita dal AWS CLI.

Importa immagini disco ISO di Windows verificate con Image Builder

Un file ISO del sistema operativo Windows è un file di immagine del disco che contiene il pacchetto di installazione completo per una versione specifica del sistema operativo Windows. Microsoft fornisce file ISO ufficiali del sistema operativo Windows da scaricare, direttamente dal proprio sito Web o

tramite rivenditori autorizzati. È importante assicurarsi di ottenere i file ISO da una fonte attendibile e legittima per evitare potenziali malware o versioni non autorizzate.

EC2 Image Builder utilizza il `build-image-from-iso` flusso di lavoro di importazione per importare il file su disco ISO e crearne un volume secondario. Una volta completata la configurazione, Image Builder scatta un'istantanea del volume creato dall'importazione e la utilizza per creare un'Amazon Machine Image (AMI).

Sistemi operativi supportati per l'importazione di immagini disco ISO

Image Builder supporta le seguenti immagini del disco ISO del sistema operativo Windows:

- Windows 11 Enterprise versione 24H2
- Windows 11 Enterprise versione 23H2
- Windows 11 Enterprise versione 22H2

Image Builder non supporta le seguenti immagini del disco ISO del sistema operativo Windows:

- Immagini LTSC (Long-Term Servicing Channel)
- Immagini del disco ISO create con Windows Media Creation Tool
- Immagini di valutazione

Prerequisiti per importare un'immagine disco ISO

Per importare un'immagine disco ISO, è necessario innanzitutto soddisfare i seguenti prerequisiti:

- Il sistema operativo dell'immagine del disco deve essere supportato da Image Builder. Per un elenco dei sistemi operativi supportati, vedereSistemi operativi supportati per l'importazione di immagini disco ISO.
- Per assicurarti di poter importare l'immagine ISO, scaricala dall'interfaccia di amministrazione di Microsoft 365.
- Prima di poter eseguire il processo di importazione, devi caricare il file del disco ISO su Amazon S3 nello stesso Account AWS luogo in Regione AWS cui viene eseguita l'importazione.
- L'estensione del file fa distinzione tra maiuscole e minuscole per il processo di importazione e deve esserlo `.ISO`. Se l'estensione del file è minuscola, puoi eseguire uno dei seguenti comandi per rinominarla:

Command

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

PowerShell

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

- Le licenze Microsoft non sono incluse automaticamente nell'importazione. È necessario portare la propria licenza (BYOL). Per ulteriori informazioni sulle licenze per il software Microsoft, consulta la pagina Licenze su Amazon Web Services e Microsoft Frequently Asked Questions.
- Il processo di importazione utilizza due ruoli IAM separati, come segue:

  Ruolo di esecuzione

  Questo ruolo consente a Image Builder di effettuare Servizi AWS chiamate per conto dell'utente. È possibile specificare il ruolo AWSServiceRoleForImageBuildercollegato al servizio, che include le autorizzazioni necessarie per il ruolo di esecuzione, oppure è possibile creare un ruolo personalizzato.

  Ruolo del profilo dell'istanza

  Questo ruolo concede l'autorizzazione per le azioni eseguite dal servizio sull' EC2 istanza. È possibile specificare un ruolo del profilo di istanza nella risorsa di configurazione dell'infrastruttura. Allega le seguenti politiche gestite al ruolo del profilo dell'istanza per assicurarti di disporre di tutte le autorizzazioni necessarie per il processo di importazione.

  - EC2InstanceProfileForImageBuilder
  - AmazonSSMManagedInstanceCore

  Per ulteriori informazioni, consulta Gestione della configurazione dell'infrastruttura Image Builder.

Importazione di un'immagine disco ISO in Image Builder

Prima di iniziare il processo di importazione, assicurati di aver soddisfatto tutti i. Prerequisiti

Il processo di importazione installa inoltre i seguenti software e driver sull'immagine:

- EC2Avvia v2
- AWS Systems Manager agente
- AWS NVMe autista
- AWS driver di rete ENA
- AWS Driver seriale PCI
- EC2 Utilità Windows

Console

Per importare un'immagine disco ISO con la console Image Builder, procedi nel seguente modo:

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Scegli Immagini dal pannello di navigazione.
3. Per aprire la finestra di dialogo di importazione, scegli Importa immagine.
4. Inserite le seguenti informazioni generali:
   - Specificate un nome univoco per l'immagine.
   - Specificate una versione per l'immagine di base. Utilizza il seguente formato: *major*.*minor*.*patch*.
5. Scegli il tipo di importazione: importazione ISO.
6. Inserisci i seguenti dettagli di configurazione di importazione ISO. Quindi scegli Importa immagine quando hai finito.
   - URI S3: inserisci la posizione in cui è archiviato il file del disco ISO. Per cercare il file, scegli Browse S3.
   - Ruolo IAM: per associare un ruolo IAM alla configurazione di importazione, seleziona il ruolo dall'elenco a discesa dei ruoli IAM o scegli Crea nuovo ruolo per crearne uno nuovo. Se crei un nuovo ruolo, la pagina della console IAM Roles si apre in una scheda separata.

     Puoi specificare il ruolo AWSServiceRoleForImageBuildercollegato al servizio oppure puoi specificare il tuo ruolo personalizzato per l'accesso al servizio.
7. Facoltativamente, è possibile aggiungere tag alla risorsa di immagini Image Builder. Ciò non aggiunge i tag all'AMI.

8. La configurazione dell'infrastruttura ISO definisce le impostazioni per l'istanza che Image Builder avvia per ospitare il processo di importazione. È possibile utilizzare una configurazione dell'infrastruttura creata da Image Builder, in base ai valori predefiniti del servizio, oppure è possibile utilizzare una configurazione di infrastruttura esistente. Per ulteriori informazioni, consulta Gestione della configurazione dell'infrastruttura Image Builder.

   Per creare una nuova configurazione dell'infrastruttura, selezionare Crea configurazione dell'infrastruttura. Si apre in una scheda separata. Quando hai finito di creare la nuova risorsa, puoi tornare alla configurazione di importazione e scegliere Usa la configurazione dell'infrastruttura esistente.
9. Per avviare il processo di importazione, scegli Importa immagine.

Una volta completata l'importazione, l'immagine viene visualizzata nell'elenco delle immagini di cui sei proprietario. Per ulteriori dettagli, consulta Elenca immagini.

AWS CLI

Questo esempio mostra come importare un'immagine da un file disco ISO e crearne un'AMI con AWS CLI.

Ecco un riepilogo dei parametri specificati in questo esempio:

- name (string, required) — Il nome della risorsa immagine Image Builder da creare come output dall'importazione.
- semanticVersion <major>(string, required) — La versione semantica dell'immagine di output che specifica la versione nel seguente formato, con valori numerici in ogni posizione per indicare una versione specifica:. <minor>. <patch>. Ad esempio, `1.0.0`. Per ulteriori informazioni sul controllo delle versioni semantiche per le risorse di Image Builder, vedere. Versionamento semantico in Image Builder
- description (string) — La descrizione della ricetta dell'immagine.
- ExecutionRole (string) — Il nome o Amazon Resource Name (ARN) per il ruolo IAM che concede a Image Builder l'accesso per eseguire azioni del flusso di lavoro per importare un'immagine da un file ISO Microsoft. Puoi specificare il ruolo AWSServiceRoleForImageBuildercollegato al servizio oppure puoi specificare il tuo ruolo personalizzato per l'accesso al servizio.
- platform (string, required): la piattaforma del sistema operativo per l'immagine del disco ISO. I valori validi includonoWindows.

- osVersion (stringa, obbligatorio) — La versione del sistema operativo per l'immagine disco ISO. I valori validi includonoMicrosoft Windows 11.
- infrastructureConfigurationArn(stringa, obbligatorio) — L'Amazon Resource Name (ARN) della risorsa di configurazione dell'infrastruttura utilizzata per avviare l'EC2 istanza su cui è costruita l'immagine ISO.
- uri (stringa, obbligatorio) — L'URI del file su disco ISO archiviato in Amazon S3.

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

Una volta completata l'importazione, l'immagine viene visualizzata nell'elenco delle immagini di cui sei proprietario. Per ulteriori dettagli, consulta Elenca immagini.

PowerShell

Questo esempio mostra come importare un'immagine da un file disco ISO e crearne un'AMI con PowerShell.

Ecco un riepilogo dei parametri specificati in questo esempio:

- name (string, required) — Il nome della risorsa immagine Image Builder da creare come output dall'importazione.
- semanticVersion <major>(string, required) — La versione semantica dell'immagine di output che specifica la versione nel seguente formato, con valori numerici in ogni posizione per indicare una versione specifica:. <minor>. <patch>. Ad esempio, `1.0.0`. Per ulteriori informazioni sul controllo delle versioni semantiche per le risorse di Image Builder, vedere. Versionamento semantico in Image Builder
- description (string) — La descrizione della ricetta dell'immagine.
- ExecutionRole (string) — Il nome o Amazon Resource Name (ARN) per il ruolo IAM che concede a Image Builder l'accesso per eseguire azioni del flusso di

lavoro per importare un'immagine da un file ISO Microsoft. Puoi specificare il ruolo AWSServiceRoleForImageBuildercollegato al servizio oppure puoi specificare il tuo ruolo personalizzato per l'accesso al servizio.

- platform (string, required): la piattaforma del sistema operativo per l'immagine del disco ISO. I valori validi includonoWindows.
- osVersion (stringa, obbligatorio) — La versione del sistema operativo per l'immagine disco ISO. I valori validi includonoMicrosoft Windows 11.
- infrastructureConfigurationArn(stringa, obbligatorio) — L'Amazon Resource Name (ARN) della risorsa di configurazione dell'infrastruttura utilizzata per avviare l'EC2 istanza su cui è costruita l'immagine ISO.
- uri (stringa, obbligatorio) — L'URI del file su disco ISO archiviato in Amazon S3.

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

Una volta completata l'importazione, l'immagine viene visualizzata nell'elenco delle immagini di cui sei proprietario. Per ulteriori dettagli, consulta Elenca immagini.

Gestione dei risultati di sicurezza per le immagini di Image Builder

Quando attivi la scansione di sicurezza con Amazon Inspector, esegue una scansione continua delle immagini delle macchine e delle istanze in esecuzione nel tuo account alla ricerca di vulnerabilità del sistema operativo e del linguaggio di programmazione. Se attivata, la scansione di sicurezza è automatica e Image Builder può salvare un'istantanea dei risultati dall'istanza di test quando si crea una nuova immagine. Amazon Inspector è un servizio a pagamento.

Quando Amazon Inspector rileva vulnerabilità nel software o nelle impostazioni di rete, intraprende le seguenti azioni:

- Ti avvisa che c'è stata una scoperta.
- Valuta la gravità del risultato. La classificazione di gravità classifica le vulnerabilità per aiutarvi a dare priorità ai risultati e include i seguenti valori:
  - Non classificato
  - Messaggio informativo
  - Bassa
  - Media
  - Elevata
  - Critico
- Fornisce informazioni sulla scoperta e collegamenti a risorse aggiuntive per ulteriori dettagli.
- Offre indicazioni sulla correzione per aiutarvi a risolvere i problemi che hanno generato il risultato.

Configurare le scansioni di sicurezza per le immagini di Image Builder in AWS Management Console

Se hai attivato Amazon Inspector per il tuo account, Amazon Inspector analizza automaticamente EC2 le istanze avviate da Image Builder per creare e testare una nuova immagine. Queste istanze hanno una durata breve durante il processo di creazione e test e i risultati in genere scadono non appena tali istanze vengono chiuse. Per aiutarti a esaminare e correggere i risultati della tua nuova immagine, Image Builder può facoltativamente salvare come snapshot tutti i risultati identificati da Amazon Inspector sull'istanza di test durante il processo di creazione.

Passaggio 1: attiva le scansioni di sicurezza di Amazon Inspector per il tuo account

Per attivare le scansioni di sicurezza di Amazon Inspector per il tuo account dalla console Image Builder, segui questi passaggi:

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Scegli le impostazioni di scansione di sicurezza dal pannello di navigazione. Si apre la finestra di dialogo di scansione di sicurezza.

   La finestra di dialogo mostra lo stato della scansione dell'account. Se Amazon Inspector è già attivato per il tuo account, lo stato mostra Abilitato.
3. Segui i passaggi 1 e 2 delle istruzioni per attivare la scansione di Amazon Inspector.

> **Note**
>
> Amazon Inspector prevede dei costi. Per ulteriori informazioni, consulta Prezzi di Amazon Inspector.

Se hai attivato la scansione per la pipeline, Image Builder scatta un'istantanea dei risultati per l'istanza di build quando crei una nuova immagine. In questo modo, è possibile accedere ai risultati dopo che Image Builder ha terminato l'istanza di compilazione.

Passaggio 2: configura la pipeline per salvare istantanee per rilevare eventuali vulnerabilità

Per configurare le istantanee di rilevamento delle vulnerabilità per la tua pipeline, procedi nel seguente modo:

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Scegli Image pipelines dal pannello di navigazione.
3. Scegliete uno dei seguenti metodi per specificare i dettagli della tubazione:

   Create una nuova pipeline

   1. Dalla pagina Image pipelines, scegliete Crea pipeline di immagini. Viene visualizzata la pagina Specificare i dettagli della pipeline nella procedura guidata della pipeline.

   Aggiorna una tubazione esistente

   1. Dalla pagina Image pipelines, scegliete il link Pipeline name per la pipeline che desiderate aggiornare. Verrà aperta la pagina dei dettagli della pipeline.

      > **Note**
      >
      > In alternativa, puoi selezionare la casella di controllo accanto al nome della pipeline che desideri aggiornare e quindi scegliere Visualizza dettagli.

   2. Dalla pagina dei dettagli della pipeline, seleziona Modifica pipeline dal menu Azione. Verrà visualizzata la pagina Modifica pipeline.
4. Nella sezione Generale della procedura guidata della pipeline o della pagina Modifica tubazione, selezionate la casella di controllo Abilita scansione di sicurezza.

> **Note**
>
> Se desideri disattivare le istantanee in un secondo momento, puoi modificare la pipeline per deselezionare la casella di controllo. Ciò non disattiva la scansione di Amazon Inspector per il tuo account. Per disattivare la scansione di Amazon Inspector, consulta la sezione Disattivazione di Amazon Inspector nella Guida per l'utente di Amazon Inspector.

Gestisci i risultati di sicurezza per le immagini di Image Builder in AWS Management Console

Le pagine con l'elenco dei risultati di sicurezza mostrano informazioni di alto livello sui risultati delle tue risorse, con visualizzazioni basate su diversi filtri che puoi applicare. Ogni visualizzazione include le seguenti opzioni nella parte superiore per modificare la visualizzazione:

- Tutti i risultati di sicurezza: questa è la visualizzazione predefinita se si sceglie la pagina Risultati di sicurezza dal riquadro di navigazione della console Image Builder.
- Per vulnerabilità: questa visualizzazione mostra un elenco di alto livello di tutte le risorse di immagini presenti nell'account che presentano dei risultati. Il Finding ID è collegato a informazioni più dettagliate sul risultato. Queste informazioni vengono visualizzate in un pannello che si apre sul lato destro della pagina. Il pannello include le seguenti informazioni:
  - Una descrizione dettagliata del risultato.
  - Una scheda con i dettagli del ritrovamento. Questa scheda include una panoramica dei risultati, i pacchetti interessati, un riepilogo dei consigli per la correzione, i dettagli sulle vulnerabilità e le vulnerabilità correlate. Il Vulnerability ID si collega a informazioni dettagliate sulle vulnerabilità nel National Vulnerability Database.
  - Una scheda di ripartizione dei punteggi. Questa scheda include un side-by-side confronto tra i punteggi CVSS e Amazon Inspector in modo da poter vedere dove Amazon Inspector ha modificato un punteggio, se applicabile.
- Per pipeline di immagini: questa visualizzazione mostra il numero di risultati per ogni pipeline di immagini nel tuo account. Image Builder visualizza i conteggi per i risultati di gravità media e superiore, oltre a un totale per tutti i risultati. Tutti i dati dell'elenco sono collegati nel modo seguente:

- La colonna Image pipeline name rimanda alla pagina di dettaglio della pipeline di immagini specificata.
- I collegamenti alla colonna del livello di gravità aprono la visualizzazione Tutti i risultati di sicurezza, filtrata in base al nome e al livello di gravità della pipeline di immagini associata.

Puoi anche utilizzare i criteri di ricerca per affinare i risultati.

- Per immagine: questa visualizzazione mostra il numero di risultati per ogni immagine creata nel tuo account. Image Builder visualizza i conteggi per i risultati di gravità media e superiore, oltre a un totale per tutti i risultati. Tutti i dati dell'elenco sono collegati nel modo seguente:
  - La colonna Nome immagine rimanda alla pagina di dettaglio dell'immagine per la build dell'immagine specificata. Per ulteriori informazioni, consulta Visualizza i dettagli delle risorse di immagini.
  - I collegamenti alla colonna del livello di gravità aprono la visualizzazione Tutti i risultati di sicurezza, filtrata in base al nome e al livello di gravità della build dell'immagine associati.

  Puoi anche utilizzare i criteri di ricerca per affinare i risultati.

Image Builder mostra i seguenti dettagli nella sezione Elenco risultati della visualizzazione predefinita Tutti i risultati di sicurezza.

Gravità

Il livello di gravità del risultato CVE. I valori sono i seguenti:

- Non classificato
- Messaggio informativo
- Bassa
- Media
- Elevata
- Critico

ID risultato

L'identificatore univoco del risultato CVE rilevato da Amazon Inspector per la tua immagine durante la scansione dell'istanza di build. L'ID è collegato alla pagina Risultati di sicurezza > Per vulnerabilità.

Immagine ARN

L'Amazon Resource Name (ARN) per l'immagine con il risultato specificato nella colonna Finding ID.

Pipeline

La pipeline che ha creato l'immagine specificata nella colonna Image ARN.

Descrizione

Una breve descrizione del risultato.

Punteggio Inspector

Il punteggio assegnato da Amazon Inspector per il risultato CVE.

Riparazione

Collegamenti ai dettagli sulla linea d'azione consigliata per correggere il problema.

Data di pubblicazione

La data e l'ora in cui questa vulnerabilità è stata aggiunta per la prima volta al database del fornitore.

Pulisci le risorse di Image Builder

Per evitare addebiti imprevisti, assicuratevi di ripulire le risorse e le pipeline create sulla base degli esempi riportati in questa guida. Per ulteriori informazioni sull'eliminazione di risorse in Image Builder, vedere. Eliminare risorse Image Builder obsolete o inutilizzate

Gestione delle politiche del ciclo di vita per le immagini di Image Builder

Quando si creano immagini personalizzate, è importante disporre di un piano per ritirarle prima che diventino obsolete. Le pipeline di Image Builder possono applicare automaticamente aggiornamenti e patch di sicurezza. Tuttavia, ogni build crea una nuova versione dell'immagine e tutte le risorse associate che distribuisce. Le versioni precedenti rimangono nel tuo account finché non le elimini manualmente o crei uno script per eseguire l'operazione.

Con le policy di gestione del ciclo di vita di Image Builder, è possibile automatizzare il processo di deprecazione, disabilitazione ed eliminazione delle immagini obsolete e delle relative risorse associate. Le risorse associate possono includere immagini di output distribuite ad altre organizzazioni e unità organizzative () in tutto il mondo Account AWS. OUs Regioni AWS Sei tu a definire le regole su come e quando eseguire ogni fase del processo del ciclo di vita e quali passaggi includere nella tua politica.

Vantaggi della gestione automatizzata del ciclo di vita

I vantaggi complessivi della gestione automatizzata del ciclo di vita includono quanto segue:

- Semplifica la gestione del ciclo di vita delle immagini personalizzate con un modo automatizzato per ritirare le immagini e le risorse associate.
- Aiuta a prevenire i rischi di conformità derivanti dall'utilizzo di immagini obsolete per il lancio di nuove istanze.
- Mantiene aggiornati gli inventari delle immagini rimuovendo le immagini obsolete.
- Può ridurre i costi di archiviazione e trasferimento dei dati rimuovendo facoltativamente le risorse associate per le immagini che vengono eliminate.

Realizza risparmi sui costi

L'utilizzo di EC2 Image Builder per creare AMI o immagini di container personalizzate è gratuito. Tuttavia, per gli altri servizi utilizzati nel processo si applicano i prezzi standard. Rimuovendo immagini inutilizzate o obsolete e le relative risorse dal vostro Account AWS computer, potete risparmiare tempo e denaro nei seguenti modi:

- Riduci il tempo necessario per applicare le patch alle immagini esistenti quando non stai correggendo anche immagini inutilizzate o obsolete.

- Per le risorse di immagini AMI che elimini, puoi scegliere di rimuovere anche le istantanee distribuite AMIs e le relative istantanee associate. Questo approccio consente di risparmiare sui costi di archiviazione delle istantanee.
- Per le risorse di immagine del contenitore che elimini, puoi scegliere di eliminare le risorse sottostanti. Questo approccio può far risparmiare sui costi di storage e sulle velocità di trasferimento dei dati di Amazon ECR per le Docker immagini archiviate negli archivi ECR.

> **Note**
>
> Image Builder non è in grado di valutare il potenziale impatto di tutte le possibili dipendenze a valle, come i gruppi di Auto Scaling o i modelli di avvio. È necessario considerare le dipendenze a valle delle immagini quando si configurano le azioni politiche.

Indice

- Prerequisiti per la gestione del ciclo di vita per le immagini di Image Builder
- Elenca le politiche di gestione del ciclo di vita per le risorse di immagine Image Builder
- Visualizza i dettagli delle policy relative al ciclo di vita
- Crea politiche per il ciclo di vita
- Come funzionano le regole di gestione del ciclo di vita per le risorse di immagini Image Builder

Prerequisiti per la gestione del ciclo di vita per le immagini di Image Builder

Prima di poter definire le politiche e le regole di gestione del ciclo di vita di EC2 Image Builder per le risorse di immagini, è necessario soddisfare i seguenti prerequisiti.

- Crea un ruolo IAM che conceda l'autorizzazione a Image Builder di eseguire le policy del ciclo di vita. Per creare il ruolo, consulta Crea un ruolo IAM per la gestione del ciclo di vita di Image Builder.
- Crea un ruolo IAM nell'account di destinazione per le risorse associate che sono state distribuite tra gli account. Il ruolo concede a Image Builder l'autorizzazione a eseguire azioni del ciclo di vita nell'account di destinazione per le risorse associate. Per creare il ruolo, consulta Crea un ruolo IAM per la gestione del ciclo di vita tra account di Image Builder.

> **Note**
>
> Questo prerequisito non si applica se hai concesso le autorizzazioni di avvio per un'AMI di output. Con le autorizzazioni di avvio, l'account con cui hai condiviso possiede le istanze avviate dall'AMI condivisa, ma tutte le risorse AMI rimangono nel tuo account.

- Per le immagini dei contenitori, è necessario aggiungere il seguente tag ai repository ECR per consentire a Image Builder di eseguire azioni del ciclo di vita sulle immagini del contenitore archiviate nel repository:. `LifecycleExecutionAccess: EC2 Image Builder`

Crea un ruolo IAM per la gestione del ciclo di vita di Image Builder

Per concedere l'autorizzazione a Image Builder di eseguire le politiche del ciclo di vita, devi prima creare il ruolo IAM che utilizza per eseguire le azioni del ciclo di vita. Segui questi passaggi per creare il ruolo di servizio che concede l'autorizzazione.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Selezionare Roles (Ruoli) dal riquadro di navigazione.
3. Scegliere Crea ruolo. Si apre la prima fase del processo Seleziona un'entità affidabile per creare il tuo ruolo.
4. Seleziona l'opzione Politica di fiducia personalizzata per il tipo di entità affidabile.
5. Copia la seguente politica di fiducia JSON e incollala nell'area di testo della politica di fiducia personalizzata, sostituendo il testo di esempio. Questa politica di attendibilità consente a Image Builder di assumere il ruolo creato dall'utente per eseguire le azioni del ciclo di vita.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sts:AssumeRole"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "imagebuilder.amazonaws.com"
                ]
```

```
            }
        }
    ]
}
```

6. Seleziona la seguente politica gestita dall'elenco: EC2ImageBuilderLifecycleExecutionPolicy, quindi scegli Avanti. Verrà aperta la pagina Nome, revisione e creazione.

> **Tip**
>
> Filtra `image` per semplificare i risultati.

7. Immettere un nome di ruolo in Role name (Nome ruolo).
8. Dopo aver esaminato le impostazioni, scegli Crea ruolo.

Crea un ruolo IAM per la gestione del ciclo di vita tra account di Image Builder

Per concedere l'autorizzazione a Image Builder di eseguire azioni del ciclo di vita negli account di destinazione per le risorse associate, è necessario innanzitutto creare il ruolo IAM che utilizza per eseguire azioni del ciclo di vita in tali account. È necessario creare il ruolo nell'account di destinazione.

Segui questi passaggi per creare il ruolo di servizio che concede l'autorizzazione nell'account di destinazione.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Selezionare Roles (Ruoli) dal riquadro di navigazione.
3. Scegliere Crea ruolo. Si apre la prima fase del processo Seleziona l'entità attendibile per creare il tuo ruolo.
4. Seleziona l'opzione Politica di fiducia personalizzata per il tipo di entità affidabile.
5. Copia la seguente politica di fiducia JSON e incollala nell'area di testo della politica di fiducia personalizzata, sostituendo il testo di esempio. Questa politica di attendibilità consente a Image Builder di assumere il ruolo creato dall'utente per eseguire le azioni del ciclo di vita.

> **Note**
>
> Quando Image Builder utilizza questo ruolo nell'account di destinazione per agire sulle risorse associate distribuite tra gli account, agisce per conto del proprietario dell'account di destinazione. L' Account AWS account configurato come indicato `aws:SourceAccount` nella politica di attendibilità è l'account in cui Image Builder ha distribuito tali risorse.

```
{
    "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. Seleziona la seguente politica gestita dall'elenco: EC2ImageBuilderLifecycleExecutionPolicy, quindi scegli Avanti. Verrà aperta la pagina Nome, revisione e creazione.

> **Tip**
>
> Filtra `image` per semplificare i risultati.

7. Inserisci `Ec2ImageBuilderCrossAccountLifecycleAccess` come nome del ruolo.

> **Important**
>
> `Ec2ImageBuilderCrossAccountLifecycleAccess`deve essere il nome di questo ruolo.

8. Dopo aver esaminato le impostazioni, scegli Crea ruolo.

Elenca le politiche di gestione del ciclo di vita per le risorse di immagine Image Builder

È possibile ottenere un elenco delle politiche di gestione del ciclo di vita delle immagini che include le colonne dei dettagli chiave nella pagina di elenco dei criteri del ciclo di vita in AWS Management Console, o con comandi o azioni nell'API Image Builder, oppure. SDKs AWS CLI

È possibile utilizzare uno dei seguenti metodi per elencare le risorse relative ai criteri del ciclo di vita delle immagini di Image Builder presenti nel proprio. Account AWS Per l'azione API, vedere ListLifecyclePoliciesnel riferimento all'API EC2 Image Builder. Per la richiesta SDK associata, fai riferimento al link Vedi anche nella stessa pagina.

AWS Management Console

I seguenti dettagli sono mostrati nella console per le politiche esistenti. Puoi selezionare qualsiasi colonna per modificare l'ordinamento dei risultati. L'elenco delle politiche è inizialmente ordinato in base al nome della politica. Il nome della colonna per l'ordinamento corrente è in grassetto.

Se hai più di una pagina di risultati, le frecce di paginazione nell'angolo superiore destro del pannello diventano attive. È possibile filtrare i risultati per nome della politica, stato della politica, tipo di immagine di output e ARN della risorsa immagine con la barra di ricerca.

- Nome della politica: il nome della politica.
- Stato della politica: se la politica è attiva o inattiva.
- Tipo: il tipo di immagine di output che Image Builder distribuisce quando si crea una nuova versione dell'immagine (AMI o immagine contenitore).
- Data dell'ultima esecuzione: l'ultima volta in cui è stata eseguita la policy del ciclo di vita.
- Data di creazione: il timestamp dalla creazione della politica del ciclo di vita.
- ARN: Amazon Resource Name (ARN) della risorsa relativa alla policy del ciclo di vita.

Per elencare le politiche relative al ciclo di vita in, segui questi passaggi: AWS Management Console

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Seleziona le politiche del ciclo di vita dal riquadro di navigazione. Questo mostra un elenco delle politiche relative al ciclo di vita delle immagini presenti nel tuo account.

Operazioni disponibili

Puoi anche eseguire le seguenti azioni per la tua politica del ciclo di vita dalla pagina di elenco delle politiche del ciclo di vita.

Per creare una nuova policy sul ciclo di vita dell'immagine, scegli Crea policy per il ciclo di vita. Per ulteriori informazioni su come creare una policy, consulta. Crea politiche per il ciclo di vita

Per tutte le azioni seguenti, è necessario prima selezionare la politica. Per selezionare una politica, è possibile selezionare la casella di controllo accanto al nome della politica.

- Per attivare o disattivare la policy, seleziona Disabilita policy o Abilita policy dal menu Azioni.
- Per modificare la politica, seleziona Modifica politica dal menu Azioni.
- Per eliminare una politica, seleziona Elimina politica dal menu Azioni.
- Per creare una nuova politica che utilizzi la politica selezionata per le impostazioni di base, seleziona Clona policy dal menu Azioni.

AWS CLI

L'esempio di comando seguente mostra come utilizzare la per elencare le politiche relative AWS CLI al ciclo di vita delle immagini per uno specifico. Regione AWS Per ulteriori informazioni sui parametri e le opzioni che è possibile utilizzare con questo comando, vedere il list-lifecycle-policiescomando nella Guida ai AWS CLI comandi.

Esempio:

```
aws imagebuilder list-lifecycle-policies \
--region us-west-1
```

Output:

```
{
    "lifecyclePolicySummaryList": [
        {
            "arn": "arn:aws:imagebuilder:us-west-2:111122223333:lifecycle-policy/
sample-lifecycle-policy1",
            "name": "sample-lifecycle-policy1",
            "status": "DISABLED",
            "executionRole": "arn:aws:iam::111122223333:role/sample-lifecycle-role",
            "resourceType": "AMI_IMAGE",
            "dateCreated": "2023-11-07T14:57:01.603000-08:00",
            "tags": {}
        },
        {
            "arn": "arn:aws:imagebuilder:us-west-2:111122223333:lifecycle-policy/
sample-lifecycle-policy2",
            "name": "sample-lifecycle-policy2",
            "status": "ENABLED",
            "executionRole": "arn:aws:iam::111122223333:role/sample-lifecycle-role",
            "resourceType": "AMI_IMAGE",
            "dateCreated": "2023-09-06T10:43:21.436000-07:00",
            "dateLastRun": "2023-11-13T04:43:46.106000-08:00",
            "tags": {}
        },
        {
            "arn": "arn:aws:imagebuilder:us-west-2:111122223333:lifecycle-policy/
sample-lifecycle-policy3",
            "name": "sample-lifecycle-policy3",
            "status": "ENABLED",
            "executionRole": "arn:aws:iam::111122223333:role/sample-lifecycle-role",
            "resourceType": "AMI_IMAGE",
            "dateCreated": "2023-10-19T15:16:40.046000-07:00",
            "dateUpdated": "2023-10-21T20:07:15.958000-07:00",
            "dateLastRun": "2023-11-12T09:27:45.830000-08:00"
}]}
```

> **Note**
>
> Per utilizzare l'impostazione predefinita Regione AWS, esegui questo comando senza il `--region` parametro.

Visualizza i dettagli delle policy relative al ciclo di vita

La pagina di dettaglio dei criteri del ciclo di vita nella console Image Builder include una sezione di riepilogo, con informazioni aggiuntive raggruppate in schede. L'intestazione della pagina è il nome della politica.

Nella pagina dei dettagli della politica del ciclo di vita nella console Image Builder, è possibile visualizzare i dettagli di una politica del ciclo di vita specifica. Puoi anche utilizzare comandi o azioni con l'API Image Builder o AWS CLI per ottenere dettagli sulle politiche. SDKs

Indice

- Visualizza i dettagli delle policy relative al ciclo di vita nella console Image Builder

Visualizza i dettagli delle policy relative al ciclo di vita nella console Image Builder

La pagina dei dettagli dell'immagine nella console Image Builder include una sezione di riepilogo, con informazioni aggiuntive raggruppate in schede. L'intestazione della pagina è il nome e la versione di build della ricetta che ha creato l'immagine.

Sezioni e schede con i dettagli della console

- Sezione riepilogo
- Scheda Regole
- Scheda Ambito
- RunLog scheda

Sezione riepilogo

La sezione di riepilogo copre l'intera larghezza della pagina e include i seguenti dettagli. Questi dettagli vengono sempre visualizzati.

Stato della politica

Se la politica è attiva o inattiva.

Tipo

Il tipo di immagine di output che Image Builder distribuisce quando si crea una nuova versione dell'immagine (AMI o immagine contenitore).

Date created (Data di creazione)

Il timestamp dalla creazione della politica del ciclo di vita.

Data modificata

L'ultima volta che la politica del ciclo di vita è stata aggiornata.

Data dell'ultima esecuzione

L'ultima volta che è stata eseguita la politica del ciclo di vita.

Ruolo IAM

Il ruolo IAM utilizzato da Image Builder per eseguire azioni del ciclo di vita.

ARN

L'Amazon Resource Name (ARN) della risorsa relativa alla policy del ciclo di vita.

Descrizione

La descrizione della politica del ciclo di vita, se inserita.

Scheda Regole

La scheda Regole mostra le regole del ciclo di vita configurate per la politica che stai visualizzando. La scheda include i seguenti dettagli:

- Nome: il nome della regola. Questi nomi sono statici e si basano sulle azioni politiche che è possibile configurare.
  - `Deprecation rule`
  - `Disable rule`
  - `Deletion rule`
- Regola: una breve descrizione dell'azione configurata per la regola.
- Condizioni delle regole: elenca la configurazione per la gestione delle risorse associate, le eccezioni alla regola e le impostazioni di conservazione, se applicabili.

Per ulteriori informazioni sulla configurazione delle regole, vedereCome funzionano le regole del ciclo di vita.

Scheda Ambito

La scheda Ambito mostra i criteri di selezione delle risorse configurati per la politica che stai visualizzando. La scheda include i seguenti dettagli:

- Filtro: ***`type of filter`*** — Il tipo di filtro utilizzato per definire l'ambito. Il tipo di filtro può essere uno dei seguenti:
  - `recipes`— Le ricette utilizzate per creare le immagini a cui si applica la politica del ciclo di vita.
  - `tags`— Un set di tag che Image Builder utilizza per selezionare le risorse di immagine a cui si applica la politica del ciclo di vita.
- Una barra di ricerca: è possibile filtrare l'elenco per nome per semplificare i risultati visualizzati nella scheda.
- Nome: ogni riga contiene un nome o un tag che hai configurato per i criteri di filtro.
- Versione: se è stato configurato un filtro per le ricette, Image Builder visualizza la versione della ricetta.

RunLog scheda

Ogni volta che si esegue la policy per le risorse configurate, Image Builder salva i dettagli di runtime. Ogni riga della tabella rappresenta una singola istanza di runtime. La scheda include i seguenti dettagli:

- Execution ID: identifica l'istanza di runtime della policy del ciclo di vita.
- Stato di esecuzione: stato di runtime che segnala se l'azione della policy è attualmente in esecuzione, eseguita con successo, non riuscita o è stata annullata.
- Risorsa interessata: indica se l'istanza di runtime ha identificato risorse di immagine per le azioni del ciclo di vita.
- Data di inizio: il timestamp in cui è stata avviata l'istanza di runtime.
- Data di fine: il timestamp in cui è terminata l'istanza di runtime.

Crea politiche per il ciclo di vita

Quando si crea una nuova politica del ciclo di vita di EC2 Image Builder, la configurazione dipende dal tipo di immagine a cui è destinata la politica. L'azione dell'API per creare una politica del ciclo di vita per le risorse di immagine AMI e le risorse di immagine del contenitore è la stessa () CreateLifecyclePolicy. Tuttavia, la configurazione delle risorse di immagine e delle risorse associate è diversa. Questa sezione mostra come creare politiche di gestione del ciclo di vita per entrambi.

> **Note**
>
> Prima di creare una politica del ciclo di vita, assicurati di aver soddisfatto tutti i criteri. Prerequisiti

Crea politiche di gestione del ciclo di vita per le risorse di immagine AMI di Image Builder

È possibile utilizzare uno dei seguenti metodi per creare una politica del ciclo di vita delle immagini AMI con o. AWS Management Console AWS CLIÈ inoltre possibile utilizzare l'azione CreateLifecyclePolicyAPI. Per la richiesta SDK associata, puoi fare riferimento al link Vedi anche per quel comando nel riferimento all'API EC2 Image Builder.

AWS Management Console

Per creare una politica del ciclo di vita per le risorse di immagini AMI in AWS Management Console, procedi nel seguente modo:

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Scegli le politiche del ciclo di vita dal pannello di navigazione.
3. Scegli Crea policy sul ciclo di vita.
4. Configura le impostazioni dei criteri descritte nelle seguenti procedure.
5. Per creare la politica del ciclo di vita dopo aver configurato le impostazioni, scegli Crea politica.

Configura le impostazioni generali per la tua politica.

1. Seleziona l'opzione AMI da Tipo di policy.
2. Immettere il nome della politica.
3. Facoltativamente, inserisci una descrizione per la tua politica del ciclo di vita.
4. Per impostazione predefinita, l'opzione Activate è attivata. L'impostazione predefinita attiva la politica del ciclo di vita e la aggiunge immediatamente alla pianificazione. Per creare una policy inizialmente disattivata, puoi disattivare Activate.
5. Seleziona il ruolo IAM che hai creato per le autorizzazioni delle policy del ciclo di vita. Se non hai ancora creato questo ruolo, consulta Prerequisiti per ulteriori informazioni.

Configura l'ambito della regola per la tua politica.

Questa sezione configura la selezione delle risorse per la politica del ciclo di vita, in base al tipo di filtro utilizzato.

1. Tipo di filtro: ricette: per applicare le regole del ciclo di vita alle risorse di immagini in base alla ricetta che le ha create, seleziona fino a 50 versioni di ricette per la policy.
2. Tipo di filtro: tag: per applicare le regole del ciclo di vita alle risorse di immagini in base ai tag delle risorse, inserisci un elenco di un massimo di 50 coppie di valori chiave a cui applicare la policy.

Attiva una o più delle seguenti regole del ciclo di vita da applicare alle risorse selezionate dalla politica del ciclo di vita. Se una risorsa corrisponde a più di una regola del ciclo di vita quando viene eseguita la policy, Image Builder esegue le azioni delle regole nel seguente ordine: 1) Obsoleto, 2) Disabilita, 3) Elimina.

Regola obsoleta

Imposta lo stato della risorsa immagine Image Builder su. `Deprecated` Le pipeline di Image Builder continuano a funzionare per immagini obsolete. Facoltativamente, è possibile impostare il periodo di obsolescenza di Associated AMIs senza compromettere la possibilità di avviare nuove istanze.

- Numero di unità: specifica il valore intero per il periodo di tempo che deve trascorrere dopo la creazione di una risorsa immagine prima che venga contrassegnata come. `Deprecated`
- Unità: seleziona l'intervallo di tempo da utilizzare. L'intervallo può essere `DaysWeeks`,`Months`, o`Years`.

- Obsoleto AMIs: seleziona la casella di controllo per contrassegnare l'Amazon associato EC2 AMIs con una data di obsolescenza. AMIs Rimangono disponibili e puoi comunque avviare nuove istanze da esse.

Disabilita la regola

Imposta lo stato della risorsa immagine Image Builder su. `Disabled` Ciò impedisce l'esecuzione delle pipeline di Image Builder per questa immagine. Facoltativamente, puoi disabilitare l'AMI associata per impedire l'avvio di nuove istanze.

- Numero di unità: specifica il valore intero per il periodo di tempo che deve trascorrere dopo la creazione di una risorsa di immagine prima che venga contrassegnata come. `Disabled`
- Unità: seleziona l'intervallo di tempo da utilizzare. L'intervallo può essere `DaysWeeks`,`Months`, `oYears`.
- Disabilita AMIs: seleziona la casella di controllo per disabilitare Amazon EC2 AMIs associato. Non puoi più utilizzare AMIs o avviare nuove istanze da essi.

Elimina regola

Elimina le risorse di immagine per età o per numero. Siete voi a definire la soglia che soddisfa le vostre esigenze. Quando una risorsa immagine Image Builder supera la soglia, viene rimossa. Facoltativamente, è possibile annullare la registrazione associata AMIs o eliminare le relative istantanee. AMIs È inoltre possibile specificare i tag per le risorse che si desidera conservare oltre la soglia.

Quando si configura la regola di eliminazione per età, Image Builder elimina la risorsa immagine dopo un periodo di tempo configurato. Ad esempio, elimina le risorse di immagine dopo 6 mesi. Quando si esegue la configurazione in base al numero, Image Builder conserva il numero di immagini specificato più recente o il più vicino possibile a tale numero ed elimina le versioni precedenti.

- Per età
  - Numero di unità: specifica il valore intero per il periodo di tempo che deve trascorrere dopo la creazione di una risorsa di immagine prima che venga eliminata.
  - Unità: seleziona l'intervallo di tempo da utilizzare. L'intervallo può essere `DaysWeeks`,`Months`, `oYears`.

- Conserva almeno un'immagine per ricetta: seleziona la casella di controllo per conservare l'ultima risorsa di immagini disponibile per ogni versione di ricetta interessata da questa regola.

Per conteggio

- Numero di immagini: specifica il valore intero per il numero di risorse di immagini recenti da conservare per ogni versione della ricetta.

- Annulla registrazione AMIs: seleziona la casella di controllo per annullare la registrazione di Amazon associata. EC2 AMIs Non puoi più utilizzare AMIs o avviare nuove istanze da essi.
- Conserva immagini e istantanee con tag associati: seleziona la casella di controllo per inserire un elenco di tag per le risorse di immagini che desideri conservare. AMIs I tag si applicano alle risorse di immagini e ad Amazon EC2 AMIs. Puoi inserire fino a 50 coppie chiave-valore.

Tag (opzionale)

Aggiungi tag alla tua politica sul ciclo di vita.

AWS CLI

Per creare una nuova policy del ciclo di vita di Image Builder, è possibile utilizzare il create-lifecycle-policy comando in. AWS CLI

Crea politiche di gestione del ciclo di vita per le risorse di immagini dei contenitori Image Builder

È possibile utilizzare uno dei seguenti metodi per creare una politica del ciclo di vita dell'immagine del contenitore con o. AWS Management Console AWS CLI Puoi anche utilizzare l'azione CreateLifecyclePolicyAPI. Per la richiesta SDK associata, puoi fare riferimento al link Vedi anche per quel comando nel riferimento all'API EC2 Image Builder.

AWS Management Console

Per creare una politica del ciclo di vita per le risorse di immagine del contenitore in AWS Management Console, procedi nel seguente modo:

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Scegli le politiche del ciclo di vita dal pannello di navigazione.

3. Scegli Crea policy sul ciclo di vita.
4. Configura le impostazioni dei criteri descritte nelle seguenti procedure.
5. Per creare la politica del ciclo di vita dopo aver configurato le impostazioni, scegli Crea politica.

Configurazione della politica: impostazioni generali

Configura le impostazioni generali per la tua politica.

1. Seleziona l'opzione AMI da Tipo di policy.
2. Immettere il nome della politica.
3. Facoltativamente, inserisci una descrizione per la tua politica del ciclo di vita.
4. Per impostazione predefinita, l'opzione Activate è attivata. L'impostazione predefinita attiva la politica del ciclo di vita e la aggiunge immediatamente alla pianificazione. Per creare una policy inizialmente disattivata, puoi disattivare Activate.
5. Seleziona il ruolo IAM che hai creato per le autorizzazioni delle policy del ciclo di vita. Se non hai ancora creato questo ruolo, consulta Prerequisiti per ulteriori informazioni.

Configura l'ambito della regola per la tua politica.

Questa sezione configura la selezione delle risorse per la politica del ciclo di vita, in base al tipo di filtro utilizzato.

1. Tipo di filtro: ricette: per applicare le regole del ciclo di vita alle risorse di immagini in base alla ricetta che le ha create, seleziona fino a 50 versioni di ricette per la policy.
2. Tipo di filtro: tag: per applicare le regole del ciclo di vita alle risorse di immagini in base ai tag delle risorse, inserisci un elenco di un massimo di 50 coppie di valori chiave a cui applicare la policy.

Eliminare la regola

Per le immagini del contenitore, questa regola elimina la risorsa immagine del contenitore Image Builder. Facoltativamente, puoi rimuovere le immagini Docker che sono state distribuite negli archivi ECR per impedire che vengano utilizzate per eseguire nuovi contenitori.

Quando si configura la regola di eliminazione per età, Image Builder elimina la risorsa immagine dopo un periodo di tempo configurato. Ad esempio, elimina le risorse di immagine dopo 6 mesi.

Quando si esegue la configurazione in base al numero, Image Builder conserva il numero di immagini specificato più recente o il più vicino possibile a tale numero ed elimina le versioni precedenti.

- Per età
  - Numero di unità: specifica il valore intero per il periodo di tempo che deve trascorrere dopo la creazione di una risorsa di immagine prima che venga eliminata.
  - Unità: seleziona l'intervallo di tempo da utilizzare. L'intervallo può essere `DaysWeeks`,`Months`, `oYears`.
  - Conserva almeno un'immagine: seleziona la casella di controllo per conservare solo l'ultima risorsa di immagini disponibile per ogni versione di ricetta interessata da questa regola.

  Per conteggio
  - Numero di immagini: specifica il valore intero per il numero di risorse di immagini recenti da conservare per ogni versione della ricetta.
- Elimina le immagini del contenitore ECR: seleziona la casella di controllo per eliminare le immagini del contenitore associate archiviate in un repository ECR. Non è più possibile utilizzare l'immagine del contenitore come base per creare nuove immagini o per eseguire nuovi contenitori.
- Conserva le immagini con i tag associati: seleziona la casella di controllo per inserire un elenco di tag per le risorse di immagini che desideri conservare.

Tag (opzionale)

Aggiungi tag alla tua politica sul ciclo di vita.

AWS CLI

Per creare una nuova policy del ciclo di vita di Image Builder, è possibile utilizzare il create-lifecycle-policy comando in. AWS CLI

Come funzionano le regole di gestione del ciclo di vita per le risorse di immagini Image Builder

Le policy relative al ciclo di vita delle immagini utilizzano le regole del ciclo di vita definite dall'utente per implementare la strategia generale di gestione delle risorse. Le regole che definisci aiutano a

garantire l'aggiornamento delle immagini disponibili e a ridurre al minimo i costi per l'infrastruttura sottostante, come lo storage delle istantanee per l'output o lo storage nel repository ECR e le AMIs velocità di trasferimento dei dati per le immagini dei container.

È possibile configurare i seguenti tipi di regole per le politiche.

Regola obsoleta

Imposta lo stato della risorsa immagine Image Builder su. `Deprecated` Le pipeline di Image Builder continuano a funzionare per immagini obsolete. Facoltativamente, è possibile impostare il periodo di obsolescenza di Associated AMIs senza compromettere la possibilità di avviare nuove istanze.

Quando un AMI è obsoleto, viene ignorato dalle ricerche generiche. Ad esempio, se esegui il EC2 describe-images comando Amazon in AWS CLI, non risulterà deprecato AMIs nel set di risultati. Tuttavia, puoi comunque trovarli obsoleti con il AMIs loro ID AMI.

Questa regola non è disponibile per le immagini dei contenitori.

Disabilita la regola

Imposta lo stato della risorsa immagine Image Builder su. `Disabled` Ciò impedisce l'esecuzione delle pipeline di Image Builder per questa immagine. Facoltativamente, puoi disabilitare l'AMI associata per impedire l'avvio di nuove istanze.

Quando un'AMI è disabilitata, diventa privata e non può essere utilizzata per avviare nuove istanze. Se hai condiviso l'AMI con account, organizzazioni o unità organizzative, questi perderanno l'accesso all'AMI quando questa diventa privata.

Questa regola non è disponibile per le immagini dei contenitori.

Regola di eliminazione

Elimina le risorse di immagine in base all'età o al numero. Siete voi a definire la soglia che soddisfa le vostre esigenze. Quando una risorsa immagine Image Builder supera la soglia, viene rimossa. Facoltativamente, è possibile annullare la registrazione associata AMIs o eliminare le relative istantanee. AMIs È inoltre possibile specificare i tag per le risorse che si desidera conservare oltre la soglia.

Per le immagini del contenitore, questa regola elimina la risorsa immagine del contenitore Image Builder. Facoltativamente, puoi rimuovere le immagini dei contenitori che sono state distribuite negli archivi ECR per impedire che vengano utilizzate per eseguire nuovi contenitori.

Indice

- Regole di esclusione del ciclo di vita delle AMI
- Visualizza i dettagli delle regole di gestione del ciclo di vita per una policy

Regole di esclusione del ciclo di vita delle AMI

Le seguenti regole di esclusione definiscono le eccezioni alle regole del ciclo di vita per. AMIs AMIs che soddisfano i criteri specificati dalle regole di esclusione sono escluse dalle azioni del ciclo di vita. Le regole di esclusione non sono disponibili in. AWS Management Console

I termini seguenti utilizzano la notazione API del tipo di `LifecyclePolicyDetailExclusionRules` dati.

Regole di esclusione

obiettivi

Contiene le impostazioni `LifecyclePolicyDetailExclusionRulesAmis` mostrate nell'elenco che segue.

TagMap

Puoi fornire un elenco di un massimo di 50 tag che ignorano le azioni del ciclo di vita per qualsiasi tipo di risorsa.

I termini seguenti utilizzano la notazione API del tipo di dati. `LifecyclePolicyDetailExclusionRulesAmis`

Regole di esclusione AMI

isPublic

Configura se il pubblico AMIs è escluso dall'azione del ciclo di vita.

Lanciato per ultimo

Specifica i dettagli di configurazione per Image Builder per escludere le risorse più recenti dalle azioni del ciclo di vita.

regioni

Configurazioni Regioni AWS escluse dall'azione del ciclo di vita.

Account condivisi

Speciifica le risorse Account AWS escluse dall'azione del ciclo di vita.

TagMap

Elenca i tag che devono essere esclusi dalle azioni del ciclo di vita per coloro AMIs che li hanno.

Visualizza i dettagli delle regole di gestione del ciclo di vita per una policy

Le regole sono definite all'interno delle politiche di gestione del ciclo di vita create per le risorse di immagini Image Builder. Nella console, la pagina dei dettagli della politica del ciclo di vita contiene una pagina Scheda Regole che mostra i dettagli delle regole configurate per la politica.

Per ottenere i dettagli della politica in AWS CLI, è possibile eseguire il get-lifecycle-policycomando. I dettagli della politica nella risposta contengono un elenco delle azioni (regole) definite per la politica, che includono tutte le impostazioni configurate.

Configurazione di immagini personalizzate con Image Builder

Le risorse di configurazione sono gli elementi costitutivi che costituiscono le pipeline di immagini, nonché le immagini prodotte da tali pipeline. Questo capitolo tratta la creazione, la manutenzione e la condivisione delle risorse di Image Builder, inclusi componenti, ricette e immagini, insieme alla configurazione dell'infrastruttura e alle impostazioni di distribuzione.

> **Note**
>
> Per aiutarti a gestire le risorse di Image Builder, puoi assegnare i tuoi metadati a ciascuna risorsa sotto forma di tag. I tag vengono utilizzati per classificare le AWS risorse in diversi modi, ad esempio per scopo, proprietario o ambiente. Questa funzione è utile quando si dispone di numerose risorse dello stesso tipo. Puoi identificare più facilmente una risorsa specifica in base ai tag che le hai assegnato.
> Per ulteriori informazioni sull'etichettatura delle risorse utilizzando i comandi Image Builder in, AWS CLI consultate Aggiunta di tag alle risorse la sezione di questa guida.

Indice

- Gestire le ricette in Image Builder
- Gestione della configurazione dell'infrastruttura Image Builder
- Gestione delle impostazioni di distribuzione di Image Builder

Gestire le ricette in Image Builder

Una ricetta di EC2 Image Builder definisce l'immagine di base da utilizzare come punto di partenza per creare una nuova immagine, insieme al set di componenti da aggiungere per personalizzare l'immagine e verificare che tutto funzioni come previsto. Image Builder offre scelte di versione automatiche per ogni componente. Per impostazione predefinita, è possibile applicare fino a 20 componenti a una ricetta. Ciò include sia i componenti di compilazione che quelli di test.

Dopo aver creato una ricetta, non puoi modificarla o sostituirla. Per aggiornare i componenti dopo aver creato una ricetta, è necessario creare una nuova ricetta o una nuova versione della ricetta. Puoi

sempre applicare tag alle tue ricette esistenti. Per ulteriori informazioni sull'etichettatura delle risorse utilizzando i comandi Image Builder in, AWS CLI consultate Aggiunta di tag alle risorse la sezione di questa guida.

> **Tip**
>
> Puoi utilizzare i componenti gestiti da Amazon nelle tue ricette oppure puoi sviluppare componenti personalizzati. Per ulteriori informazioni, consulta Sviluppa componenti personalizzati per la tua immagine Image Builder. Per le ricette di immagini che creano output AMIs, puoi anche utilizzare prodotti e componenti di Marketplace AWS immagini. Per ulteriori informazioni sull'integrazione con Marketplace AWS i prodotti, consultaMarketplace AWS integrazione in Image Builder.

Questa sezione spiega come elencare, visualizzare e creare ricette.

Indice

- Elenca e visualizza i dettagli della ricetta tramite immagine
- Elenca e visualizza i dettagli della ricetta del contenitore
- Crea una nuova versione di una ricetta di immagini
- Crea una nuova versione di una ricetta contenitore
- Pulizia delle risorse

Elenca e visualizza i dettagli della ricetta tramite immagine

Questa sezione descrive i vari modi in cui è possibile trovare informazioni e visualizzare i dettagli delle ricette di EC2 immagini di Image Builder.

Dettagli della ricetta basata sull'immagine

- Elenca le ricette di immagini dalla console
- Elenca le ricette con immagini tratte da AWS CLI
- Visualizza le immagini e i dettagli della ricetta dalla console.
- Ottieni i dettagli dell'immagine della ricetta dal AWS CLI
- Ottieni i dettagli della politica relativa alle ricette relative alle immagini dal AWS CLI

Elenca le ricette di immagini dalla console

Per visualizzare un elenco delle ricette di immagini create con il tuo account nella console Image Builder, procedi nel seguente modo:

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Scegli Ricette di immagini dal pannello di navigazione. Questo mostra un elenco delle ricette di immagini create con il tuo account.
3. Per visualizzare i dettagli o creare una nuova versione della ricetta, scegli il link Nome ricetta. Verrà aperta la visualizzazione dettagliata della ricetta.

> **Note**
>
> Puoi anche selezionare la casella di controllo accanto al nome della ricetta, quindi scegliere Visualizza dettagli.

Elenca le ricette con immagini tratte da AWS CLI

L'esempio seguente mostra come elencare tutte le ricette di immagini, utilizzando AWS CLI.

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

Visualizza le immagini e i dettagli della ricetta dalla console.

Per visualizzare i dettagli di una ricetta di immagini specifica utilizzando la console Image Builder, selezionare la ricetta dell'immagine da esaminare, seguendo i passaggi descritti in. Elenca le ricette di immagini dalla console

Nella pagina dei dettagli della ricetta, puoi:

- Eliminare la ricetta. Per ulteriori informazioni sull'eliminazione di risorse in Image Builder, vedere. Eliminare risorse Image Builder obsolete o inutilizzate
- Creare una nuova versione.
- Crea una pipeline dalla ricetta. Dopo aver scelto Crea pipeline da questa ricetta, verrai indirizzato alla procedura guidata per la pipeline. Per ulteriori informazioni sulla creazione di una pipeline Image Builder utilizzando la procedura guidata per la pipeline, vedere Tutorial: crea una pipeline di immagini con l'AMI di output dalla procedura guidata della console Image Builder

> **Note**
>
> Quando si crea una pipeline da una ricetta esistente, l'opzione per creare una nuova ricetta non è disponibile.

Ottieni i dettagli dell'immagine della ricetta dal AWS CLI

L'esempio seguente mostra come utilizzare un comando imagebuilder CLI per ottenere i dettagli di una ricetta di immagine specificandone l'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
```

Ottieni i dettagli della politica relativa alle ricette relative alle immagini dal AWS CLI

L'esempio seguente mostra come utilizzare un comando imagebuilder CLI per ottenere i dettagli di una policy di image recipe specificandone l'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
```

Elenca e visualizza i dettagli della ricetta del contenitore

Questa sezione descrive i modi in cui è possibile trovare informazioni e visualizzare i dettagli delle ricette dei contenitori EC2 Image Builder.

Dettagli della ricetta del contenitore

- Elenca le ricette dei contenitori nella console
- Elenca le ricette in contenitore con il AWS CLI
- Visualizza i dettagli della ricetta del contenitore nella console
- Ottieni i dettagli della ricetta del contenitore con AWS CLI
- Ottieni i dettagli della politica sulla ricetta del contenitore con AWS CLI

Elenca le ricette dei contenitori nella console

Per visualizzare un elenco delle ricette di contenitori che sono state create con il tuo account nella console Image Builder, procedi nel seguente modo:

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Scegli Container recipes dal pannello di navigazione. Questo mostra un elenco delle ricette in contenitore create con il tuo account.
3. Per visualizzare i dettagli o creare una nuova versione della ricetta, scegli il link Nome ricetta. Verrà aperta la visualizzazione dettagliata della ricetta.

> **Note**
>
> Puoi anche selezionare la casella di controllo accanto al nome della ricetta, quindi scegliere Visualizza dettagli.

Elenca le ricette in contenitore con il AWS CLI

L'esempio seguente mostra come elencare tutte le ricette del contenitore, utilizzando il AWS CLI.

```
aws imagebuilder list-container-recipes
```

Visualizza i dettagli della ricetta del contenitore nella console

Per visualizzare i dettagli di una ricetta contenitore specifica con la console Image Builder, selezionare la ricetta del contenitore da esaminare e utilizzare i passaggi descritti in. Elenca le ricette dei contenitori nella console

Nella pagina dei dettagli della ricetta, puoi fare quanto segue:

- Eliminare la ricetta. Per ulteriori informazioni su come eliminare risorse in Image Builder, vedere. Eliminare risorse Image Builder obsolete o inutilizzate
- Creare una nuova versione.
- Crea una pipeline dalla ricetta. Dopo aver scelto Crea pipeline da questa ricetta, si passa alla procedura guidata per la pipeline. Per ulteriori informazioni su come creare una pipeline Image Builder utilizzando la procedura guidata per la pipeline, vedere Tutorial: crea una pipeline di immagini con l'AMI di output dalla procedura guidata della console Image Builder

> **Note**
>
> Quando si crea una pipeline da una ricetta esistente, l'opzione per creare una nuova ricetta non è disponibile.

Ottieni i dettagli della ricetta del contenitore con AWS CLI

L'esempio seguente mostra come utilizzare un comando imagebuilder CLI per ottenere i dettagli di una ricetta contenitore specificandone l'ARN.

```
aws imagebuilder get-container-recipe --container-recipe-arn arn:aws:imagebuilder:us-
west-2:123456789012:container-recipe/my-example-recipe/2020.12.03
```

Ottieni i dettagli della politica sulla ricetta del contenitore con AWS CLI

L'esempio seguente mostra come utilizzare un comando imagebuilder CLI per ottenere i dettagli di una politica di ricetta del contenitore specificandone l'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
```

Crea una nuova versione di una ricetta di immagini

Questa sezione descrive come creare una nuova versione di una ricetta di immagini.

Indice

- Crea una nuova versione di image recipe dalla console
- Crea una ricetta di immagini con AWS CLI
- Importa una macchina virtuale come immagine di base nella console

Crea una nuova versione di image recipe dalla console

Quando crei una nuova versione di ricetta, è praticamente come creare una nuova ricetta. La differenza è che alcuni dettagli sono preselezionati in base alla ricetta di base, nella maggior parte dei

casi. L'elenco seguente descrive le differenze tra la creazione di una nuova ricetta e la creazione di una nuova versione di una ricetta esistente.

Dettagli della ricetta di base nella nuova versione

- Nome: non modificabile.
- Versione: richiesta. Questo dettaglio di base non è precompilato con la versione corrente o con alcun tipo di sequenza. <major>Immettete il numero di versione che desiderate creare nel formato. <minor>. <patch>. Se la versione esiste già, si verifica un errore.
- L'opzione Seleziona immagine: preselezionata, ma è possibile modificarla. Se modificate la scelta della fonte dell'immagine di base, potreste perdere altri dettagli che dipendono dall'opzione originale scelta.

  Per visualizzare i dettagli associati alla selezione dell'immagine di base, scegliete la scheda corrispondente alla selezione.

  Managed image

  - Image Operating System (OS): non modificabile.
  - Nome dell'immagine: preselezionato, in base alla combinazione di immagini di base scelte per la ricetta esistente. Tuttavia, se si modifica l'opzione Seleziona immagine, si perde il nome dell'immagine preselezionato.
  - Opzioni di creazione automatica delle versioni: non corrisponde alla ricetta base. L'impostazione predefinita di questa opzione di immagine è l'opzione Usa la versione del sistema operativo selezionata.

  > **Important**
  >
  > Se utilizzi il controllo delle versioni semantiche per avviare le build della pipeline, assicurati di modificare questo valore in Usa l'ultima versione del sistema operativo disponibile. Per ulteriori informazioni sul controllo delle versioni semantiche per le risorse di Image Builder, vedere. Versionamento semantico in Image Builder

  Marketplace AWS image

  - Abbonamenti: questa scheda deve essere aperta e l'immagine da cui è stata effettuata la sottoscrizione Marketplace AWS deve essere preselezionata in modo che corrisponda alla ricetta di base. Se modifichi l'immagine utilizzata dalla ricetta come immagine di base, potresti perdere altri dettagli che dipendono dall'immagine originale che hai scelto.

Per ulteriori informazioni sui Marketplace AWS prodotti, consulta la sezione Acquisto di prodotti nella Guida Marketplace AWS all'acquisto.

Custom AMI

Sorgente AMI (obbligatorio): immettere l'ID AMI o un parametro AWS Systems Manager (SSM) Parameter Store che contiene un ID AMI da utilizzare come immagine di base. L'agente SSM deve essere preinstallato nell'AMI selezionata.

- ID AMI: questa impostazione non è precompilata con l'immissione originale. Inserisci l'ID AMI per l'immagine di base. Esempio: *ami-1234567890abcdef1*.
- Parametro SSM: inserisci il nome o l'ARN del parametro SSM Parameter Store che contiene l'ID AMI per l'immagine di base. Esempio: */ib/test/param* o `arn:aws:ssm:`*us-east-1*`:`*111122223333*`:parameter`*/ib/test/param*.

- Configurazione dell'istanza: le impostazioni sono preselezionate, ma è possibile modificarle.
  - Agente Systems Manager: è possibile selezionare o deselezionare questa casella di controllo per controllare l'installazione dell'agente Systems Manager sulla nuova immagine. Per impostazione predefinita, la casella di controllo è deselezionata per includere l'agente Systems Manager nella nuova immagine. Per rimuovere l'agente Systems Manager dall'immagine finale, seleziona la casella di controllo in modo che l'agente non sia incluso nell'AMI.
  - Dati utente: è possibile utilizzare quest'area per fornire comandi o uno script di comando da eseguire all'avvio dell'istanza di build. Tuttavia, questo valore sostituisce tutti i comandi che Image Builder potrebbe aver aggiunto per garantire l'installazione di Systems Manager. Questi comandi includono lo script di pulizia che Image Builder normalmente esegue per le immagini Linux prima di creare la nuova immagine.

    Quando Image Builder avvia un'istanza, gli script dei dati utente vengono eseguiti durante la fase di inizializzazione del cloud, prima che inizi l'esecuzione dei componenti. Questo passaggio viene registrato nel seguente file sull'istanza:. `var/log/cloud-init.log`

    > **Note**
    >
    > - Se inserisci dati utente, assicurati che l'agente Systems Manager sia preinstallato sull'immagine di base o che includa l'installazione nei dati utente.
    > - Per le immagini Linux, assicuratevi che la procedura di pulizia venga eseguita includendo un comando per creare un file vuoto denominato `perform_cleanup` nello script dei dati utente. Image Builder rileva questo file ed esegue lo script di pulizia

prima di creare la nuova immagine. Per ulteriori informazioni e uno script di esempio, vedere. Best practice di sicurezza per Image Builder

- Cartella di lavoro: preselezionata, ma è possibile modificarla.
- Componenti: i componenti già inclusi nella ricetta vengono visualizzati nella sezione Componenti selezionati alla fine di ciascuno degli elenchi di componenti (build e test). È possibile rimuovere o riordinare i componenti selezionati in base alle proprie esigenze.

  I componenti di protezione avanzata CIS non seguono le regole standard di ordinamento dei componenti nelle ricette di Image Builder. I componenti di protezione avanzata CIS funzionano sempre per ultimi per garantire che i test di benchmark vengano eseguiti rispetto all'immagine di output.

  Note

  Gli elenchi dei componenti di compilazione e test mostrano i componenti disponibili in base al tipo di proprietario del componente. Per aggiungere un componente, scegli Aggiungi componenti di compilazione e seleziona il filtro di proprietà applicabile. Ad esempio, per aggiungere un componente di compilazione associato a un Marketplace AWS prodotto, seleziona`Marketplace AWS`. Si apre un pannello di selezione sul lato destro dell'interfaccia della console che elenca Marketplace AWS i componenti.
  Per il componente CIS, selezionate`Third party managed`.

  È possibile configurare le seguenti impostazioni per il componente selezionato:

  - Opzioni di controllo delle versioni: preselezionate, ma è possibile modificarle. Ti consigliamo di scegliere l'opzione Usa l'ultima versione disponibile del componente per assicurarti che le tue build di immagini utilizzino sempre la versione più recente del componente. Se devi utilizzare una versione specifica del componente nella tua ricetta, puoi scegliere Specificare la versione del componente e inserire la versione nella casella Versione del componente che appare.
  - Parametri di input: visualizza i parametri di input accettati dal componente. Il valore è precompilato con il valore della versione precedente della ricetta. Se si utilizza questo componente per la prima volta in questa ricetta ed è stato definito un valore predefinito per il parametro di input, il valore predefinito viene visualizzato nella casella Valore con testo in grigio. Se non viene immesso nessun altro valore, Image Builder utilizza il valore predefinito.

Se un parametro di input è obbligatorio, ma non ha un valore predefinito definito nel componente, è necessario fornire un valore. Image Builder non creerà la versione della ricetta se mancano dei parametri obbligatori e non è stato definito un valore predefinito.

> **Important**
>
> I parametri dei componenti sono valori di testo semplice e vengono registrati. AWS CloudTrail Ti consigliamo di utilizzare AWS Secrets Manager o AWS Systems Manager Parameter Store per archiviare i tuoi segreti. Per ulteriori informazioni su Secrets Manager, vedi Cos'è Secrets Manager? nella Guida AWS Secrets Manager per l'utente. Per ulteriori informazioni su AWS Systems Manager Parameter Store, vedere AWS Systems Manager Parameter Store nella Guida AWS Systems Manager per l'utente.

Per espandere le impostazioni per le opzioni di controllo delle versioni o i parametri di input, puoi scegliere la freccia accanto al nome dell'impostazione. Per espandere tutte le impostazioni per tutti i componenti selezionati, è possibile disattivare e attivare l'opzione Espandi tutto.

- Archiviazione (volumi): sono precompilate. Le selezioni Device Name, Snapshot e IOPS del volume principale non sono modificabili. Tuttavia, è possibile modificare tutte le impostazioni rimanenti, ad esempio la dimensione. È inoltre possibile aggiungere nuovi volumi e crittografare volumi nuovi o esistenti.

  Per crittografare i volumi per le immagini che Image Builder crea con il proprio account nella regione di origine (dove viene eseguita la build), è necessario utilizzare la crittografia del volume di archiviazione nella ricetta dell'immagine. La crittografia eseguita durante la fase di distribuzione della build riguarda solo le immagini distribuite ad altri account o regioni.

  > **Note**
  >
  > Se si utilizza la crittografia per i volumi, è necessario selezionare la chiave per ogni volume separatamente, anche se la chiave è la stessa utilizzata per il volume principale.

Per creare una nuova versione della ricetta di immagini:

1. Nella parte superiore della pagina dei dettagli della ricetta, scegli Crea nuova versione. Verrai indirizzato alla pagina Crea ricetta con immagine.

2. Per creare la nuova versione, apporta le modifiche, quindi scegli Crea ricetta.

   L'immagine finale può contenere fino a quattro codici prodotto tratti da Marketplace AWS immagini di prodotti e componenti. Se l'immagine di base e i componenti selezionati contengono più di quattro codici prodotto, Image Builder restituisce un errore quando si tenta di creare la ricetta.

Per ulteriori informazioni su come creare una ricetta di immagini quando si crea una pipeline di immagini, consulta la sezione Fase 2: Scegli la ricetta Guida introduttiva di questa guida.

Crea una ricetta di immagini con AWS CLI

Per creare una ricetta di immagini con il `create-image-recipe` comando Image Builder in AWS CLI, procedi nel seguente modo:

Prerequisiti

Prima di eseguire i comandi Image Builder in questa sezione per creare una ricetta di immagini da AWS CLI, è necessario creare i componenti utilizzati dalla ricetta. L'esempio di image recipe riportato nel passaggio seguente si riferisce ai componenti di esempio creati nella Creare un componente personalizzato dal AWS CLI sezione di questa guida.

Dopo aver creato i componenti, o se state utilizzando componenti esistenti, prendete nota di quelli ARNs che desiderate includere nella ricetta.

1. Creazione di un file JSON di input nella CLI

   È possibile fornire tutto l'input per il create-image-recipe comando con parametri di comando in linea. Tuttavia, il comando risultante può essere piuttosto lungo. Per semplificare il comando, puoi invece fornire un file JSON che contenga tutte le impostazioni della ricetta.

   > **Note**
   >
   > La convenzione di denominazione per i valori dei dati nel file JSON segue lo schema specificato per i parametri di richiesta operativa dell'API Image Builder. Per esaminare i parametri di richiesta dell'operazione API, vedere il CreateImageRecipecomando nell'EC2 Image Builder API Reference.
   > Per fornire i valori dei dati come parametri della riga di comando, fate riferimento ai nomi dei parametri specificati nel AWS CLI Command Reference.

Di seguito è riportato un riepilogo dei parametri specificati da questi esempi:

- name (string, required) — Il nome della ricetta dell'immagine.
- description (string) — La descrizione della ricetta dell'immagine.
- parentImage (string, required) — L'immagine che la ricetta dell'immagine utilizza come base per l'immagine personalizzata. È possibile specificare l'immagine principale utilizzando una delle seguenti opzioni:
  - ID AMI
  - ARN della risorsa di immagini Image Builder
  - AWS Systems Manager (SSM) Parametro Store, con il prefisso e seguito dal `ssm:` nome del parametro o dall'ARN.
  - Marketplace AWS ID del prodotto

  > **Note**
  >
  > Gli esempi Linux e macOS utilizzano un'AMI Image Builder e l'esempio Windows utilizza un ARN.

- SemanticVersion <major>(string, required) — La versione semantica della ricetta dell'immagine, espressa nel seguente formato, con valori numerici in ogni posizione per indicare una versione specifica:. <minor>. <patch>. Ad esempio, un valore potrebbe essere`1.0.0`. Per ulteriori informazioni sul controllo delle versioni semantiche per le risorse di Image Builder, vedere. Versionamento semantico in Image Builder
- components (array, obbligatorio): contiene una serie di oggetti. `ComponentConfiguration` È necessario specificare almeno un componente di build:

  > **Note**
  >
  > Image Builder installa i componenti nell'ordine specificato nella ricetta. Tuttavia, i componenti di protezione avanzata CIS funzionano sempre per ultimi per garantire che i test di benchmark vengano eseguiti rispetto all'immagine di output.

  - componentARN (stringa, obbligatorio) — L'ARN del componente.

> **Tip**
>
> Per utilizzare uno degli esempi per creare una ricetta di immagini personalizzata, è necessario sostituire l'esempio ARNs con il ARNs relativo ai componenti utilizzati per la ricetta.

- parameters (matrice di oggetti) — Contiene una serie di `ComponentParameter` oggetti. Se un parametro di input è obbligatorio, ma non ha un valore predefinito definito nel componente, è necessario fornire un valore. Image Builder non creerà la versione della ricetta se mancano dei parametri obbligatori e non è stato definito un valore predefinito.

  > **Important**
  >
  > I parametri dei componenti sono valori di testo semplice e vengono registrati. AWS CloudTrail Ti consigliamo di utilizzare AWS Secrets Manager o AWS Systems Manager Parameter Store per archiviare i tuoi segreti. Per ulteriori informazioni su Secrets Manager, vedi Cos'è Secrets Manager? nella Guida AWS Secrets Manager per l'utente. Per ulteriori informazioni su AWS Systems Manager Parameter Store, vedere AWS Systems Manager Parameter Store nella Guida AWS Systems Manager per l'utente.

  - name (string, required) — Il nome del parametro del componente da impostare.
  - value (array of string, obbligatorio) — Contiene una matrice di stringhe per impostare il valore per il parametro del componente denominato. Se è stato definito un valore predefinito per il componente e non viene fornito nessun altro valore, AWSTOE utilizza il valore predefinito.

- additionalInstanceConfiguration(oggetto): specifica impostazioni aggiuntive e avvia gli script per le tue istanze di build.
  - systemsManagerAgent(oggetto) — Contiene le impostazioni per l'agente Systems Manager sull'istanza di build.
    - uninstallAfterBuild(Boolean) — Controlla se l'agente Systems Manager viene rimosso dall'immagine di build finale prima di creare la nuova AMI. Se questa opzione è impostata `sutrue`, l'agente viene rimosso dall'immagine finale. Se l'opzione è impostata `sufalse`,

l'agente viene lasciato attivo in modo da essere incluso nella nuova AMI. Il valore predefinito è `false`.

> **Note**
>
> Se l'`uninstallAfterBuild`attributo non è incluso nel file JSON e le seguenti condizioni sono vere, Image Builder rimuove l'agente Systems Manager dall'immagine finale in modo che non sia disponibile nell'AMI:
>
> - `userDataOverride`È vuoto o è stato omesso dal file JSON.
> - Image Builder ha installato automaticamente l'agente Systems Manager sull'istanza di build per un sistema operativo che non aveva l'agente preinstallato nell'immagine di base.

- userDataOverride(stringa) — Fornisce comandi o uno script di comando da eseguire all'avvio dell'istanza di compilazione.

> **Note**
>
> I dati utente sono sempre codificati in base 64. Ad esempio, i seguenti comandi sono codificati come `IyEvYmluL2Jhc2gKbWtkaXIgLXAgL3Zhci9iYi8KdG91Y2ggL3Zhcg==`:
>
> ```
> #!/bin/bash
> mkdir -p /var/bb/
> touch /var
> ```
>
> L'esempio Linux utilizza questo valore codificato.

Linux

L'immagine di base (`parentImage`proprietà) nell'esempio seguente è un AMI. Quando si utilizza un'AMI, è necessario avere accesso all'AMI e l'AMI deve trovarsi nella regione di origine (la stessa regione in cui Image Builder esegue il comando). Salva il file con nome e utilizzalo nel create-image-recipe comando. `create-image-recipe.json`

```
{
"name": "BB Ubuntu Image recipe",
```

```
"description": "Hello World image recipe for Linux.",
"parentImage": "ami-1234567890abcdef1",
"semanticVersion": "1.0.0",
"components": [
 {
  "componentArn": "arn:aws:imagebuilder:us-west-2:111122223333:component/bb$"
 }
],
"additionalInstanceConfiguration": {
 "systemsManagerAgent": {
   "uninstallAfterBuild": true
 },
 "userDataOverride": "IyEvYmluL2Jhc2gKbWtkaXIgLXAgL3Zhci9iYi8KdG91Y2ggL3Zhcg=="
}
}
```

Windows

L'esempio seguente si riferisce alla versione più recente dell'immagine Full Base inglese di Windows Server 2016. L'ARN in questo esempio fa riferimento all'immagine più recente in base ai filtri della versione semantica che hai specificato:. `arn:aws:imagebuilder:us-west-2:aws:image/windows-server-2016-english-full-base-x86/x.x.x`

```
{
"name": "MyBasicRecipe",
"description": "This example image recipe creates a Windows 2016 image.",
"parentImage": "arn:aws:imagebuilder:us-west-2:aws:image/windows-server-2016-english-full-base-x86/x.x.x",
"semanticVersion": "1.0.0",
"components": [
 {
  "componentArn": "arn:aws:imagebuilder:us-west-2:111122223333:component/my-example-component/2019.12.02/1"
 },
 {
  "componentArn": "arn:aws:imagebuilder:us-west-2:111122223333:component/my-imported-component/1.0.0/1"
 }
]
}
```

> **Note**
>
> Per ulteriori informazioni sul controllo delle versioni semantiche per le risorse di Image Builder, vedere. Versionamento semantico in Image Builder

macOS

L'immagine di base (`parentImage`proprietà) nell'esempio seguente è un AMI. Quando si utilizza un'AMI, è necessario avere accesso all'AMI e l'AMI deve trovarsi nella regione di origine (la stessa regione in cui Image Builder esegue il comando). Salva il file con nome e utilizzalo nel create-image-recipe comando. `create-image-recipe.json`

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

2. Crea la ricetta

Usa il seguente comando per creare la ricetta. Fornisci il nome del file JSON che hai creato nel passaggio precedente nel `--cli-input-json` parametro:

```
aws imagebuilder create-image-recipe --cli-input-json file://create-image-
recipe.json
```

> **Note**
>
> - È necessario includere l'annotazione `file://` all'inizio del percorso del file JSON.
> - Il percorso del file JSON deve seguire la convenzione appropriata per il sistema operativo di base in cui si esegue il comando. Ad esempio, Windows utilizza la barra rovesciata (\) per fare riferimento al percorso della directory, mentre Linux e macOS utilizzano la barra (/).

L'immagine finale può contenere fino a quattro codici prodotto tratti da prodotti e componenti di Marketplace AWS immagini. Se l'immagine di base e i componenti selezionati contengono più di quattro codici prodotto, Image Builder restituisce un errore quando esegue il `create-image-recipe` comando.

Importa una macchina virtuale come immagine di base nella console

In questa sezione, ci concentriamo su come importare una macchina virtuale (VM) come immagine di base per la ricetta dell'immagine. Non trattiamo gli altri passaggi relativi alla creazione di una ricetta o di una versione di ricetta qui. Per ulteriori passaggi per creare una nuova ricetta di immagini con la procedura guidata di creazione della pipeline nella console Image Builder, vedere. Procedura guidata per la pipeline: crea AMI Per ulteriori passaggi per creare una nuova ricetta di immagini o una nuova versione di ricetta, consulta. Crea una nuova versione di una ricetta di immagini

Per importare una macchina virtuale come immagine di base per la tua ricetta di immagini nella console Image Builder, segui questi passaggi, insieme a tutti gli altri passaggi richiesti, per creare la tua ricetta o versione della ricetta.

1. Nella sezione Seleziona immagine per l'immagine di base, seleziona l'opzione Importa immagine di base.
2. Scegliete il sistema operativo (OS) dell'immagine e la versione del sistema operativo come fareste normalmente.

Configurazione di importazione delle macchine virtuali

Quando si esporta una macchina virtuale dal relativo ambiente di virtualizzazione, tale processo crea un set di uno o più file contenitore su disco che fungono da istantanee dell'ambiente, delle

impostazioni e dei dati della macchina virtuale. È possibile utilizzare questi file per importare la macchina virtuale come immagine di base per la ricetta dell'immagine. Per ulteriori informazioni sull'importazione VMs in Image Builder, vedere Importa ed esporta immagini VM

Per specificare la posizione della fonte di importazione, procedi nel seguente modo:

Fonte di importazione

Specificare la fonte per il primo contenitore o istantanea di immagini VM da importare nella sezione Contenitore disco 1.

1. Fonte: può essere un bucket S3 o un'istantanea EBS.
2. Seleziona la posizione S3 del disco: inserisci la posizione in Amazon S3 in cui sono archiviate le immagini del disco. Per cercare la posizione, scegli Browse S3.
3. Per aggiungere un contenitore per dischi, scegli Aggiungi contenitore disco.

Ruolo IAM

Per associare un ruolo IAM alla configurazione di importazione delle macchine virtuali, seleziona il ruolo dall'elenco a discesa dei ruoli IAM o scegli Crea nuovo ruolo per crearne uno nuovo. Se crei un nuovo ruolo, la pagina della console IAM Roles si apre in una scheda separata.

Impostazioni avanzate: facoltative

Le seguenti impostazioni sono opzionali. Con queste impostazioni, è possibile configurare la crittografia, le licenze, i tag e altro per l'immagine di base creata dall'importazione.

Generali

1. Specificate un nome univoco per l'immagine di base. Se non si inserisce un valore, l'immagine di base eredita il nome della ricetta.
2. Specificate una versione per l'immagine di base. Utilizza il seguente formato: *<major>.<minor>.<patch>*. Se non si inserisce un valore, l'immagine di base eredita la versione della ricetta.
3. È inoltre possibile inserire una descrizione per l'immagine di base.

Architettura dell'immagine di base

Per specificare l'architettura della sorgente di importazione delle macchine virtuali, selezionare un valore dall'elenco Architettura.

Crittografia

Se le immagini del disco della macchina virtuale sono crittografate, è necessario fornire una chiave da utilizzare per il processo di importazione. Per specificare un valore AWS KMS key per l'importazione, selezionare un valore dall'elenco Encryption (chiave KMS). L'elenco contiene le chiavi KMS a cui il tuo account ha accesso nella regione corrente.

Gestione delle licenze

Quando si importa una macchina virtuale, il processo di importazione rileva automaticamente il sistema operativo della macchina virtuale e applica la licenza appropriata all'immagine di base. A seconda della piattaforma del sistema operativo, i tipi di licenza sono i seguenti:

- Licenza inclusa: all'immagine di base viene applicata una AWS licenza appropriata per la piattaforma in uso.
- Bring your own license (BYOL): conserva la licenza della macchina virtuale, se applicabile.

Per allegare le configurazioni di licenza create con AWS License Manager all'immagine di base, selezionatele dall'elenco dei nomi di configurazione della licenza. Per ulteriori informazioni su License Manager, vedere Working with AWS License Manager

> **Note**
>
> - Le configurazioni delle licenze contengono regole di licenza basate sui termini dei contratti aziendali.
> - Linux supporta solo le licenze BYOL.

Tag (immagine di base)

I tag utilizzano coppie chiave-valore per assegnare testo ricercabile alla risorsa Image Builder. Per specificare i tag per l'immagine di base importata, inserite le coppie chiave-valore nelle caselle Chiave e Valore.

Per aggiungere un tag, scegli Add tag (Aggiungi tag). Per rimuovere un tag, scegli Remove (Rimuovi).

Crea una nuova versione di una ricetta contenitore

Questa sezione mostra come creare una nuova versione di una ricetta contenitore.

Indice

- Crea una nuova versione di ricetta in contenitore con la console
- Crea una ricetta in un contenitore con AWS CLI

Crea una nuova versione di ricetta in contenitore con la console

Creare una nuova versione di una ricetta contenitore è praticamente la stessa cosa che creare una nuova ricetta. La differenza è che alcuni dettagli sono preselezionati in base alla ricetta di base, nella maggior parte dei casi. L'elenco seguente descrive le differenze tra la creazione di una nuova ricetta e la creazione di una nuova versione di una ricetta esistente.

Dettagli della ricetta

- Nome: non modificabile.
- Versione: obbligatoria. Questo dettaglio non è precompilato con la versione corrente o con alcun tipo di sequenza. Immettete il numero di versione che desiderate creare nel formato major.minor.patch. Se la versione esiste già, si verifica un errore.

Immagine di base

- Opzione Seleziona immagine: preselezionata, ma modificabile. Se modificate la scelta della fonte dell'immagine di base, potreste perdere altri dettagli che dipendono dall'opzione originale scelta.

  Per le immagini dei container Docker, puoi scegliere tra immagini pubbliche ospitate su DockerHub, immagini di container esistenti in Amazon ECR o immagini di container gestite da Amazon. Per visualizzare i dettagli associati alla selezione dell'immagine di base, scegli la scheda corrispondente alla selezione.

  Managed images

  - Image Operating System (OS): non modificabile.
  - Nome immagine: preselezionato, in base alla combinazione di immagini di base scelte per la ricetta esistente. Tuttavia, se si modifica l'opzione Seleziona immagine, si perde il nome dell'immagine preselezionato.

- Opzioni di creazione automatica delle versioni: non corrisponde alla ricetta base. Per impostazione predefinita, le opzioni di controllo automatico delle versioni utilizzano l'opzione Usa la versione del sistema operativo selezionata.

> **Important**
>
> Se utilizzi il controllo delle versioni semantiche per avviare le build della pipeline, assicurati di modificare questo valore in Usa l'ultima versione del sistema operativo disponibile. Per ulteriori informazioni sul controllo delle versioni semantiche per le risorse di Image Builder, vedere. Versionamento semantico in Image Builder

ECR image

- Image Operating System (OS): preselezionato, ma modificabile.
- Versione del sistema operativo: preselezionata, ma modificabile.
- ID immagine ECR: precompilato, ma modificabile.

Docker Hub image

- Image Operating System (OS): non modificabile.
- Versione del sistema operativo: preselezionata, ma modificabile.
- ID immagine Docker: precompilato, ma modificabile.

Configurazione dell'istanza

- Sorgente AMI (obbligatorio): identifica un AMI personalizzato da utilizzare come immagine di base per la creazione e l'istanza di test del contenitore. Può essere un ID AMI o un parametro AWS Systems Manager (SSM) Parameter Store che contiene un ID AMI.
  - ID AMI: questa impostazione non è precompilata con l'immissione originale. Inserisci l'ID AMI per l'immagine di base. Esempio: *ami-1234567890abcdef1*.
  - Parametro SSM: inserisci il nome o l'ARN del parametro SSM Parameter Store che contiene l'ID AMI per l'immagine di base. Esempio: */ib/test/param* o `arn:aws:ssm:`*us-east-1*`:`*111122223333*`:parameter`*/ib/test/param*.
- Archiviazione (volumi)

  EBS volume 1 (AMI root) — Precompilato. Non è possibile modificare le selezioni Device Name, Snapshot o IOPS del volume root. Tuttavia, è possibile modificare tutte le impostazioni rimanenti, ad esempio la dimensione. Puoi anche aggiungere nuovi volumi.

> **Note**
>
> Se hai specificato un'AMI di base condivisa con te da un altro account, anche le istantanee per tutti i volumi secondari specificati devono essere condivise con il tuo account.

Directory di lavoro

- Percorso della directory di lavoro: precompilato, ma modificabile.

Componenti

- Componenti: i componenti già inclusi nella ricetta vengono visualizzati nella sezione Componenti selezionati alla fine di ciascuno degli elenchi di componenti (build e test). È possibile rimuovere o riordinare i componenti selezionati in base alle proprie esigenze.

  I componenti di protezione avanzata CIS non seguono le regole standard di ordinamento dei componenti nelle ricette di Image Builder. I componenti di protezione avanzata CIS funzionano sempre per ultimi per garantire che i test di benchmark vengano eseguiti rispetto all'immagine di output.

  > **Note**
  >
  > Gli elenchi dei componenti di compilazione e test mostrano i componenti disponibili in base al tipo di proprietario del componente. Per aggiungere un componente, scegli Aggiungi componenti di compilazione e seleziona il filtro di proprietà applicabile. Ad esempio, per aggiungere un componente di compilazione associato a un Marketplace AWS prodotto, seleziona`Marketplace AWS`. Si apre un pannello di selezione sul lato destro dell'interfaccia della console che elenca Marketplace AWS i componenti.
  > Per il componente CIS, selezionate`Third party managed`.

  È possibile configurare le seguenti impostazioni per il componente selezionato:

  - Opzioni di controllo delle versioni: preselezionate, ma è possibile modificarle. Ti consigliamo di scegliere l'opzione Usa l'ultima versione disponibile del componente per assicurarti che le tue build di immagini utilizzino sempre la versione più recente del componente. Se devi utilizzare una

versione specifica del componente nella tua ricetta, puoi scegliere Specificare la versione del componente e inserire la versione nella casella Versione del componente che appare.

- Parametri di input: visualizza i parametri di input accettati dal componente. Il valore è precompilato con il valore della versione precedente della ricetta. Se si utilizza questo componente per la prima volta in questa ricetta ed è stato definito un valore predefinito per il parametro di input, il valore predefinito viene visualizzato nella casella Valore con testo in grigio. Se non viene immesso nessun altro valore, Image Builder utilizza il valore predefinito.

  Se un parametro di input è obbligatorio, ma non ha un valore predefinito definito nel componente, è necessario fornire un valore. Image Builder non creerà la versione della ricetta se mancano dei parametri obbligatori e non è stato definito un valore predefinito.

  > **Important**
  >
  > I parametri dei componenti sono valori di testo semplice e vengono registrati. AWS CloudTrail Ti consigliamo di utilizzare AWS Secrets Manager o AWS Systems Manager Parameter Store per archiviare i tuoi segreti. Per ulteriori informazioni su Secrets Manager, vedi Cos'è Secrets Manager? nella Guida AWS Secrets Manager per l'utente. Per ulteriori informazioni su AWS Systems Manager Parameter Store, vedere AWS Systems Manager Parameter Store nella Guida AWS Systems Manager per l'utente.

Per espandere le impostazioni per le opzioni di controllo delle versioni o i parametri di input, puoi scegliere la freccia accanto al nome dell'impostazione. Per espandere tutte le impostazioni per tutti i componenti selezionati, è possibile disattivare e attivare l'opzione Espandi tutto.

Modello di file Dockerfile

- Modello Dockerfile: precompilato, ma modificabile. È possibile specificare una qualsiasi delle seguenti variabili contestuali che Image Builder sostituisce con le informazioni di build in fase di esecuzione.

  parentImage (obbligatorio)

  In fase di compilazione, questa variabile diventa l'immagine di base della ricetta.

  Esempio:

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

ambienti (richiesti se vengono specificati i componenti)

Questa variabile si risolverà in uno script che esegue i componenti.

Esempio:

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

componenti (opzionale)

Image Builder risolve gli script dei componenti di compilazione e test per i componenti inclusi nella ricetta del contenitore. Questa variabile può essere posizionata ovunque nel Dockerfile, dopo la variabile environment.

Esempio:

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

Archivio Target

- Nome del repository di destinazione: l'archivio Amazon ECR in cui è archiviata l'immagine di output se non è specificato nessun altro repository nella configurazione di distribuzione della pipeline per la regione in cui viene eseguita la pipeline (Regione 1).

Per creare una nuova versione della ricetta del contenitore:

1. Nella parte superiore della pagina dei dettagli della ricetta del contenitore, scegli Crea nuova versione. Verrai indirizzato alla pagina Crea ricetta per le ricette in contenitore.
2. Per creare la nuova versione, apporta le modifiche, quindi scegli Crea ricetta.

Per ulteriori informazioni su come creare una ricetta contenitore quando si crea una pipeline di immagini, consulta Fase 2: Scegli la ricetta la sezione Guida introduttiva di questa guida.

Crea una ricetta in un contenitore con AWS CLI

Per creare una ricetta contenitore Image Builder con il `imagebuilder create-container-recipe` comando in AWS CLI, procedi nel seguente modo:

Prerequisiti

Prima di eseguire i comandi Image Builder in questa sezione per creare una ricetta contenitore con AWS CLI, è necessario creare i componenti che la ricetta utilizzerà. L'esempio di ricetta del contenitore riportato nel passaggio seguente si riferisce ai componenti di esempio creati nella Creare un componente personalizzato dal AWS CLI sezione di questa guida.

Dopo aver creato i componenti, o se state utilizzando componenti esistenti, prendete nota di quelli ARNs che desiderate includere nella ricetta.

1. Creazione di un file JSON di input nella CLI

   È possibile fornire tutto l'input per il create-container-recipe comando con parametri di comando in linea. Tuttavia, il comando risultante può essere piuttosto lungo. Per semplificare il comando, puoi invece fornire un file JSON che contenga tutte le impostazioni delle ricette del contenitore

   > **Note**
   >
   > La convenzione di denominazione per i valori dei dati nel file JSON segue lo schema specificato per i parametri di richiesta operativa dell'API Image Builder. Per esaminare i parametri di richiesta dell'operazione API, vedere il CreateContainerRecipecomando nell'EC2 Image Builder API Reference.
   > Per fornire i valori dei dati come parametri della riga di comando, fate riferimento ai nomi dei parametri specificati nel AWS CLI Command Reference.

   Ecco un riepilogo dei parametri di questo esempio:

   - components (array di oggetti, obbligatorio) — Contiene un array di `ComponentConfiguration` oggetti. È necessario specificare almeno un componente di build:

> **Note**
>
> Image Builder installa i componenti nell'ordine specificato nella ricetta. Tuttavia, i componenti di protezione avanzata CIS funzionano sempre per ultimi per garantire che i test di benchmark vengano eseguiti rispetto all'immagine di output.

- componentARN (stringa, obbligatorio) — L'ARN del componente.

  > **Tip**
  >
  > Per utilizzare l'esempio per creare la tua ricetta di contenitore, sostituisci l'esempio ARNs con la ARNs per i componenti che stai utilizzando per la tua ricetta,. Questi includono Regione AWS il nome e il numero di versione di ciascuno.

- parametri (matrice di oggetti): contiene una matrice di `ComponentParameter` oggetti. Se un parametro di input è obbligatorio, ma non ha un valore predefinito definito nel componente, è necessario fornire un valore. Image Builder non creerà la versione della ricetta se mancano dei parametri obbligatori e non è stato definito un valore predefinito.

  > **Important**
  >
  > I parametri dei componenti sono valori di testo semplice e vengono registrati. AWS CloudTrail Ti consigliamo di utilizzare AWS Secrets Manager o AWS Systems Manager Parameter Store per archiviare i tuoi segreti. Per ulteriori informazioni su Secrets Manager, vedi Cos'è Secrets Manager? nella Guida AWS Secrets Manager per l'utente. Per ulteriori informazioni su AWS Systems Manager Parameter Store, vedere AWS Systems Manager Parameter Store nella Guida AWS Systems Manager per l'utente.

  - name (string, required) — Il nome del parametro del componente da impostare.
  - value (array of string, obbligatorio) — Contiene una matrice di stringhe per impostare il valore per il parametro del componente denominato. Se è stato definito un valore predefinito per il componente e non viene fornito nessun altro valore, AWSTOE utilizza il valore predefinito.

- containerType (string, required) — Il tipo di contenitore da creare. I valori validi includono. `DOCKER`
- dockerfileTemplateData(stringa) — Il modello Dockerfile utilizzato per creare l'immagine, espresso come blob di dati in linea.
- name (string, required) — Il nome della ricetta del contenitore.
- description (string) — La descrizione della ricetta del contenitore.
- parentImage (string, required) — L'immagine del contenitore Docker da utilizzare nella ricetta del contenitore come base per l'immagine personalizzata.
  - Immagini pubbliche ospitate su DockerHub
  - Immagini di container esistenti in Amazon ECR
  - Immagini di container gestite da Amazon
- PlatformOverride (stringa): specifica la piattaforma del sistema operativo quando si utilizza un'immagine di base personalizzata.
- SemanticVersion <major>(string, required) — La versione semantica della ricetta del contenitore specificata nel formato seguente, con valori numerici in ogni posizione per indicare una versione specifica:. <minor>. <patch>. Ad esempio, sarebbe `1.0.0`. Per ulteriori informazioni sul controllo delle versioni semantiche per le risorse di Image Builder, vedere. Versionamento semantico in Image Builder
- tags (string map) — Tag allegati alla ricetta del contenitore.
- instanceConfiguration (oggetto): un gruppo di opzioni che possono essere utilizzate per configurare un'istanza per la creazione e il test delle immagini dei container.
  - image (string) — L'immagine di base per un'istanza di compilazione e test di un container. Può contenere un ID AMI o specificare un parametro AWS Systems Manager (SSM) Parameter Store, preceduto da`ssm:`, seguito dal nome del parametro o dall'ARN. Se si utilizza un parametro SSM, il valore del parametro deve contenere un ID AMI. Se non specifichi un'immagine di base, Image Builder utilizza l'AMI ottimizzata Amazon ECS appropriata come immagine di base.
  - blockDeviceMappings(matrice di oggetti) — Definisce i dispositivi a blocchi da collegare per creare un'istanza dall'AMI Image Builder specificata nel image parametro.
    - deviceName (stringa) — Il dispositivo a cui si applicano queste mappature.
    - ebs (object) — Utilizzato per gestire la configurazione specifica di Amazon EBS per questa mappatura.

    - deleteOnTermination(Boolean) — Utilizzato per configurare l'eliminazione alla chiusura del dispositivo associato.
    - criptato (booleano) — Utilizzato per configurare la crittografia del dispositivo.
    - VolumeSize (numero intero): utilizzato per sovrascrivere la dimensione del volume del dispositivo.
    - volumeType (string) — Utilizzato per sovrascrivere il tipo di volume del dispositivo.
- targetRepository (oggetto, obbligatorio): l'archivio di destinazione per l'immagine del contenitore se non è specificato nessun altro repository nella configurazione di distribuzione della pipeline per la regione in cui viene eseguita la pipeline (Regione 1).
    - repositoryName (stringa, obbligatorio) — Il nome del repository del contenitore in cui è archiviata l'immagine del contenitore di output. Questo nome ha come prefisso la posizione del repository.
    - service (string, required): specifica il servizio in cui è stata registrata l'immagine.
- workingDirectory (string) — La directory di lavoro da utilizzare durante i flussi di lavoro di compilazione e test.

```
{
 "components": [
   {
   "componentArn": "arn:aws:imagebuilder:us-west-2:111122223333:component/
helloworldal2/x.x.x"
   }
 ],
 "containerType": "DOCKER",
 "description": "My Linux Docker container image",
 "dockerfileTemplateData": "FROM
 {{{ imagebuilder:parentImage }}}\n{{{ imagebuilder:environments }}}\n{{{ imagebuilder:comp
 "name": "amazonlinux-container-recipe",
 "parentImage": "amazonlinux:latest",
 "platformOverride": "Linux",
 "semanticVersion": "1.0.2",
 "tags": {
   "sometag" : "Tag detail"
 },
 "instanceConfiguration": {
   "image": "ami-1234567890abcdef1",
   "blockDeviceMappings": [
   {
```

```
    "deviceName": "/dev/xvda",
    "ebs": {
     "deleteOnTermination": true,
     "encrypted": false,
     "volumeSize": 8,
     "volumeType": "gp2"
     }
     }
    ]
 },
 "targetRepository": {
    "repositoryName": "myrepo",
    "service": "ECR"
 },
 "workingDirectory": "/tmp"
}
```

2. Crea la ricetta

   Usa il seguente comando per creare la ricetta. Fornisci il nome del file JSON che hai creato nel passaggio precedente nel `--cli-input-json` parametro:

   ```
   aws imagebuilder create-container-recipe --cli-input-json file://create-container-recipe.json
   ```

   > **Note**
   >
   > - È necessario includere l'annotazione `file://` all'inizio del percorso del file JSON.
   > - Il percorso del file JSON deve seguire la convenzione appropriata per il sistema operativo di base in cui si esegue il comando. Ad esempio, Windows utilizza la barra rovesciata (\) per fare riferimento al percorso della directory, mentre Linux e macOS utilizzano la barra (/).

Pulizia delle risorse

Per evitare addebiti imprevisti, assicurati di ripulire le risorse e le pipeline che hai creato sulla base degli esempi di questa guida. Per ulteriori informazioni sull'eliminazione di risorse in Image Builder, vedere. Eliminare risorse Image Builder obsolete o inutilizzate

Gestione della configurazione dell'infrastruttura Image Builder

Puoi utilizzare le configurazioni dell'infrastruttura per specificare l' EC2 infrastruttura Amazon utilizzata da Image Builder per creare e testare la tua immagine Image EC2 Builder. Le impostazioni dell'infrastruttura includono:

- Tipi di istanze per la tua infrastruttura di build e test. Si consiglia di specificare più di un tipo di istanza perché ciò consente a Image Builder di avviare un'istanza da un pool con capacità sufficiente. In questo modo è possibile ridurre gli errori di compilazione temporanei.

  Per le immagini Mac, potresti scegliere tipi di istanze che supportano nativamente il sistema operativo macOS. Per ulteriori informazioni, consulta le istanze Amazon EC2 Mac nella Amazon EC2 User Guide.
- Impostazioni di posizionamento delle istanze in cui puoi specificare l'host, il gruppo di collocamento dell'host o la zona di disponibilità in cui devono essere collocate le istanze che vengono avviate dalla tua immagine.
- Un profilo di istanza che fornisce alle istanze di compilazione e test le autorizzazioni necessarie per eseguire attività di personalizzazione. Ad esempio, se disponi di un componente che recupera risorse da Amazon S3, il profilo dell'istanza richiede le autorizzazioni per accedere a tali file. Il profilo dell'istanza richiede anche un set minimo di autorizzazioni per consentire a EC2 Image Builder di comunicare correttamente con l'istanza. Per ulteriori informazioni, consulta Preparati per creare immagini personalizzate con Image Builder.
- I VPC, la sottorete e i gruppi di sicurezza per le istanze di compilazione e test della pipeline.
- La posizione Amazon S3 in cui Image Builder archivia i log delle applicazioni relativi alla compilazione e al test. Se configuri la registrazione, il profilo di istanza specificato nella configurazione dell'infrastruttura deve disporre delle `s3:PutObject` autorizzazioni per il bucket di destinazione (). `arn:aws:s3:::BucketName/*`
- Una coppia di EC2 chiavi Amazon che ti consente di accedere alla tua istanza per risolvere i problemi se la compilazione fallisce e lo hai impostato`terminateInstanceOnFailure`. `false`
- Un argomento SNS in cui Image Builder invia notifiche di eventi. Per ulteriori informazioni su come Image Builder si integra con Amazon SNS, consulta. Integrazione di Amazon SNS in Image Builder

> **Note**
>
> Se l'argomento SNS è crittografato, la chiave che crittografa questo argomento deve risiedere nell'account in cui viene eseguito il servizio Image Builder. Image Builder non può inviare notifiche ad argomenti SNS crittografati con chiavi di altri account.

È possibile creare e gestire configurazioni dell'infrastruttura utilizzando la console Image Builder, tramite l'API Image Builder o imagebuilder con i comandi in. AWS CLI

Indice

- Elenca e visualizza i dettagli di configurazione dell'infrastruttura
- Creazione di una configurazione dell'infrastruttura
- Aggiornare una configurazione dell'infrastruttura
- Image Builder e endpoint AWS PrivateLink VPC di interfaccia

> **Tip**
>
> Quando hai più risorse dello stesso tipo, l'etichettatura ti aiuta a identificare una risorsa specifica in base ai tag che le hai assegnato. Per ulteriori informazioni sull'etichettatura delle risorse utilizzando i comandi Image Builder in, AWS CLI consultate Aggiunta di tag alle risorse la sezione di questa guida.

Elenca e visualizza i dettagli di configurazione dell'infrastruttura

Questa sezione descrive i vari modi in cui è possibile trovare informazioni e visualizzare i dettagli per le configurazioni EC2 dell'infrastruttura Image Builder.

Dettagli sulla configurazione dell'infrastruttura

- Elenca le configurazioni dell'infrastruttura da AWS CLI
- Ottieni i dettagli sulla configurazione dell'infrastruttura da AWS CLI

Elenca le configurazioni dell'infrastruttura da AWS CLI

L'esempio seguente mostra come elencare tutte le configurazioni dell'infrastruttura, utilizzando il list-infrastructure-configurations comando in. AWS CLI

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

Ottieni i dettagli sulla configurazione dell'infrastruttura da AWS CLI

L'esempio seguente mostra come utilizzare il get-infrastructure-configuration comando in AWS CLI per ottenere i dettagli di una configurazione dell'infrastruttura specificandone 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
```

Creazione di una configurazione dell'infrastruttura

Questa sezione descrive come utilizzare la console Image Builder o imagebuilder i comandi in AWS CLI per creare una configurazione dell'infrastruttura,

Console

Per creare una risorsa di configurazione dell'infrastruttura dalla console Image Builder, procedi nel seguente modo:

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Dal pannello di navigazione, selezionare Configurazione dell'infrastruttura.
3. Scegli Crea configurazione dell'infrastruttura.
4. Nella sezione Generale, inserisci le seguenti informazioni obbligatorie:
   - Inserisci il nome della risorsa di configurazione dell'infrastruttura.
   - Seleziona un ruolo IAM da associare al profilo dell'istanza per le autorizzazioni dei componenti sulle tue istanze di build e test. Image Builder utilizza queste autorizzazioni per scaricare ed eseguire i componenti, caricare i log ed eseguire eventuali azioni aggiuntive specificate dai componenti della ricetta. CloudWatch

5. Nel pannello dell'AWS infrastruttura, è possibile configurare le impostazioni dell'infrastruttura rimanenti disponibili. Inserisci le seguenti informazioni obbligatorie:

   - Tipo di istanza: puoi specificare uno o più tipi di istanza da utilizzare per questa build. Il servizio sceglierà uno di questi tipi di istanza in base alla disponibilità.

     > **Note**
     >
     > Le istanze Mac vengono eseguite su tipi di `.metal` istanze su un host dedicato. Il tipo di istanza deve corrispondere a uno dei tipi definiti per l'host su cui viene eseguita. Per ulteriori informazioni sulle istanze Mac e un elenco di tipi di istanze che supportano nativamente il sistema operativo macOS, consulta le istanze Amazon EC2 Mac nella Amazon User Guide. EC2

   - Argomento SNS (opzionale): selezionare un argomento SNS per ricevere notifiche e avvisi da Image Builder EC2 .

   Se non fornisci valori per le seguenti impostazioni, queste utilizzano impostazioni predefinite specifiche del servizio, ove applicabile.

   - VPC, sottorete e gruppi di sicurezza: Image Builder utilizza il VPC e la sottorete predefiniti. Per ulteriori informazioni sulla configurazione degli endpoint dell'interfaccia VPC, consulta. Image Builder e endpoint AWS PrivateLink VPC di interfaccia
   - Nella sezione Impostazioni di risoluzione dei problemi, puoi configurare i seguenti valori:
     - Per impostazione predefinita, è selezionata la casella di controllo Termina istanza in caso di errore. Tuttavia, quando una compilazione fallisce, puoi accedere all' EC2 istanza per risolvere i problemi. Se desideri che l'istanza continui a funzionare dopo un errore di compilazione, deseleziona la casella di controllo.
     - Coppia di chiavi: se l' EC2 istanza continua a funzionare dopo un errore di compilazione, puoi creare una coppia di chiavi o utilizzare una coppia di chiavi esistente per accedere all'istanza e risolvere i problemi.
     - Registri: è possibile specificare un bucket S3 in cui Image Builder può scrivere i log delle applicazioni per facilitare la risoluzione dei problemi di build e test. Se non specificate un bucket S3, Image Builder scrive i log dell'applicazione nell'istanza.

- Nella sezione Impostazioni dei metadati dell'istanza, è possibile configurare i seguenti valori da applicare alle EC2 istanze utilizzate da Image Builder per creare e testare l'immagine:
  - Seleziona la versione dei metadati per determinare se è EC2 necessaria un'intestazione token firmata, ad esempio le richieste di recupero dei metadati.
    - V1 e V2 (token opzionale): valore predefinito se non si seleziona nulla.
    - V2 (token richiesto)

    > **Note**
    >
    > Si consiglia di configurare tutte le EC2 istanze che Image Builder avvia da una build di pipeline da IMDSv2 utilizzare in modo che le richieste di recupero dei metadati dell'istanza richiedano un'intestazione token firmata.

  - Limite dell'hop di risposta del token di metadati: il numero di hop di rete che il token di metadati può percorrere. Salti minimi: 1, hop massimi: 64, con il valore predefinito di un hop.
- Nella sezione Impostazioni di posizionamento delle istanze, è possibile configurare i seguenti valori da applicare alle EC2 istanze utilizzate da Image Builder per creare e testare l'immagine:
  - È possibile selezionare la zona di disponibilità in cui Image Builder avvia le istanze durante la creazione dell'immagine.
  - Facoltativamente, seleziona Tenancy per i server che eseguono le istanze avviate. Per impostazione predefinita, EC2 le istanze vengono eseguite su hardware di locazione condiviso. Ciò significa che più dispositivi Account AWS potrebbero condividere lo stesso hardware fisico. Un'istanza con `dedicated` tenancy viene eseguita su hardware single-tenant. Un'istanza con `host` tenancy viene eseguita su un host dedicato.

    Le istanze Mac richiedono un host dedicato creato come prerequisito prima di creare un'immagine personalizzata. Seleziona `host` per la tua immagine macOS. È quindi possibile selezionare un host di destinazione o un gruppo di risorse host per avviare le istanze, ma non è necessario se l'host dedicato ha abilitato il posizionamento automatico. Per ulteriori informazioni, consulta la sezione Posizionamento automatico nella Amazon EC2 User Guide.
    - ID host di tenancy: l'ID dell'host dedicato su cui vengono eseguite le istanze.

- Tenancy host resource group: Amazon Resource Name (ARN) del gruppo di risorse host in cui avviare le istanze.

6. Nella sezione Infrastructure tags (opzionale), puoi assegnare tag di metadati all' EC2 istanza Amazon che Image Builder avvia durante il processo di creazione. I tag vengono inseriti come coppie di valori chiave.
7. Nella sezione Tag (opzionale), è possibile assegnare tag di metadati alla risorsa di configurazione dell'infrastruttura creata da Image Builder come output. I tag vengono inseriti come coppie di valori chiave.

AWS CLI

La procedura seguente mostra come configurare l'infrastruttura per l'immagine con il create-infrastructure-configuration comando Image Builder in. AWS CLI Il comando nel passaggio 2 utilizza il file creato nel passaggio 1. Per questi esempi, viene chiamato il file del passaggio `1create-infrastructure-configuration.json`.

1. Creazione di un file JSON di input nella CLI

   Gli esempi seguenti mostrano le varianti del file JSON che è possibile creare per la configurazione dell'infrastruttura. Usa uno strumento di modifica dei file per creare un tuo file JSON.

   Esempio 1: configurazione per conservare un'istanza di una build fallita

   Questo esempio specifica due tipi di istanza `m5.large` `em5.xlarge`. Si consiglia di specificare più di un tipo di istanza perché ciò consente a Image Builder di avviare un'istanza da un pool con capacità sufficiente. In questo modo è possibile ridurre gli errori di compilazione temporanei.

   `instanceProfileName`specifica il profilo dell'istanza che fornisce all'istanza le autorizzazioni richieste dal profilo per eseguire attività di personalizzazione. Ad esempio, se disponi di un componente che recupera risorse da Amazon S3, il profilo dell'istanza richiede le autorizzazioni per accedere a tali file. Il profilo dell'istanza richiede anche un set minimo di autorizzazioni per consentire a EC2 Image Builder di comunicare correttamente con l'istanza. Per ulteriori informazioni, consulta Preparati per creare immagini personalizzate con Image Builder.

   ```
   {
   ```

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

Esempio 2: configurazione macOS con posizionamento automatico

Questo esempio specifica i tipi di istanza e il posizionamento per un'istanza Mac in cui l'host dedicato ha il posizionamento automatico abilitato.

```
{
   "name": "macOSInfraConfigAutoPlacement",
   "description": "An example infrastructure configuration for macOS.",
   "instanceProfileName": "EC2InstanceProfileForImageBuilder",
   "instanceTypes": ["mac1.metal, mac2.metal"],
   "terminateInstanceOnFailure": false,
   "placement": {
      "tenancy": "host"
   }
}
```

Esempio 3: configurazione macOS con ID host specificato

Questo esempio specifica il tipo e il posizionamento di un'istanza Mac destinata a un host dedicato specifico.

```
{
   "name": "macOSInfraConfigHostPlacement",
   "description": "An example infrastructure configuration for macOS.",
   "instanceProfileName": "EC2InstanceProfileForImageBuilder",
   "instanceTypes": ["mac2-m1ultra.metal"],
   "terminateInstanceOnFailure": false,
   "placement": {
      "tenancy": "host",
      "hostId" : "h-1234567890abcdef0"
   }
}
```

2. Utilizzate il file creato come input quando eseguite il comando seguente.

```
aws imagebuilder create-infrastructure-configuration --cli-input-json
 file://create-infrastructure-configuration.json
```

Aggiornare una configurazione dell'infrastruttura

Questa sezione illustra come utilizzare la console Image Builder o imagebuilder i comandi in AWS CLI per aggiornare una risorsa di configurazione dell'infrastruttura. Per tenere traccia delle risorse, è possibile applicare i tag come segue. I tag vengono inseriti come coppie di valori chiave.

- I tag di risorsa assegnano tag di metadati all' EC2 istanza Amazon che Image Builder avvia durante il processo di creazione.
- I tag assegnano tag di metadati alla risorsa di configurazione dell'infrastruttura creata da Image Builder come output.

Console

È possibile modificare i seguenti dettagli di configurazione dell'infrastruttura dalla console Image Builder:

- La descrizione della configurazione dell'infrastruttura.
- Il ruolo IAM da associare al profilo dell'istanza.
- AWS infrastruttura, incluso il tipo di istanza e un argomento SNS per le notifiche.
- VPC, sottorete e gruppi di sicurezza.

- Impostazioni per la risoluzione dei problemi, tra cui Terminate Instance in caso di errore, la coppia di chiavi per la connessione e una posizione opzionale del bucket S3 per i log delle istanze.

Per aggiornare una risorsa di configurazione dell'infrastruttura dalla console Image Builder, procedi nel seguente modo:

Scegli una configurazione dell'infrastruttura Image Builder esistente

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Per visualizzare un elenco delle risorse di configurazione dell'infrastruttura disponibili nel tuo account, scegli Configurazione dell'infrastruttura dal pannello di navigazione.
3. Per visualizzare i dettagli o modificare una configurazione dell'infrastruttura, scegli il link Nome della configurazione. Verrà aperta la visualizzazione dettagliata della configurazione dell'infrastruttura.

   > **Note**
   >
   > Puoi anche selezionare la casella di controllo accanto al nome della configurazione, quindi scegliere Visualizza dettagli.

4. Dall'angolo in alto a destra del pannello dei dettagli dell'infrastruttura, scegli Modifica.
5. Quando sei pronto a salvare gli aggiornamenti che hai apportato alla configurazione dell'infrastruttura, scegli Salva modifiche.

AWS CLI

L'esempio seguente mostra come aggiornare la configurazione dell'infrastruttura per l'immagine con il update-infrastructure-configuration comando Image Builder in. AWS CLI

1. Creazione di un file JSON di input nella CLI

   Questo esempio di configurazione dell'infrastruttura utilizza le stesse impostazioni dell'esempio create, tranne per il fatto che abbiamo aggiornato l'`terminateInstanceOnFailure`impostazione `a false`. Dopo aver eseguito il update-infrastructure-configuration comando, le pipeline che utilizzano questa configurazione

dell'infrastruttura interrompono le istanze di compilazione e test quando la compilazione fallisce.

Utilizza uno strumento di modifica dei file per creare un file JSON con le chiavi mostrate nell'esempio seguente, oltre a valori validi per il tuo ambiente. In questo esempio viene utilizzato un file denominato `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. Usa il file che hai creato come input quando esegui il comando seguente.

```
aws imagebuilder update-infrastructure-configuration --cli-input-json
 file://update-infrastructure-configuration.json
```

Image Builder e endpoint AWS PrivateLink VPC di interfaccia

È possibile stabilire una connessione privata tra il VPC e Image EC2 Builder creando un endpoint VPC di interfaccia. Gli endpoint di interfaccia sono alimentati da AWS PrivateLink, una tecnologia che consente di accedere in modo privato a Image APIs Builder senza un gateway Internet, un dispositivo

NAT, una connessione VPN o una connessione. AWS Direct Connect Le istanze nel tuo VPC non necessitano di indirizzi IP pubblici per comunicare con Image Builder. APIs Il traffico tra il tuo VPC e Image Builder non esce dalla rete Amazon.

Ogni endpoint dell'interfaccia è rappresentato da una o più interfacce di rete elastiche nelle tue sottoreti. Quando crei una nuova immagine, puoi specificare il subnet-id VPC nella configurazione dell'infrastruttura.

> **Note**
>
> Ogni servizio a cui accedi dall'interno di un VPC ha un proprio endpoint di interfaccia, con una propria policy per gli endpoint. Image Builder scarica l'applicazione di gestione AWSTOE dei componenti e accede alle risorse gestite dai bucket S3 per creare immagini personalizzate. Per concedere l'accesso a tali bucket, è necessario aggiornare la policy degli endpoint S3 per consentirlo. Per ulteriori informazioni, consulta Politiche personalizzate per l'accesso ai bucket S3.

Per ulteriori informazioni sugli endpoint dei VPC, consulta Endpoint VPC di interfaccia (AWS PrivateLink) nella Guida per l'utente di Amazon VPC.

Considerazioni sugli endpoint VPC di Image Builder

Prima di configurare un endpoint VPC di interfaccia per Image Builder, assicurati di esaminare le proprietà e le limitazioni degli endpoint dell'interfaccia nella Amazon VPC User Guide.

Image Builder supporta l'esecuzione di chiamate a tutte le sue azioni API dal tuo VPC.

Creare un endpoint VPC di interfaccia per Image Builder

Per creare un endpoint VPC per il servizio Image Builder, puoi utilizzare la console Amazon VPC o (). AWS Command Line Interface AWS CLI Per ulteriori informazioni, consulta Creazione di un endpoint dell'interfaccia nella Guida per l'utente di Amazon VPC.

Crea un endpoint VPC per Image Builder utilizzando il seguente nome di servizio:

- com.amazonaws. *region*.generatore di immagini

Se abiliti il DNS privato per l'endpoint, puoi effettuare richieste API a Image Builder utilizzando il nome DNS predefinito per la regione, ad esempio:. `imagebuilder.us-east-1.amazonaws.com`

Per cercare l'endpoint applicabile alla regione di destinazione, consulta Endpoint e quote di EC2 Image Builder nel. Riferimenti generali di Amazon Web Services

Per ulteriori informazioni, consulta Accesso a un servizio tramite un endpoint dell'interfaccia in Guida per l'utente di Amazon VPC.

Crea una policy per gli endpoint VPC per Image Builder

Puoi allegare una policy per gli endpoint all'endpoint VPC che controlla l'accesso a Image Builder. La policy specifica le informazioni riportate di seguito:

- Il principale che può eseguire operazioni.
- Le azioni che possono essere eseguite.
- Le risorse sui cui si possono eseguire azioni.

Se utilizzi componenti gestiti da Amazon nella tua ricetta, l'endpoint VPC per Image Builder deve consentire l'accesso alla seguente libreria di componenti di proprietà del servizio:

`arn:aws:imagebuilder:`*`region`*`:aws:component/*`

> **Important**
>
> Quando viene applicata una policy non predefinita a un endpoint VPC di interfaccia per Image EC2 Builder, alcune richieste API non riuscite, ad esempio quelle `RequestLimitExceeded` provenienti da, potrebbero non essere registrate su Amazon. AWS CloudTrail CloudWatch

Per ulteriori informazioni, consulta Controllo degli accessi ai servizi con endpoint VPC nella Guida per l'utente di Amazon VPC.

Politiche personalizzate per l'accesso ai bucket S3

Image Builder utilizza un bucket S3 disponibile pubblicamente per archiviare e accedere a risorse gestite, come i componenti. Inoltre, scarica l'applicazione per la gestione dei AWSTOE componenti da un bucket S3 separato. Se utilizzi un endpoint VPC per Amazon S3 nel tuo ambiente, dovrai assicurarti che la policy degli endpoint VPC S3 consenta a Image Builder di accedere ai seguenti bucket S3. I nomi dei bucket sono univoci per AWS regione () e ambiente applicativo (). *`region`* *`environment`* Image Builder e AWSTOE supportano i seguenti ambienti applicativi: `prod`, `preprod`, e. `beta`

- Il bucket di gestione dei AWSTOE componenti:

```
s3://ec2imagebuilder-toe-region-environment
```

  Esempio: s3://ec2 imagebuilder-toe-us-west -2-prod/*

- Il bucket di risorse gestite di Image Builder:

```
s3://ec2imagebuilder-managed-resources-region-environment/components
```

  Esempio: s3://ec2 -west-2-prod/components/* imagebuilder-managed-resources-us

Esempi di policy di endpoint VPC

Questa sezione include esempi di policy personalizzate per gli endpoint VPC.

Politica generale degli endpoint VPC per le azioni di Image Builder

L'esempio seguente di policy endpoint per Image Builder nega l'autorizzazione all'eliminazione di immagini e componenti di Image Builder. La politica di esempio concede inoltre il permesso di eseguire tutte le altre azioni di 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": "*",
    }]
}
```

Limita l'accesso per organizzazione, consenti l'accesso gestito ai componenti

Il seguente esempio di policy sugli endpoint mostra come limitare l'accesso alle identità e alle risorse che appartengono alla tua organizzazione e fornire l'accesso ai componenti di Image Builder gestiti da Amazon. Sostituisci *region* e *resource-org-id* con *principal-org-id* i valori della tua organizzazione.

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

```
}
```

Policy degli endpoint VPC per l'accesso ai bucket Amazon S3

Il seguente esempio di policy per gli endpoint S3 mostra come fornire l'accesso ai bucket S3 utilizzati da Image Builder per creare immagini personalizzate. Sostituisci *region* e *environment* con i valori della tua organizzazione. Aggiungi eventuali altre autorizzazioni richieste alla policy in base ai requisiti dell'applicazione.

> **Note**
>
> Per le immagini Linux, se non specificate i dati utente nella ricetta dell'immagine, Image Builder aggiunge uno script per scaricare e installare l'agente Systems Manager sulle istanze di compilazione e test dell'immagine. Per scaricare l'agente, Image Builder accede al bucket S3 della regione di compilazione.
>
> Per garantire che Image Builder possa avviare le istanze di build e test, aggiungi la seguente risorsa aggiuntiva alla tua policy sugli endpoint 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/
*"
      ]
    }
  ]
}
```

Gestione delle impostazioni di distribuzione di Image Builder

Prima di configurare le impostazioni di distribuzione per le immagini di output, si consiglia di verificare la disponibilità di qualsiasi infrastruttura sottostante o altri requisiti per le istanze avviate dall'immagine di output nelle regioni target di distribuzione. Ad esempio, non tutte le regioni supportano gli host dedicati EC2 Mac, necessari per avviare istanze da un'immagine macOS. Per ulteriori informazioni sui tipi di istanze e sui prezzi per gli host dedicati, consulta i prezzi degli host EC2 dedicati di Amazon.

Dopo aver creato le impostazioni di distribuzione con Image Builder, è possibile gestirle utilizzando la console Image Builder, l'API Image Builder o i comandi in. imagebuilder AWS CLI Con le impostazioni di distribuzione, è possibile eseguire le seguenti azioni:

Distribuzione AMI

- Specificate il nome e la descrizione dell'AMI di uscita.
- Autorizza altre Account AWS organizzazioni e OUs ad avviare l'AMI dall'account del proprietario. All'account del proprietario vengono fatturati gli addebiti associati all'AMI.

  > **Note**
  >
  > Per rendere pubblica un'AMI, impostare l'autorizzazione di avvio account autorizzati su `all`. Per informazioni ed esempi, ModifyImageAttribute consulta Amazon EC2 API Reference.

- Crea una copia dell'AMI di output per ciascuno degli account di destinazione, delle organizzazioni e della OUs regione di destinazione specificati. Gli account, le organizzazioni e le organizzazioni destinatarie OUs possiedono le relative copie AMI e vengono fatturati gli eventuali costi associati. Per ulteriori informazioni sulla distribuzione dell'AMI a AWS Organizations e OUs, consulta Condividere un AMI con organizzazioni o OUs.
- Copia l'AMI nell'account del proprietario in altro Regioni AWS.
- Esporta dischi di immagini VM su Amazon Simple Storage Service (Amazon S3). Per ulteriori informazioni, consulta Creare impostazioni di distribuzione per i dischi VM in uscita dal AWS CLI.

Distribuzione delle immagini dei container

- Specificare il repository ECR in cui Image Builder memorizza l'immagine di output nella regione di distribuzione.

È possibile utilizzare le impostazioni di distribuzione nei seguenti modi per distribuire le immagini alle regioni, agli account AWS Organizations e alle unità organizzative di destinazione (OUs) una sola volta o con ogni creazione della pipeline:

- Fornire automaticamente immagini aggiornate a regioni, account, Organizzazioni specifici e OUs utilizzare le impostazioni di distribuzione con una pipeline di Image Builder che viene eseguita secondo una pianificazione.
- Per creare una nuova immagine e distribuirla alle regioni, agli account e alle organizzazioni specificati OUs, utilizzare le impostazioni di distribuzione con una pipeline di Image Builder che si esegue una sola volta dalla console Image Builder, utilizzando Run pipeline dal menu Azioni.
- Per creare una nuova immagine e distribuirla alle Regioni, agli account, alle Organizzazioni specificati e OUs, utilizzare le impostazioni di distribuzione con la seguente azione API o il comando Image Builder in: AWS CLI
  - L'CreateImageazione nell'API Image Builder.
  - Il create-image comando in. AWS CLI
- Per esportare dischi di immagini di macchine virtuali (VM) in bucket S3 nelle regioni di destinazione come parte del normale processo di creazione delle immagini.

> **Tip**
>
> Quando disponi di più risorse dello stesso tipo, i tag ti aiutano a identificare una risorsa specifica in base ai tag che le hai assegnato. Per ulteriori informazioni sull'etichettatura delle risorse utilizzando i comandi Image Builder in, AWS CLI consultate Aggiunta di tag alle risorse la sezione di questa guida.

Indice

- Elenca e visualizza i dettagli della configurazione della distribuzione
- Crea e aggiorna le configurazioni di distribuzione AMI
- Crea e aggiorna le impostazioni di distribuzione per le immagini dei contenitori
- Configura la distribuzione AMI tra account con Image Builder
- Configura la distribuzione AMI con un modello di EC2 lancio

Elenca e visualizza i dettagli della configurazione della distribuzione

Questa sezione descrive i vari modi in cui è possibile trovare informazioni e visualizzare i dettagli per la configurazione della distribuzione di EC2 Image Builder.

Dettagli sulla configurazione della distribuzione

- Elenca le configurazioni di distribuzione dalla console
- Visualizza i dettagli della configurazione della distribuzione dalla console
- Elenca le distribuzioni provenienti da AWS CLI
- Ottieni i dettagli sulla configurazione della distribuzione da AWS CLI

Elenca le configurazioni di distribuzione dalla console

Per visualizzare un elenco delle configurazioni di distribuzione create con il tuo account nella console Image Builder, procedi nel seguente modo:

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Scegli le impostazioni di distribuzione dal pannello di navigazione. Questo mostra un elenco delle configurazioni di distribuzione create con il tuo account.
3. Per visualizzare i dettagli o creare una nuova configurazione di distribuzione, scegli il link Nome della configurazione. Si apre la visualizzazione dettagliata delle impostazioni di distribuzione.

> **Note**
>
> Puoi anche selezionare la casella di controllo accanto al nome della configurazione, quindi scegliere Visualizza dettagli.

Visualizza i dettagli della configurazione della distribuzione dalla console

Per visualizzare i dettagli di una configurazione di distribuzione specifica utilizzando la console Image Builder, selezionare la configurazione da esaminare, seguendo i passaggi descritti in. Elenca le configurazioni di distribuzione dalla console

Nella pagina dei dettagli della distribuzione, è possibile:

- Eliminare la configurazione di distribuzione. Per ulteriori informazioni sull'eliminazione di risorse in Image Builder, vedere. Eliminare risorse Image Builder obsolete o inutilizzate
- Modificare i dettagli della distribuzione.

Elenca le distribuzioni provenienti da AWS CLI

L'esempio seguente mostra come utilizzare il list-distribution-configurations comando contenuto in AWS CLI per elencare tutte le distribuzioni.

```
aws imagebuilder list-distribution-configurations
```

Ottieni i dettagli sulla configurazione della distribuzione da AWS CLI

L'esempio seguente mostra come utilizzare il get-distribution-configuration comando in AWS CLI per ottenere i dettagli di una configurazione di distribuzione specificandone il nome 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
```

Crea e aggiorna le configurazioni di distribuzione AMI

Questa sezione descrive la creazione e l'aggiornamento delle configurazioni di distribuzione per un'AMI Image Builder.

Indice

- Prerequisiti per i parametri di output SSM
- Creare una configurazione di distribuzione AMI
- Aggiornare una configurazione di distribuzione AMI
- Crea impostazioni di distribuzione per abilitare EC2 Fast Launch per l'output AMIs
- Creare impostazioni di distribuzione per i dischi VM in uscita dal AWS CLI

Prerequisiti per i parametri di output SSM

Prima di creare una nuova configurazione di distribuzione AMI che imposti un AWS Systems Manager parametro Parameter Store (parametro SSM), assicurati di aver soddisfatto i seguenti prerequisiti.

Ruolo di esecuzione

Quando si crea una pipeline o si utilizza il comando create-image in AWS CLI, è possibile specificare un solo ruolo di esecuzione di Image Builder. Se è stato definito un ruolo di esecuzione del flusso di lavoro di Image Builder, è necessario aggiungere eventuali autorizzazioni di funzionalità aggiuntive a tale ruolo. In caso contrario, è necessario creare un nuovo ruolo personalizzato che includa le autorizzazioni richieste.

- Per memorizzare l'ID AMI di output in un parametro SSM durante la distribuzione, è necessario specificare l'`ssm:PutParameter`azione nel ruolo di esecuzione di Image Builder, con il parametro elencato come risorsa.
- Quando impostate il tipo di dati del parametro su `AWS EC2 Image` per segnalare a Systems Manager di convalidare il valore del parametro come ID AMI, dovete aggiungere anche l'`ec2:DescribeImages`azione.

Creare una configurazione di distribuzione AMI

Le configurazioni di distribuzione includono il nome dell'AMI di output, impostazioni regionali specifiche per la crittografia, le autorizzazioni di avvio e Account AWS le organizzazioni e le unità organizzative (OUs) che possono avviare l'AMI di output e le configurazioni di licenza.

Una configurazione di distribuzione consente di specificare il nome e la descrizione dell'AMI di output, autorizzare altri Account AWS ad avviare l'AMI, copiare l'AMI su altri account e replicare l'AMI in altre AWS regioni. Consente inoltre di esportare l'AMI in Amazon Simple Storage Service (Amazon S3) o di EC2 configurare Fast Launch per Windows di output. AMIs Per rendere pubblica un'AMI, impostare l'autorizzazione di avvio account autorizzati su `all`. Guarda gli esempi per rendere pubblica un'AMI su EC2 ModifyImageAttribute.

Console

Segui questi passaggi per creare una nuova configurazione di distribuzione AMI in AWS Management Console:

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Scegli le impostazioni di distribuzione dal pannello di navigazione. Questo mostra un elenco delle configurazioni di distribuzione create con il tuo account.
3. Scegli Crea impostazioni di distribuzione nella parte superiore del pannello delle impostazioni di distribuzione.
4. Nella sezione Tipo di immagine, scegli il tipo di output Amazon Machine Image (AMI).
5. Nella sezione Generale, inserisci un nome per la configurazione di distribuzione e una descrizione facoltativa.
6. Nella sezione Impostazioni della regione, inserisci i seguenti dettagli per ogni regione in cui distribuisci l'AMI:

    a. Per impostazione predefinita, l'AMI è distribuito nella regione corrente (Regione 1). La regione 1 è l'origine della distribuzione. Alcune impostazioni per la Regione 1 non sono modificabili. Per tutte le regioni che aggiungi, puoi scegliere una regione dall'elenco a discesa Regione.

    La chiave Kms identifica la AWS KMS key chiave utilizzata per crittografare i volumi EBS per l'immagine nella regione di destinazione. È importante notare che ciò non si applica all'AMI originale creata dalla build con il tuo account nella regione di origine (Regione 1). La crittografia eseguita durante la fase di distribuzione della build è valida solo per le immagini distribuite ad altri account o regioni.

    Per crittografare i volumi EBS per l'AMI creata nella regione di origine del tuo account, devi impostare la chiave KMS nella mappatura del dispositivo del blocco image recipe (Archiviazione (volumi) nella console).

    Image Builder copia l'AMI negli account Target specificati per la regione.

    > **Prerequisito**
    >
    > Per copiare un'immagine tra più account, è necessario creare il `EC2ImageBuilderDistributionCrossAccountRole` ruolo in tutti gli account del target di distribuzione e allegare la politica Policy Ec2ImageBuilderCrossAccountDistributionAccess gestita al ruolo.

Il nome dell'AMI di uscita è facoltativo. Se fornisci un nome, il nome dell'AMI di output finale include un timestamp aggiunto di quando viene creato l'AMI. Se non specificate un nome, Image Builder aggiunge il timestamp di compilazione al nome della ricetta. Ciò garantisce nomi AMI univoci per ogni build.

i. Con la condivisione AMI, puoi concedere l'accesso a determinati AWS Principal per avviare istanze dall'AMI. Se espandi la sezione di condivisione AMI, puoi inserire i seguenti dettagli:

   - Autorizzazioni di avvio: seleziona Privato se desideri mantenere privata l'AMI e consentire l'accesso a AWS Principali specifici per avviare un'istanza dall'AMI privata. Seleziona Pubblico se desideri rendere pubblica la tua AMI. Qualsiasi AWS Principal può avviare un'istanza dalla tua AMI pubblica.
   - Principali: puoi concedere l'accesso ai seguenti tipi di AWS Principal per avviare le istanze:
     - AWS account: concedi l'accesso a un account specifico AWS
     - Unità organizzativa (OU): concede l'accesso a un'unità organizzativa e a tutte le relative entità figlio. Le entità figlie includono AWS gli account OUs e.
     - Organizzazione: concedi l'accesso alla tua AWS Organizations entità e a tutte le relative entità figlie. Le entità figlie includono OUs gli AWS account.

       Innanzitutto, seleziona il tipo principale. Quindi inserisci l'ID del AWS Principal a cui desideri concedere l'accesso nella casella a destra dell'elenco a discesa. Puoi inserire più IDs tipi diversi.

ii. È possibile espandere la sezione Configurazione della licenza per allegare le configurazioni di licenza create con AWS License Manager alle immagini di Image Builder. Le configurazioni di licenza contengono regole di licenza basate sui termini dei contratti aziendali. Image Builder include automaticamente le configurazioni di licenza associate all'AMI di base.

iii. Puoi espandere la sezione di configurazione del modello di avvio per specificare un modello di EC2 avvio da utilizzare per avviare le istanze dall'AMI che hai creato.

   Se utilizzi un modello di EC2 avvio, puoi indicare a Image Builder di creare una nuova versione del modello di lancio che includa l'ID AMI più recente dopo

il completamento della build. Per aggiornare il modello di avvio, configura le impostazioni come segue:

- Nome modello di avvio: selezionare il nome del modello di lancio che si desidera aggiornare con Image Builder.
- Imposta la versione predefinita: seleziona questa casella di controllo per aggiornare la versione predefinita del modello di lancio alla nuova versione.

Per aggiungere un'altra configurazione del modello di lancio, scegli Aggiungi configurazione del modello di lancio. Puoi avere fino a cinque configurazioni di modelli di lancio per regione.

iv. Puoi espandere la sezione Configurazioni dei parametri SSM per configurare un parametro SSM che memorizzerà l'ID AMI di output per l'immagine distribuita nella regione di destinazione. Facoltativamente, puoi specificare un account di distribuzione nella regione.

Nome del parametro: immettere il nome del parametro. Ad esempio `/output/image/param`.

Tipo di dati: mantiene il valore predefinito (`AWS EC2 Image`). Ciò indica a Systems Manager di convalidare il valore del parametro per garantire che sia un ID AMI valido.

b. Per aggiungere impostazioni di distribuzione per un'altra regione, scegli Aggiungi regione.

7. Scegli Crea impostazioni quando hai finito.

AWS CLI

L'esempio seguente mostra come utilizzare il create-distribution-configuration comando per creare una nuova configurazione di distribuzione per l'AMI, utilizzando il AWS CLI.

1. Creazione di un file JSON di input nella CLI

Utilizzate uno strumento di modifica dei file per creare un file JSON con le chiavi mostrate in uno degli esempi seguenti e valori validi per il vostro ambiente. Questi esempi definiscono quali Account AWS unità organizzative (OUs) sono autorizzate a lanciare l'AMI che

distribuisci nelle regioni specificate. AWS Organizations Assegnate un nome al file`create-ami-distribution-configuration.json`, da utilizzare nel passaggio successivo:

Esempio 1: Distribuisci a Account AWS

Questo esempio distribuisce un AMI in due regioni e specifica Account AWS che dispongono delle autorizzazioni di avvio in ciascuna regione.

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

```
}
```

Esempio 2: Distribuzione a Organizzazioni e OUs

Questo esempio distribuisce un AMI nella regione di origine e specifica l'organizzazione e le autorizzazioni di avvio dell'unità organizzativa.

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

Esempio 3: memorizza l'ID AMI di output in un parametro SSM

Questo esempio memorizza l'ID AMI di output in un AWS Systems Manager parametro Parameter Store nella regione di distribuzione.

```
{
 "name": "SSMParameterOutputAMI",
 "description": "Updates an SSM parameter with the output AMI ID for the
 distribution.",
 "distributions": [
  {
   "region": "us-west-2",
   "amiDistributionConfiguration": {
```

```
    "name": "Name {{ imagebuilder:buildDate }}"
   },
   "ssmParameterConfigurations": [
    {
     "amiAccountId": "111122223333",
     "parameterName": "/output/image/param",
     "dataType": "aws:ec2:image"
    }
   ]
  }
 ]
}
```

2. Esegui il comando seguente utilizzando il file creato come input.

```
aws imagebuilder create-distribution-configuration --cli-input-json
 file://create-ami-distribution-configuration.json
```

> **Note**
>
> - È necessario includere l'annotazione `file://` all'inizio del percorso del file JSON.
> - Il percorso del file JSON deve seguire la convenzione appropriata per il sistema operativo di base in cui si esegue il comando. Ad esempio, Windows utilizza la barra rovesciata (\) per fare riferimento al percorso della directory, mentre Linux e macOS utilizzano la barra (/).

Per informazioni più dettagliate, consulta la sezione Command create-distribution-configuration Reference.AWS CLI

Aggiornare una configurazione di distribuzione AMI

È possibile modificare la configurazione di distribuzione AMI. Tuttavia, le modifiche apportate non si applicano alle risorse che Image Builder ha già distribuito. Ad esempio, se hai distribuito un AMI in una regione che successivamente rimuovi dalla distribuzione, l'AMI che era già distribuito rimane in quella regione finché non lo rimuovi manualmente.

AWS Management Console

Segui questi passaggi per una configurazione di distribuzione AMI in AWS Management Console:

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Scegli le impostazioni di distribuzione dal pannello di navigazione. Questo mostra un elenco delle configurazioni di distribuzione create con il tuo account.
3. Per visualizzare i dettagli o aggiornare una configurazione di distribuzione, scegli il link Nome della configurazione. Si apre la visualizzazione dettagliata delle impostazioni di distribuzione.

   > **Note**
   >
   > Puoi anche selezionare la casella di controllo accanto al nome della configurazione, quindi scegliere Visualizza dettagli.

4. Per modificare la configurazione di distribuzione, scegli Modifica nell'angolo in alto a destra della sezione Dettagli di distribuzione. Alcuni campi sono bloccati, come il nome della configurazione di distribuzione e la regione predefinita visualizzata come Regione 1. Per ulteriori informazioni sulle impostazioni di configurazione della distribuzione, vedereCreare una configurazione di distribuzione AMI.
5. Al termine, scegli Save changes (Salva modifiche).

AWS CLI

L'esempio seguente mostra come utilizzare il update-distribution-configuration comando per aggiornare le impostazioni di distribuzione per l'AMI, utilizzando il AWS CLI.

1. Creazione di un file JSON di input nella CLI

   Utilizzate uno strumento di modifica dei file per creare un file JSON con le chiavi mostrate nell'esempio seguente e valori validi per il vostro ambiente. Questo esempio utilizza un file denominato. `update-ami-distribution-configuration.json`

   ```
   {
    "distributionConfigurationArn": "arn:aws:imagebuilder:us-
   west-2:123456789012:distribution-configuration/update-ami-distribution-
   configuration.json",
   ```

```
 "description": "Copies AMI to eu-west-2, and specifies accounts that can launch
 instances in each Region.",
 "distributions": [
   {
   "region": "us-west-2",
   "amiDistributionConfiguration": {
    "name": "Name {{imagebuilder:buildDate}}",
    "description": "An example image name with parameter references",
    "launchPermissions": {
     "userIds": [
      "987654321012"
     ]
    }
   }
  },
  {
   "region": "eu-west-2",
   "amiDistributionConfiguration": {
    "name": "My {{imagebuilder:buildVersion}} image {{imagebuilder:buildDate}}",
    "tags": {
     "KeyName": "Some value"
    },
    "launchPermissions": {
     "userIds": [
      "100000000001"
     ]
    }
   }
  }
 ]
}
```

2. Esegui il comando seguente utilizzando il file creato come input.

```
aws imagebuilder update-distribution-configuration --cli-input-json
 file://update-ami-distribution-configuration.json
```

> **Note**
>
> - È necessario includere l'annotazione `file://` all'inizio del percorso del file JSON.
> - Il percorso del file JSON deve seguire la convenzione appropriata per il sistema operativo di base in cui si esegue il comando. Ad esempio, Windows utilizza la

barra rovesciata (\) per fare riferimento al percorso della directory, mentre Linux e macOS utilizzano la barra (/).

Per informazioni più dettagliate, consulta la sezione Command update-distribution-configuration Reference.AWS CLI Per aggiornare i tag per la risorsa di configurazione della distribuzione, consulta la Aggiunta di tag alle risorse sezione.

Crea impostazioni di distribuzione per abilitare EC2 Fast Launch per l'output AMIs

L'esempio seguente mostra come utilizzare il create-distribution-configuration comando per creare impostazioni di distribuzione con EC2 Fast Launch configurato per l'AMI, da AWS CLI.

> **Note**
>
> Image Builder non supporta la distribuzione tra account se EC2 Fast Launch è AMIs preabilitato. EC2 Fast Launch deve essere abilitato dall'account di destinazione.

1. Creazione di un file JSON di input nella CLI

   Utilizzate uno strumento di modifica dei file per creare un file JSON con chiavi, come mostrato nell'esempio seguente, più valori validi per il vostro ambiente.

   Questo esempio avvia le istanze per tutte le relative risorse di destinazione contemporaneamente, poiché il numero massimo di avvii paralleli è maggiore del conteggio delle risorse di destinazione. Questo file è denominato `ami-dist-config-win-fast-launch.json` nell'esempio di comando illustrato nel passaggio successivo.

```
{
"name": "WinFastLaunchDistribution",
"description": "An example of Windows AMI EC2 Fast Launch settings in the
 distribution configuration.",
"distributions": [
 {
  "region": "us-west-2",
  "amiDistributionConfiguration": {
   "name": "Name {{imagebuilder:buildDate}}",
   "description": "Includes Windows AMI EC2 Fast Launch settings.",
```

```
   "amiTags": {
    "KeyName": "Some Value"
   }
  },
  "fastLaunchConfigurations": [{
   "enabled": true,
   "snapshotConfiguration": {
    "targetResourceCount": 5
   },
   "maxParallelLaunches": 6,
   "launchTemplate": {
    "launchTemplateId": "lt-0ab1234c56d789012",
    "launchTemplateVersion": "1"
    }
  }],
  "launchTemplateConfigurations": [{
      "launchTemplateId": "lt-0ab1234c56d789012",
      "setDefaultVersion": true
    }]
 }]
}
```

> **Note**
>
> È possibile specificare il `launchTemplateName` posto del `launchTemplateId` nella `launchTemplate` sezione, ma non è possibile specificare sia il nome che l'ID.

2. Esegui il comando seguente utilizzando il file creato come input.

```
aws imagebuilder create-distribution-configuration --cli-input-json file://ami-dist-config-win-fast-launch.json
```

> **Note**
>
> - È necessario includere l'annotazione `file://` all'inizio del percorso del file JSON.
> - Il percorso del file JSON deve seguire la convenzione appropriata per il sistema operativo di base in cui si esegue il comando. Ad esempio, Windows utilizza la barra rovesciata (\) per fare riferimento al percorso della directory, mentre Linux e macOS utilizzano la barra (/).

Per informazioni più dettagliate, consulta la sezione Command create-distribution-configuration Reference.AWS CLI

Creare impostazioni di distribuzione per i dischi VM in uscita dal AWS CLI

L'esempio seguente mostra come utilizzare il create-distribution-configuration comando per creare impostazioni di distribuzione che esporteranno i dischi di immagine delle macchine virtuali in Amazon S3 con ogni build di immagine.

1. Creazione di un file JSON di input nella CLI

   Puoi semplificare il create-distribution-configuration comando che usi in. AWS CLI A tale scopo, create un file JSON che contenga tutta la configurazione di esportazione che desiderate passare al comando.

   > **Note**
   >
   > La convenzione di denominazione per i valori dei dati nel file JSON segue lo schema specificato per i parametri di richiesta operativa dell'API Image Builder. Per esaminare i parametri di richiesta dell'operazione API, vedere il CreateDistributionConfiguration comando nell'EC2 Image Builder API Reference.
   > Per fornire i valori dei dati come parametri della riga di comando, fate riferimento ai nomi dei parametri specificati in AWS CLI Command Reference. Al create-distribution-configuration comando come opzioni.

   Ecco un riepilogo dei parametri che specifichiamo nell'oggetto `s3ExportConfiguration` JSON per questo esempio:

   - roleName (stringa, obbligatorio): il nome del ruolo che concede alla VM l' Import/Export autorizzazione per esportare immagini nel bucket S3.
   - diskImageFormat(stringa, obbligatorio) — Esporta l'immagine del disco aggiornata in uno dei seguenti formati supportati:
     - Disco rigido virtuale (VHD): compatibile con i prodotti di virtualizzazione Citrix Xen e Microsoft Hyper-V.

- ESX Virtual Machine Disk (VMDK) ottimizzato per lo streaming: compatibile con le versioni 4, 5 VMware e 6 di ESX e VMware vSphere.
- Raw: formato Raw.

- S3bucket (stringa, obbligatorio): il bucket S3 in cui archiviare le immagini del disco di output per la macchina virtuale.

Salva il file con nome `export-vm-disks.json`. Usa il nome del file nel comando. create-distribution-configuration

```
{
 "name": "example-distribution-configuration-with-vm-export",
 "description": "example",
 "distributions": [
  {
   "region": "us-west-2",
   "amiDistributionConfiguration": {
    "description": "example-with-vm-export"

   },
   "s3ExportConfiguration": {
    "roleName": "vmimport",
    "diskImageFormat": "RAW",
    "s3Bucket": "vm-bucket-export"
   }
  }],
 "clientToken": "abc123def4567ab"
}
```

2. Esegui il comando seguente utilizzando il file creato come input.

```
aws imagebuilder create-distribution-configuration --cli-input-json file://export-vm-disks.json
```

> **Note**
>
> - È necessario includere l'annotazione `file://` all'inizio del percorso del file JSON.
> - Il percorso del file JSON deve seguire la convenzione appropriata per il sistema operativo di base in cui si esegue il comando. Ad esempio, Windows utilizza la barra

rovesciata (\) per fare riferimento al percorso della directory, mentre Linux e macOS utilizzano la barra (/).

Per informazioni più dettagliate, consulta la sezione Command create-distribution-configuration Reference.AWS CLI

Crea e aggiorna le impostazioni di distribuzione per le immagini dei contenitori

Questa sezione descrive la creazione e l'aggiornamento delle impostazioni di distribuzione per le immagini dei contenitori Image Builder.

Indice

- Creare impostazioni di distribuzione per le immagini dei contenitori Image Builder da AWS CLI
- Aggiorna le impostazioni di distribuzione per l'immagine del contenitore dal AWS CLI

Creare impostazioni di distribuzione per le immagini dei contenitori Image Builder da AWS CLI

Una configurazione di distribuzione consente di specificare il nome e la descrizione dell'immagine del contenitore di output e di replicare l'immagine del contenitore in altre AWS regioni. È inoltre possibile applicare tag separati alla risorsa di configurazione della distribuzione e alle immagini del contenitore all'interno di ciascuna regione.

1. Creazione di un file JSON di input nella CLI

   Utilizzate il vostro strumento di modifica dei file preferito per creare un file JSON con le chiavi mostrate nell'esempio seguente, più valori validi per il vostro ambiente. In questo esempio viene utilizzato un file denominato `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. Esegui il comando seguente utilizzando il file creato come input.

```
aws imagebuilder create-distribution-configuration --cli-input-json file://create-container-distribution-configuration.json
```

> **Note**
>
> - È necessario includere l'annotazione `file://` all'inizio del percorso del file JSON.
> - Il percorso del file JSON deve seguire la convenzione appropriata per il sistema operativo di base in cui si esegue il comando. Ad esempio, Windows utilizza la barra

rovesciata (\) per fare riferimento al percorso della directory, mentre Linux e macOS utilizzano la barra (/).

Per informazioni più dettagliate, consulta la sezione Command create-distribution-configuration Reference.AWS CLI

Aggiorna le impostazioni di distribuzione per l'immagine del contenitore dal AWS CLI

L'esempio seguente mostra come utilizzare il update-distribution-configuration comando per aggiornare le impostazioni di distribuzione per l'immagine del contenitore, utilizzando il AWS CLI. È inoltre possibile aggiornare i tag per le immagini del contenitore all'interno di ciascuna regione.

1. Creazione di un file JSON di input nella CLI

   Usa il tuo strumento di modifica dei file preferito per creare un file JSON con le chiavi mostrate nell'esempio seguente, oltre a valori validi per il tuo ambiente. In questo esempio viene utilizzato un file denominato `update-container-distribution-configuration.json`:

```
{
 "distributionConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/update-container-distribution-
configuration.json",
 "description": "Distributes container image to Amazon ECR repository in two
 regions.",
 "distributions": [
  {
   "region": "us-west-2",
   "containerDistributionConfiguration": {
    "description": "My test image.",
    "targetRepository": {
     "service": "ECR",
     "repositoryName": "testrepo"
    },
    "containerTags": ["west2", "image1"]
   }
  },
  {
   "region": "us-east-2",
   "containerDistributionConfiguration": {
    "description": "My test image.",
```

```
    "targetRepository": {
     "service": "ECR",
     "repositoryName": "testrepo"
    },
      "containerTags": ["east2", "imagedist"]
   }
  }
 ]
}
```

2. Esegui il comando seguente, utilizzando il file che hai creato come input:

```
aws imagebuilder update-distribution-configuration --cli-input-json file://update-container-distribution-configuration.json
```

> **Note**
>
> - È necessario includere l'annotazione `file://` all'inizio del percorso del file JSON.
> - Il percorso del file JSON deve seguire la convenzione appropriata per il sistema operativo di base in cui si esegue il comando. Ad esempio, Windows utilizza la barra rovesciata (\) per fare riferimento al percorso della directory, mentre Linux e macOS utilizzano la barra (/).

Per informazioni più dettagliate, consulta la sezione Command update-distribution-configuration Reference.AWS CLI Per aggiornare i tag per la risorsa di configurazione della distribuzione, consulta la Aggiunta di tag alle risorse sezione.

Configura la distribuzione AMI tra account con Image Builder

Questa sezione descrive come configurare le impostazioni di distribuzione per fornire un'AMI Image Builder ad altri account da te specificati.

L'account di destinazione può quindi avviare o modificare l'AMI, se necessario.

> **Note**
>
> AWS CLI gli esempi di comandi in questa sezione presuppongono che siano stati precedentemente creati file JSON di ricette di immagini e di configurazione dell'infrastruttura.

Per creare il file JSON per una ricetta di immagini, vedere. Crea una ricetta di immagini con AWS CLI Per creare il file JSON per una configurazione dell'infrastruttura, consulta. Creazione di una configurazione dell'infrastruttura

Prerequisiti per la distribuzione AMI tra account

Per garantire che gli account di destinazione possano avviare correttamente le istanze dall'immagine di Image Builder, è necessario configurare le autorizzazioni appropriate per tutti gli account di destinazione in tutte le regioni.

Se cripti la tua AMI usando AWS Key Management Service (AWS KMS), devi configurare un codice AWS KMS key per il tuo account che venga utilizzato per crittografare la nuova immagine.

Quando Image Builder esegue la distribuzione tra account crittografati AMIs, l'immagine nell'account di origine viene decrittografata e inviata alla regione di destinazione, dove viene nuovamente crittografata utilizzando la chiave designata per quella regione. Poiché Image Builder agisce per conto dell'account di destinazione e utilizza un ruolo IAM creato nell'area di destinazione, tale account deve avere accesso alle chiavi sia nella regione di origine che in quella di destinazione.

Chiavi di crittografia

I seguenti prerequisiti sono necessari se l'immagine viene crittografata utilizzando. AWS KMS I prerequisiti IAM sono descritti nella sezione successiva.

Requisiti dell'account di origine

- Crea una chiave KMS nel tuo account in tutte le regioni in cui crei e distribuisci la tua AMI. Puoi anche usare una chiave esistente.
- Aggiorna la politica delle chiavi per tutte queste chiavi per consentire agli account di destinazione di utilizzare la tua chiave.

Requisiti dell'account di destinazione

- Aggiungi una policy in linea `EC2ImageBuilderDistributionCrossAccountRole` che consenta al ruolo di eseguire le azioni richieste per distribuire un'AMI crittografata. Per i passaggi di configurazione di IAM, consulta la sezione sui Policy IAM prerequisiti.

Per ulteriori informazioni sull'utilizzo dell'accesso tra account diversi AWS KMS, consulta Consentire agli utenti di altri account di utilizzare una chiave KMS nella Guida per gli AWS Key Management Service sviluppatori.

Specificate la chiave di crittografia nella ricetta dell'immagine, come segue:

- Se utilizzi la console Image Builder, scegli la tua chiave di crittografia dall'elenco a discesa Encryption (alias KMS) nella sezione Storage (volumi) della ricetta.
- Se stai utilizzando l'azione CreateImageRecipe API o il create-image-recipe comando contenuto in AWS CLI, configura la chiave `blockDeviceMappings` nella `ebs` sezione sottostante dell'input JSON.

  Il seguente frammento di codice JSON mostra le impostazioni di crittografia per una ricetta di immagini. Oltre a fornire la chiave di crittografia, è necessario impostare il `encrypted` flag su. `true`

```
{
 ...
 "blockDeviceMappings": [
 {
  "deviceName": "Example root volume",
  "ebs": {
   "deleteOnTermination": true,
   "encrypted": true,
   "iops": 100,
   "kmsKeyId": "image-owner-key-id",
   ...
  },
  ...
 }],
 ...
}
```

Policy IAM

Per configurare le autorizzazioni di distribuzione tra account in AWS Identity and Access Management (IAM), segui questi passaggi:

1. Per utilizzare Image Builder AMIs distribuito su più account, il proprietario dell'account di destinazione deve creare un nuovo ruolo IAM nel proprio account chiamato. `EC2ImageBuilderDistributionCrossAccountRole`
2. Devono associarlo Policy Ec2ImageBuilderCrossAccountDistributionAccess al ruolo per consentire la distribuzione tra account. Per ulteriori informazioni sulle politiche gestite, vedere Managed Policies e Inline Policies nella Guida per l'AWS Identity and Access Management utente.
3. Verifica che l'ID dell'account di origine venga aggiunto alla policy di fiducia allegata al ruolo IAM dell'account di destinazione. L'esempio seguente mostra una politica di fiducia nell'account di destinazione che specifica l'ID dell'account dell'account di origine.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::444455556666:root"
        },
        "Action": "sts:AssumeRole"
    }]
}
```

   Per ulteriori informazioni sulle politiche di attendibilità, vedere Politiche basate sulle risorse nella Guida per l'utente.AWS Identity and Access Management
4. Se l'AMI che distribuisci è crittografato, il proprietario dell'account di destinazione deve aggiungere la seguente politica in linea al proprio account `EC2ImageBuilderDistributionCrossAccountRole` in modo da poter utilizzare le tue chiavi KMS. La `Principal` sezione contiene il loro numero di account. Ciò consente a Image Builder di agire per loro conto quando AWS KMS crittografa e decrittografa l'AMI con le chiavi appropriate per ciascuna regione.

```
{
 "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": "*"
  }
 ]
}
```

Per ulteriori informazioni sulle politiche in linea, vedere Inline Policies nella Guida per l'utente.AWS Identity and Access Management

5. Se utilizzi `launchTemplateConfigurations` per specificare un modello di EC2 lancio Amazon, devi anche aggiungere la seguente politica al tuo account `EC2ImageBuilderDistributionCrossAccountRole` in ogni destinazione.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateLaunchTemplateVersion",
                "ec2:ModifyLaunchTemplate"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/CreatedBy": "EC2 Image Builder"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeLaunchTemplates"
            ],
            "Resource": "*"
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*:*:launch-template/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/CreatedBy": "EC2 Image Builder"
                }
            }
        }
    ]
}
```

6. Se utilizzi un AWS Systems Manager parametro Parameter Store per memorizzare l'ID AMI dell'AMI di output per l'account di distribuzione e la regione, devi aggiungere la seguente politica al tuo account di destinazione `EC2ImageBuilderDistributionCrossAccountRole` in ogni account di destinazione.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:*:111122223333:parameter/ImageBuilder-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeImages"
            ],
            "Resource": "*"
        }
    ]
}
```

Limiti per la distribuzione tra account

Esistono alcune limitazioni nella distribuzione delle immagini di Image Builder tra gli account:

- L'account di destinazione è limitato a 50 copie AMI simultanee per ogni regione di destinazione.
- Se desideri copiare un'AMI di virtualizzazione paravirtuale (PV) in un'altra regione, la regione di destinazione deve supportare la virtualizzazione PV. AMIs Per ulteriori informazioni, consulta Tipi di virtualizzazione delle AMI Linux.
- Non è possibile creare una copia non crittografata di un'istantanea crittografata. Se non specifichi una chiave gestita dal cliente AWS Key Management Service (AWS KMS) per il `KmsKeyId` parametro, Image Builder utilizza la chiave predefinita per Amazon Elastic Block Store (Amazon EBS) Elastic Block Store (Amazon EBS). Per ulteriori informazioni, consulta Crittografia Amazon EBS nella Guida per l'utente di Amazon Elastic Compute Cloud.

Per ulteriori informazioni, consulta l'CreateDistributionConfigurationarticolo EC2 Image Builder API Reference.

Configurare la distribuzione tra account per un'AMI Image Builder dalla console

Questa sezione descrive come creare e configurare le impostazioni di distribuzione per la distribuzione tra account di Image AMIs Builder utilizzando. AWS Management Console La configurazione della distribuzione tra account richiede autorizzazioni IAM specifiche. È necessario completare la sezione Prerequisiti per la distribuzione AMI tra account per questa prima di continuare.

Per creare impostazioni di distribuzione nella console Image Builder, procedi nel seguente modo:

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Scegli le impostazioni di distribuzione dal pannello di navigazione. Questo mostra un elenco delle impostazioni di distribuzione create con il tuo account.
3. Nella parte superiore della pagina delle impostazioni di distribuzione, scegli Crea impostazioni di distribuzione. Verrà visualizzata la pagina Crea impostazioni di distribuzione.
4. Nella sezione Tipo di immagine, scegli Amazon Machine Image (AMI) come tipo di output. Si tratta dell'impostazione di default.
5. Nella sezione Generale, inserisci il nome della risorsa di impostazioni di distribuzione che desideri creare (obbligatorio).
6. Nella sezione Impostazioni della regione, inserisci un ID account a 12 cifre su cui vuoi distribuire l'AMI negli account Target per la regione selezionata e premi Invio. Questo verifica la

formattazione corretta, quindi visualizza l'ID dell'account che hai inserito sotto la casella. Ripeti la procedura per aggiungere altri account.

Per rimuovere un account che hai inserito, scegli la X visualizzata a destra dell'ID dell'account.

Inserisci il nome dell'AMI di uscita per ogni regione.

7. Continua a specificare le impostazioni aggiuntive di cui hai bisogno e scegli Crea impostazioni per creare la nuova risorsa per le impostazioni di distribuzione.

Configurare la distribuzione tra account per un'AMI Image Builder da AWS CLI

Questa sezione descrive come configurare un file di impostazioni di distribuzione e utilizzare il create-image comando in AWS CLI per creare e distribuire un'AMI Image Builder tra gli account.

La configurazione della distribuzione tra account richiede autorizzazioni IAM specifiche. È necessario completare la sezione Prerequisiti per la distribuzione AMI tra account per questa sezione prima di eseguire il comando. create-image

1. Configurare un file di impostazioni di distribuzione

   Prima di utilizzare il create-image comando contenuto in AWS CLI per creare un'AMI Image Builder distribuita su un altro account, è necessario creare una struttura `DistributionConfiguration` JSON che specifichi l'account IDs di destinazione nelle impostazioni. `AmiDistributionConfiguration` È necessario specificarne almeno uno `AmiDistributionConfiguration` nella regione di origine.

   Il seguente file di esempio, denominato`create-distribution-configuration.json`, mostra la configurazione per la distribuzione delle immagini tra account nella regione di origine.

```
{
 "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. Create le impostazioni di distribuzione

   Per creare una risorsa per le impostazioni di distribuzione di Image Builder utilizzando il create-distribution-configurationcomando in AWS CLI, fornire i seguenti parametri nel comando:

   - Immettete il nome della distribuzione nel `--name` parametro.
   - Allega il file JSON di configurazione della distribuzione che hai creato nel `--cli-input-json` parametro.

```
aws imagebuilder create-distribution-configuration --name my distribution name --
cli-input-json file://create-distribution-configuration.json
```

> **Note**
>
> - È necessario includere l'annotazione `file://` all'inizio del percorso del file JSON.
> - Il percorso del file JSON deve seguire la convenzione appropriata per il sistema operativo di base in cui si esegue il comando. Ad esempio, Windows utilizza la barra rovesciata (\) per fare riferimento al percorso della directory, mentre Linux e macOS utilizzano la barra (/).

È inoltre possibile fornire JSON direttamente nel comando, utilizzando il parametro. `--distributions`

Configura la distribuzione AMI con un modello di EC2 lancio

Per garantire un'esperienza di lancio coerente per l'AMI Image Builder negli account e nelle regioni di destinazione, puoi specificare un modello di EC2 lancio Amazon nelle impostazioni di distribuzione, utilizzando. `launchTemplateConfigurations` Quando `launchTemplateConfigurations` sono presenti durante il processo di distribuzione, Image Builder crea una nuova versione del modello di lancio che include tutte le impostazioni originali del modello e il nuovo ID AMI della build. Per ulteriori informazioni sull'avvio di un' EC2 istanza utilizzando un modello di avvio, consultate uno dei seguenti collegamenti, a seconda del sistema operativo di destinazione.

- Avvia un'istanza Linux da un modello di avvio
- Avvia un'istanza Windows da un modello di avvio

> **Note**
>
> Quando si include un modello di avvio per abilitare Windows Fast Launch nell'immagine, il modello di avvio deve includere il tag seguente in modo che Image Builder possa abilitare Windows Fast Launch per conto dell'utente.
> `CreatedBy: EC2 Image Builder`

Aggiungi un modello di EC2 avvio alle impostazioni di distribuzione AMI dalla console

Per fornire un modello di lancio con l'AMI di output, segui questi passaggi nella console:

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Scegliete le impostazioni di distribuzione dal pannello di navigazione. Questo mostra un elenco delle impostazioni di distribuzione create con il tuo account.
3. Nella parte superiore della pagina delle impostazioni di distribuzione, scegli Crea impostazioni di distribuzione. Si apre la pagina Crea impostazioni di distribuzione.
4. Nella sezione Tipo di immagine, scegli il tipo di output di Amazon Machine Image (AMI). Si tratta dell'impostazione di default.
5. Nella sezione Generale, inserisci il nome della risorsa di impostazioni di distribuzione che desideri creare (obbligatorio).
6. Nella sezione Impostazioni regionali, seleziona il nome di un modello di EC2 lancio dall'elenco. Se non ci sono modelli di lancio nel tuo account, scegli Crea nuovo modello di lancio, che apre i modelli di lancio nella EC2dashboard.

   Seleziona la casella di controllo Imposta la versione predefinita per aggiornare la versione predefinita del modello di avvio alla nuova versione creata da Image Builder con l'AMI di output.

   Per aggiungere un altro modello di avvio alla regione selezionata, scegli Aggiungi configurazione del modello di avvio.

   Per rimuovere un modello di lancio, scegli Rimuovi.

7. Continua a specificare tutte le impostazioni aggiuntive di cui hai bisogno e scegli Crea impostazioni per creare la nuova risorsa per le impostazioni di distribuzione.

Aggiungi un modello di EC2 avvio alle impostazioni di distribuzione AMI dal AWS CLI

Questa sezione descrive come configurare un file di impostazioni di distribuzione con un modello di avvio e utilizzare il create-image comando in AWS CLI per creare e distribuire un'AMI Image Builder e una nuova versione del modello di avvio che la utilizza.

1. Configurare un file delle impostazioni di distribuzione

   Prima di poter creare un'AMI Image Builder con un modello di avvio, utilizzando AWS CLI, è necessario creare una struttura JSON di configurazione della distribuzione che specifichi le impostazioni. `launchTemplateConfigurations` È necessario specificare almeno una `launchTemplateConfigurations` voce nella regione di origine.

   Il seguente file di esempio, denominato`create-distribution-config-launch-template.json`, mostra alcuni possibili scenari per la configurazione del modello di avvio nella regione di origine.

```
{
    "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. Create le impostazioni di distribuzione

   Per creare una risorsa per le impostazioni di distribuzione di Image Builder utilizzando il create-distribution-configurationcomando in AWS CLI, fornire i seguenti parametri nel comando:

   - Immettete il nome della distribuzione nel `--name` parametro.
   - Allega il file JSON di configurazione della distribuzione che hai creato nel `--cli-input-json` parametro.

```
aws imagebuilder create-distribution-configuration --name my distribution name--
cli-input-json file://create-distribution-config-launch-template.json
```

> **Note**
>
> - È necessario includere l'annotazione `file://` all'inizio del percorso del file JSON.
> - Il percorso del file JSON deve seguire la convenzione appropriata per il sistema operativo di base in cui si esegue il comando. Ad esempio, Windows utilizza la barra rovesciata (\) per fare riferimento al percorso della directory, mentre Linux e macOS utilizzano la barra (/).

È inoltre possibile fornire JSON direttamente nel comando, utilizzando il parametro. `--distributions`

Condividi le risorse di Image Builder con AWS RAM

EC2 Image Builder si integra con AWS Resource Access Manager (AWS RAM) in modo da poter condividere i seguenti tipi di risorse di Image Builder con chiunque o tramite. Account AWS AWS Organizations

- Componenti
- Immagini
- Ricette

Per condividere risorse tramite AWS RAM, è necessario creare una condivisione di risorse. Una condivisione di risorse specifica le risorse da condividere e i consumatori con cui condividerle. I consumatori possono essere singoli individui Account AWS, unità organizzative o un'intera organizzazione. AWS Organizations L'elenco seguente include i tipi di account e organizzazioni con cui è possibile condividere.

- Specifico Account AWS all'interno o all'esterno della sua organizzazione in AWS Organizations.
- Un'unità organizzativa (OU) all'interno della sua organizzazione in AWS Organizations.
- L'intera organizzazione in AWS Organizations.
- AWS Organizations o OUs all'esterno della sua organizzazione.

In questo modello, il Account AWS proprietario della risorsa (proprietario) la condivide con altri Account AWS o tramite AWS Organizations (consumatori) all'interno della stessa regione. Quando una risorsa condivisa viene aggiornata, i consumatori ricevono tali aggiornamenti automaticamente.

> **Note**
>
> I componenti, le immagini e le ricette di immagini condivisi vengono conteggiati solo per i limiti di risorse corrispondenti del proprietario. I limiti di risorse dei consumatori non sono influenzati dalle risorse condivise con loro.

Argomenti

- Proprietari delle risorse
- Consumatori di risorse

- Crea una condivisione di AWS RAM risorse per le tue risorse di Image Builder
- Annullare la condivisione di una risorsa Image Builder da AWS RAM

Proprietari delle risorse

Le risorse di Image Builder possono essere condivise solo nel luogo in Regione AWS cui sono state create. Quando condividi queste risorse, non verranno replicate tra le regioni.

Per ottenere un elenco delle risorse di Image Builder che possiedi e che puoi condividere, specifica il filtro di proprietà nella console o quando esegui il comando in. AWS CLI

- Elenca i componenti di Image Builder
- Elenca immagini
- Elenca e visualizza i dettagli della ricetta tramite immagine
- Elenca e visualizza i dettagli della ricetta del contenitore

Per ulteriori informazioni su AWS RAM, consulta la Guida per l'AWS RAM utente.

Prerequisiti per la condivisione delle risorse di Image Builder

Per condividere una risorsa Image Builder, ad esempio un componente, un'immagine o una ricetta:

- Account AWS È necessario possedere la risorsa Image Builder che si desidera condividere. Non puoi condividere risorse che sono state condivise con te.
- La chiave AWS Key Management Service (AWS KMS) associata alle risorse crittografate deve essere condivisa in modo esplicito con gli account, le organizzazioni o OUs.
- Per condividere AWS Organizations e OUs utilizzare le risorse di Image Builder AWS RAM, è necessario abilitare la condivisione. Per ulteriori informazioni, consulta Abilita la condivisione con AWS Organizations nella Guida per l'utente di AWS RAM .
- Se distribuisci un'immagine crittografata AWS KMS tra account in diverse regioni, devi creare una chiave KMS e un alias in ciascuna regione di destinazione. Inoltre, le persone che avvieranno le istanze in quelle regioni dovranno accedere alla chiave KMS specificata tramite la Key Policy.

Le seguenti risorse che Image Builder crea dalla build della pipeline non sono considerate risorse Image Builder, ma sono risorse esterne che Image Builder distribuisce nel tuo account e agli account

e alle organizzazioni o unità organizzative (OUs) specificate nella configurazione di distribuzione. Regioni AWS

- Immagini di macchine Amazon (AMIs)
- Immagini di container che risiedono in Amazon ECR

Per ulteriori informazioni sulle impostazioni di distribuzione per l'AMI, consultaCrea e aggiorna le configurazioni di distribuzione AMI. Per ulteriori informazioni sulle impostazioni di distribuzione per l'immagine del contenitore in Amazon ECR, consultaCrea e aggiorna le impostazioni di distribuzione per le immagini dei contenitori.

Per ulteriori informazioni sulla condivisione dell'AMI con AWS Organizations e OUs, consulta Condividere un AMI con organizzazioni o OUs.

Consumatori di risorse

I consumatori possono utilizzare una risorsa condivisa, ma non possono modificarla in alcun modo. Quando creano ricette di Image Builder, possono specificare un'immagine condivisa come immagine di base e aggiungere componenti condivisi. Possono anche specificare una ricetta condivisa quando creano una pipeline di immagini Image Builder o quando utilizzano il create-image comando in. AWS CLI

Se si appartiene a un'organizzazione in AWS Organizations e la condivisione all'interno dell'organizzazione è abilitata, ai consumatori dell'organizzazione viene automaticamente concesso l'accesso alla risorsa condivisa. In caso contrario, i consumatori ricevono un invito a partecipare alla condivisione di risorse e ottengono l'accesso alla risorsa condivisa dopo aver accettato l'invito.

Crea una condivisione di AWS RAM risorse per le tue risorse di Image Builder

Per condividere un componente, un'immagine o una ricetta di Image Builder, è necessario aggiungerlo a una condivisione di AWS Resource Access Manager risorse. La condivisione di risorse specifica le risorse da condividere e i consumatori con cui vengono condivise.

Le seguenti opzioni sono disponibili per la condivisione delle risorse.

Opzione 1: creare una condivisione di risorse RAM

Quando crei una condivisione di risorse RAM, puoi condividere un componente, un'immagine o una ricetta di tua proprietà in un unico passaggio. Utilizza uno dei seguenti metodi per creare la tua condivisione di risorse:

- Console

  Per creare la condivisione di risorse utilizzando la AWS RAM console, consulta Condividere AWS le risorse di tua proprietà nella Guida per l'AWS RAM utente.
- AWS CLI

  Per creare la condivisione di risorse utilizzando l'interfaccia a AWS RAM riga di comando, esegui il create-resource-share comando in AWS CLI.

Opzione 2: applicare una politica delle risorse e passare a una condivisione di risorse esistente

La seconda opzione per condividere le risorse prevede due passaggi, ovvero l'esecuzione dei comandi in entrambi AWS CLI i casi. Il primo passaggio utilizza i comandi Image Builder AWS CLI per applicare politiche basate sulle risorse alla risorsa condivisa. Il secondo passaggio promuove la risorsa in una condivisione di risorse RAM utilizzando il promote-resource-share-created-from-policy AWS RAM comando in AWS CLI per garantire che la risorsa sia visibile a tutti i principali con cui l'hai condivisa.

1. Applica la politica delle risorse

   Per applicare correttamente la politica delle risorse, devi assicurarti che l'account con cui stai condividendo sia autorizzato ad accedere a tutte le risorse sottostanti.

   Scegliete la scheda corrispondente al tipo di risorsa per il comando applicabile.

   Image

   È possibile applicare una politica delle risorse a un'immagine per consentire ad altri di utilizzarla come immagine di base nelle proprie ricette.

   Esegui il comando put-image-policy Image Builder in AWS CLI, per identificare AWS i principali con cui condividere l'immagine.

```
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

È possibile applicare una politica delle risorse a un componente di compilazione o test per abilitare la condivisione tra account. Questo comando concede agli altri account il permesso di utilizzare il componente nelle loro ricette. Per applicare correttamente la politica delle risorse, è necessario assicurarsi che l'account con cui si sta condividendo disponga dell'autorizzazione ad accedere a tutte le risorse a cui fa riferimento il componente condiviso, ad esempio i file ospitati in archivi privati.

Esegui il comando put-component-policy Image Builder in AWS CLI, per identificare AWS i principali con cui condividere il componente.

```
aws imagebuilder put-component-policy --component-arn arn:aws:imagebuilder:us-
west-2:123456789012:component/my-example-component/2019.12.03/1 --policy
 '{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal":
 { "AWS": [ "123456789012" ] }, "Action": [ "imagebuilder:GetComponent",
 "imagebuilder:ListComponents" ], "Resource": [ "arn:aws:imagebuilder:us-
west-2:123456789012:component/my-example-component/2019.12.03/1" ] } ] }'
```

Image recipe

È possibile applicare una politica delle risorse a una ricetta di immagini per abilitare la condivisione tra account. Questo comando autorizza gli altri account a utilizzare la tua ricetta per creare immagini nei loro account. Per applicare correttamente la politica delle risorse, devi assicurarti che l'account con cui stai condividendo sia autorizzato ad accedere a tutte le risorse a cui fa riferimento la ricetta, come l'immagine di base o i componenti selezionati.

Esegui il comando put-image-recipe-policy Image Builder in AWS CLI, per identificare AWS i principali con cui condividere l'immagine.

```
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

È possibile applicare una politica delle risorse a una ricetta contenitore per abilitare la condivisione tra account. Questo comando autorizza gli altri account a utilizzare la tua ricetta per creare immagini nei loro account. Per applicare correttamente la politica delle risorse, devi assicurarti che l'account con cui stai condividendo sia autorizzato ad accedere a tutte le risorse a cui fa riferimento la ricetta, come l'immagine di base o i componenti selezionati.

Esegui il comando put-container-recipe-policy Image Builder in AWS CLI, per identificare AWS i principali con cui condividere l'immagine.

```
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**
>
> Per impostare le politiche corrette per condividere e annullare la condivisione di una risorsa, il proprietario della risorsa deve disporre delle autorizzazioni. `imagebuilder:put*`

2. Promuovi come condivisione di risorse RAM

Per assicurarti che la risorsa sia visibile a tutti i principali destinatari con cui l'hai condivisa, esegui il promote-resource-share-created-from-policy AWS RAM comando in. AWS CLI

Annullare la condivisione di una risorsa Image Builder da AWS RAM

Per annullare la condivisione di una risorsa Image Builder di tua proprietà, come un componente, un'immagine o una ricetta condivisi, devi AWS Resource Access Manager rimuoverla dalla condivisione di risorse. È possibile effettuare tale operazione mediante la console AWS RAM o AWS CLI.

> **Note**
>
> I proprietari non possono eliminare una risorsa condivisa finché non è più condivisa. Un proprietario non può annullare la condivisione di queste risorse finché nessuno dei consumatori non dipende da esse.

Per annullare la condivisione di un componente, un'immagine o una ricetta condivisi di tua proprietà utilizzando la console AWS Resource Access Manager

Consulta Aggiornamento di una condivisione di risorse in Guida per l'utente di AWS RAM .

Per annullare la condivisione di un componente, un'immagine o una ricetta condivisi di tua proprietà utilizzando il AWS CLI

Utilizzate il disassociate-resource-share comando per interrompere la condivisione della risorsa.

Risorse di output per Tag Image Builder

L'etichettatura delle risorse può essere utile per filtrare e tenere traccia dei costi delle risorse o di altre categorie. Puoi anche controllare l'accesso in base ai tag. Per ulteriori informazioni sull'autorizzazione basata su tag, consulta Autorizzazione basata sui tag Image Builder

Image Builder supporta i seguenti tag dinamici:

- `- {{imagebuilder:buildDate}}`

  Risolve alla compilazione in date/time fase di compilazione.
- `- {{imagebuilder:buildVersion}}`

  Risolve in una versione build, che è un numero che si trova alla fine di un Image Builder Amazon Resource Name (ARN). Ad esempio, `"arn:aws:imagebuilder:us-west-2:123456789012:component/myexample-component/2019.12.02/1"` mostra la versione build come. 1

Per aiutarti a tenere traccia delle Amazon Machine Images (AMIs) che hai distribuito, Image Builder aggiunge automaticamente i seguenti tag all'output. AMIs

- `"CreatedBy":"EC2 Image Builder"`
- `"Ec2ImageBuilderArn":"arn:aws:imagebuilder:`*`us-west-2`*`:`*`123456789012`*`:image/`*`simple-recipe-linux`*`/`*`1.0.0/10`*`"`. Questo tag contiene l'ARN della risorsa immagine Image Builder utilizzata per creare l'AMI.

Indice

- Assegna un tag a una risorsa dal AWS CLI
- Togliere il tag a una risorsa da AWS CLI
- Elenca tutti i tag per una risorsa specifica dal AWS CLI

Assegna un tag a una risorsa dal AWS CLI

L'esempio seguente mostra come utilizzare un comando imagebuilder CLI per aggiungere e contrassegnare una risorsa in Image EC2 Builder. È necessario fornire i tag `resourceArn` e i tag da applicare.

I `tag-resource.json` contenuti di esempio sono i seguenti:

```
{
    "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/my-example-pipeline",
    "tags": {
        "KeyName": "KeyValue"
    }
}
```

Eseguite il comando seguente, che fa riferimento al `tag-resource.json` file precedente.

```
aws imagebuilder tag-resource --cli-input-json file://tag-resource.json
```

Togliere il tag a una risorsa da AWS CLI

L'esempio seguente mostra come utilizzare un comando imagebuilder CLI per rimuovere un tag da una risorsa. È necessario fornire le chiavi `resourceArn` e le chiavi per rimuovere il tag.

I `untag-resource.json` contenuti dell'esempio sono i seguenti:

```
{
    "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/my-example-pipeline",
    "tagKeys": [
        "KeyName"
    ]
}
```

Eseguite il comando seguente, che fa riferimento al `untag-resource.json` file precedente.

```
aws imagebuilder untag-resource --cli-input-json file://untag-resource.json
```

Elenca tutti i tag per una risorsa specifica dal AWS CLI

L'esempio seguente mostra come utilizzare un comando imagebuilder CLI per elencare tutti i tag per una risorsa specifica.

```
aws imagebuilder list-tags-for-resource --resource-arn arn:aws:imagebuilder:us-
west-2:123456789012:image-pipeline/my-example-pipeline
```

Eliminare risorse Image Builder obsolete o inutilizzate

Il tuo ambiente Image Builder, proprio come la tua abitazione, necessita di una manutenzione regolare per aiutarti a trovare ciò di cui hai bisogno e a completare le tue attività senza sprecare terreno. Assicurati di pulire regolarmente le risorse temporanee che hai creato per i test. Altrimenti, potresti dimenticarti di quelle risorse e poi non ricordare per cosa sono state utilizzate. A quel punto, potrebbe non essere chiaro se è possibile liberarsene in sicurezza.

L'eliminazione delle risorse non elimina alcuna immagine di container Amazon EC2 AMIs o Amazon ECR creata durante il processo di creazione dell'immagine. È necessario pulirli separatamente, utilizzando le azioni o le API EC2 o i AWS CLI comandi della console Amazon o Amazon ECR appropriati.

> **Tip**
>
> Per evitare errori di dipendenza quando elimini risorse, assicurati di eliminare le risorse nell'ordine seguente:
>
> 1. Pipeline di immagini
> 2. Ricetta di immagini
> 3. Tutte le risorse rimanenti

Eliminare risorse da AWS Management Console

Per eliminare una pipeline di immagini e le relative risorse, procedi nel seguente modo:

Eliminare la pipeline

1. Per visualizzare un elenco delle pipeline di compilazione create con il tuo account, scegli Image pipelines dal pannello di navigazione.
2. Seleziona la casella di controllo accanto a Pipeline name per selezionare la pipeline che desideri eliminare.
3. Nella parte superiore del pannello Image pipelines, nel menu Azioni, scegliete Elimina.
4. Per confermare l'eliminazione, inserite `Delete` nella casella e scegliete Elimina.

Elimina la ricetta

1. Per visualizzare un elenco delle ricette create con il tuo account, scegli Ricette con immagini dal pannello di navigazione.
2. Seleziona la casella di controllo accanto a Nome della ricetta per selezionare la ricetta che desideri eliminare.
3. Nella parte superiore del pannello Ricette con immagini, nel menu Azioni, scegli Elimina ricetta.
4. Per confermare l'eliminazione, entra `Delete` nella casella e scegli Elimina.

Elimina la configurazione dell'infrastruttura

1. Per visualizzare un elenco delle configurazioni dell'infrastruttura create con il tuo account, scegli Configurazione dell'infrastruttura dal pannello di navigazione.
2. Seleziona la casella di controllo accanto a Nome della configurazione per selezionare la configurazione dell'infrastruttura che desideri eliminare.
3. Nella parte superiore del pannello Configurazioni dell'infrastruttura, scegli Elimina.
4. Per confermare l'eliminazione, entra `Delete` nella casella e scegli Elimina.

Eliminare le impostazioni di distribuzione

1. Per visualizzare un elenco delle impostazioni di distribuzione create nel tuo account, scegli Impostazioni di distribuzione dal pannello di navigazione.
2. Seleziona la casella di controllo accanto a Nome di configurazione per selezionare le impostazioni di distribuzione che hai creato per questo tutorial.
3. Nella parte superiore del pannello delle impostazioni di distribuzione, scegli Elimina.
4. Per confermare l'eliminazione, entra `Delete` nella casella e scegli Elimina.

Eliminazione di un'immagine

1. Per visualizzare un elenco delle immagini create con il tuo account, scegli Immagini dal pannello di navigazione.
2. Scegli la versione dell'immagine che desideri rimuovere. Si apre la pagina delle versioni di compilazione dell'immagine.
3. Seleziona la casella di controllo accanto alla Versione per qualsiasi immagine che desideri eliminare. È possibile selezionare più di una versione dell'immagine alla volta.

4. Nella parte superiore del pannello Image build versions, scegli Elimina versione.
5. Per confermare l'eliminazione, entra `Delete` nella casella e scegli Elimina.

Eliminate una pipeline di immagini dal AWS CLI

Gli esempi seguenti mostrano come eliminare le risorse di Image Builder utilizzando. AWS CLI Come accennato in precedenza, le risorse devono essere eliminate nel seguente ordine per evitare errori di dipendenza:

1. Pipeline di immagini
2. Ricetta di immagini
3. Tutte le risorse rimanenti

Eliminare la pipeline di immagini da AWS CLI

L'esempio seguente mostra come eliminare una pipeline di immagini specificandone l'ARN.

```
aws imagebuilder delete-image-pipeline --image-pipeline-arn arn:aws:imagebuilder:us-
west-2:123456789012:image-pipeline/my-example-pipeline
```

Eliminare la ricetta dell'immagine dal AWS CLI

L'esempio seguente mostra come eliminare una ricetta di immagine specificandone l'ARN.

```
aws imagebuilder delete-image-recipe --image-recipe-arn arn:aws:imagebuilder:us-
west-2:123456789012:image-recipe/my-example-recipe/2019.12.03
```

Elimina una configurazione dell'infrastruttura

L'esempio seguente mostra come eliminare una risorsa di configurazione dell'infrastruttura specificandone l'ARN.

```
aws imagebuilder delete-infrastructure-configuration --infrastructure-configuration-arn
 arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/my-example-
infrastructure-configuration
```

Eliminare le impostazioni di distribuzione

L'esempio seguente mostra come eliminare una risorsa delle impostazioni di distribuzione specificandone l'ARN.

```
aws imagebuilder delete-distribution-configuration --distribution-configuration-arn
 arn:aws:imagebuilder:us-west-2:123456789012:distribution-configuration/my-example-
distribution-configuration
```

Eliminazione di un'immagine

L'esempio seguente mostra come eliminare una versione di build dell'immagine specificandone l'ARN.

```
aws imagebuilder delete-image --image-build-version-arn arn:aws:imagebuilder:us-
west-2:123456789012:image/my-example-image/2019.12.02/1
```

Elimina un componente

L'esempio seguente mostra come utilizzare un comando imagebuilder CLI per eliminare una versione di build del componente specificandone l'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**
>
> Assicurati che non ci siano ricette che facciano riferimento in alcun modo alla versione di build del componente prima di eliminarlo. In caso contrario, potrebbero verificarsi errori nella pipeline.

Gestisci i flussi di lavoro di creazione e test per le immagini di Image Builder

Un flusso di lavoro di immagini definisce la sequenza di passaggi che EC2 Image Builder esegue durante le fasi di creazione e test del processo di creazione dell'immagine. Questo fa parte del framework generale del flusso di lavoro di Image Builder.

Vantaggi del flusso di lavoro relativo

- I flussi di lavoro basati sulle immagini offrono maggiore flessibilità, visibilità e controllo sul processo di creazione delle immagini.
- È possibile aggiungere passaggi di flusso di lavoro personalizzati quando si definisce il documento del flusso di lavoro oppure è possibile scegliere di utilizzare il flusso di lavoro predefinito di Image Builder.
- È possibile escludere le fasi del flusso di lavoro incluse nei flussi di lavoro con immagini predefiniti.
- È possibile creare flussi di lavoro di solo test che ignorano completamente il processo di creazione. Puoi fare lo stesso per creare flussi di lavoro di sola compilazione.

> **Note**
>
> Non puoi modificare un flusso di lavoro esistente, ma puoi clonarlo o creare una nuova versione.

Argomenti sul workflow

- Framework del flusso di lavoro: fasi
- Accesso al servizio
- Usa flussi di lavoro gestiti per le tue immagini
- Elenca i flussi di lavoro con immagini
- Crea un flusso di lavoro basato su immagini
- Creare un documento di workflow YAML

Framework del flusso di lavoro: fasi

Per personalizzare i flussi di lavoro relativi alle immagini, è importante comprendere le fasi del flusso di lavoro che costituiscono il framework del flusso di lavoro per la creazione di immagini.

Il framework del flusso di lavoro per la creazione di immagini include le seguenti fasi distinte.

1. Fase di creazione (pre-snapshot): durante la fase di creazione, apporti modifiche all'istanza di EC2 build di Amazon che esegue la tua immagine di base, per creare la linea di base per la tua nuova immagine. Ad esempio, la ricetta può includere componenti che installano un'applicazione o modificano le impostazioni del firewall del sistema operativo.

   Una volta completata con successo questa fase, Image Builder crea un'istantanea o un'immagine contenitore che utilizza per la fase di test e oltre.
2. Fase di test (post-snapshot): durante la fase di test, ci sono alcune differenze tra le immagini di creazione e le immagini del contenitore. AMIs Per i flussi di lavoro AMI, Image Builder avvia EC2 un'istanza dall'istantanea creata come fase finale della fase di creazione. I test vengono eseguiti sulla nuova istanza per convalidare le impostazioni e garantire che l'istanza funzioni come previsto. Per i flussi di lavoro in container, i test vengono eseguiti sulla stessa istanza utilizzata per la creazione.

Il framework del flusso di lavoro include anche una fase di distribuzione. Tuttavia, Image Builder gestisce i flussi di lavoro per quella fase.

Accesso al servizio

Per eseguire flussi di lavoro di immagini, Image Builder necessita dell'autorizzazione per eseguire azioni del flusso di lavoro. È possibile specificare il ruolo AWSServiceRoleForImageBuildercollegato al servizio oppure specificare il proprio ruolo personalizzato per l'accesso al servizio, come segue.

- Console: nella procedura guidata alla pipeline, fase 3, definisci il processo di creazione dell'immagine, seleziona il ruolo collegato al servizio o il tuo ruolo personalizzato dall'elenco dei ruoli IAM nel pannello di accesso al servizio.
- API Image Builder: nella richiesta di CreateImageazione, specifica il ruolo collegato al servizio o il tuo ruolo personalizzato come valore per il parametro. `executionRole`

Per ulteriori informazioni su come creare un ruolo di servizio, consulta Creazione di un ruolo per delegare le autorizzazioni a un AWS servizio nella Guida per l'utente.AWS Identity and Access Management

Usa flussi di lavoro gestiti per le tue immagini

I flussi di lavoro gestiti vengono creati e gestiti da. AWS Quando utilizzi flussi di lavoro gestiti nelle tue pipeline di immagini o per la creazione di immagini una tantum, puoi selezionare l'Amazon Resource Name (ARN) del flusso di lavoro gestito che desideri utilizzare. Amazon fornisce le versioni più recenti a cui sono state applicate patch e altri aggiornamenti. Per ottenere un elenco di flussi di lavoro gestitiElenca i flussi di lavoro con immagini, consulta e filtra su Owner = Amazon (console).

Elenca i flussi di lavoro con immagini

Nella pagina di elenco dei flussi di lavoro di immagini nella console Image Builder, puoi ottenere un elenco delle risorse del flusso di lavoro delle immagini che possiedi o a cui hai accesso, insieme ad alcuni dettagli chiave su queste risorse. Puoi anche utilizzare comandi o azioni con l'API Image Builder o AWS CLI elencare i SDKs flussi di lavoro di immagini nel tuo account.

Puoi utilizzare uno dei seguenti metodi per elencare le risorse del flusso di lavoro basato sulle immagini che possiedi o a cui hai accesso. Per l'azione API, vedere ListWorkflowsnel riferimento all'API EC2 Image Builder. Per la richiesta SDK associata, fai riferimento al link Vedi anche nella stessa pagina.

Puoi filtrare i seguenti dettagli per semplificare l'elenco dei risultati. Ad esempio, se filtri Owner = Amazon nella console, l'elenco mostra solo i flussi di lavoro AWS gestiti.

- Flusso di lavoro
- Versione
- Tipo
- Owner

Console

Dettagli del flusso di lavoro

I dettagli sulla pagina con l'elenco dei flussi di lavoro di immagini nella console Image Builder includono quanto segue:

- Workflow: il nome della versione più recente della risorsa Image Workflow. Nella console Image Builder, la colonna Workflow si collega alla pagina dei dettagli del flusso di lavoro.
- Versione: la versione più recente della risorsa Image Workflow.
- Tipo: il tipo di flusso di lavoro: `BUILD` o`TEST`.
- Proprietario: il proprietario della risorsa del flusso di lavoro.
- Ora di creazione: la data e l'ora in cui Image Builder ha creato la versione più recente della risorsa Image Workflow.
- ARN: Amazon Resource Name (ARN) della versione corrente della risorsa Image Workflow.

Elenca i flussi di lavoro con immagini

Per elencare le risorse del flusso di lavoro delle immagini nella console Image Builder, effettuate le seguenti operazioni:

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Scegliete Image Workflows dal pannello di navigazione.

Filtra i risultati

Nella pagina con l'elenco dei flussi di lavoro con immagini, puoi cercare flussi di lavoro di immagini specifici per filtrare i risultati. I seguenti filtri sono disponibili per i flussi di lavoro relativi alle immagini:

`Workflow`

Puoi inserire tutto o parte del nome di un flusso di lavoro per semplificare i risultati. L'impostazione predefinita prevede la visualizzazione di tutti i flussi di lavoro nell'elenco.

`Version`

È possibile inserire tutto o parte di un numero di versione per semplificare i risultati. L'impostazione predefinita prevede la visualizzazione di tutte le versioni nell'elenco.

`Type`

È possibile filtrare in base al tipo di flusso di lavoro o visualizzare tutti i tipi. L'impostazione predefinita prevede la visualizzazione di tutti i tipi di flusso di lavoro nell'elenco.

- BUILD
- TEST

`Owner`

Quando selezioni il filtro proprietario dalla barra di ricerca, Image Builder mostra un elenco dei proprietari dei flussi di lavoro di immagini nel tuo account. È possibile selezionare un proprietario dall'elenco per semplificare i risultati. L'impostazione predefinita prevede la visualizzazione di tutti i proprietari nell'elenco.

- Account AWS— L'account proprietario della risorsa del flusso di lavoro.
- Amazon: risorse per il flusso di lavoro che Amazon possiede e gestisce.

AWS CLI

Quando esegui il list-workflows comando in AWS CLI, puoi ottenere un elenco di flussi di lavoro di immagini di cui sei proprietario o a cui hai accesso.

L'esempio di comando seguente mostra come utilizzare il list-workflows comando senza filtri per elencare tutte le risorse del flusso di lavoro di immagini di Image Builder di cui si è proprietari o a cui si ha accesso.

Esempio: elenca tutti i flussi di lavoro relativi alle immagini

```
aws imagebuilder list-workflows
```

Output:

```
{
 "workflowVersionList": [
  {
    "name": "example-test-workflow",
    "dateCreated": "2023-11-21T22:53:14.347Z",
    "version": "1.0.0",
    "owner": "111122223333",
    "type": "TEST",
    "arn": "arn:aws:imagebuilder:us-west-2:111122223333:workflow/test/example-test-
workflow/1.0.0"
  },
```

```
  {
    "name": "example-build-workflow",
    "dateCreated": "2023-11-20T12:26:10.425Z",
    "version": "1.0.0",
    "owner": "111122223333",
    "type": "BUILD",
    "arn": "arn:aws:imagebuilder:us-west-2:111122223333:workflow/build/example-
build-workflow/1.0.0"
  }
 ]
}
```

Quando si esegue il list-workflows comando, è possibile applicare filtri per semplificare i risultati, come illustrato nell'esempio seguente. Per ulteriori informazioni su come filtrare i risultati, vedete il comando list-workflows nel Command Reference.AWS CLI

Esempio: filtro per creare flussi di lavoro

```
aws imagebuilder list-workflows --filters name="type",values="BUILD"
```

Output:

```
{
 "workflowVersionList": [
  {
    "name": "example-build-workflow",
    "dateCreated": "2023-11-20T12:26:10.425Z",
    "version": "1.0.0",
    "owner": "111122223333",
    "type": "BUILD",
    "arn": "arn:aws:imagebuilder:us-west-2:111122223333:workflow/build/example-
build-workflow/1.0.0"
  }
 ]
}
```

Crea un flusso di lavoro basato su immagini

Quando crei un flusso di lavoro di immagini, hai un maggiore controllo sul processo di creazione delle immagini. È possibile specificare quale flusso di lavoro viene eseguito quando Image Builder crea

l'immagine e quali flussi di lavoro vengono eseguiti quando verifica l'immagine. È inoltre possibile specificare una chiave gestita dal cliente per crittografare le risorse del flusso di lavoro. Per ulteriori informazioni sulla crittografia per le risorse del flusso di lavoro, consultaCrittografia e gestione delle chiavi in Image Builder.

Per la creazione di immagini, puoi specificare un flusso di lavoro in fase di creazione e uno o più flussi di lavoro in fase di test. Puoi persino saltare completamente la fase di compilazione o test, a seconda delle tue esigenze. Le azioni intraprese dal flusso di lavoro vengono configurate nel documento di definizione YAML utilizzato dal flusso di lavoro. Per ulteriori informazioni sulla sintassi del documento YAML, consulta. Creare un documento di workflow YAML

Per i passaggi per creare un nuovo flusso di lavoro di compilazione o test, seleziona la scheda corrispondente all'ambiente che utilizzerai.

AWS Management Console

È possibile utilizzare il seguente processo per creare un nuovo flusso di lavoro nella console Image Builder.

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Scegliete Image Workflows dal pannello di navigazione. Viene visualizzato un elenco di flussi di lavoro basati su immagini di cui il tuo account è proprietario o a cui ha accesso.

   > **Note**
   >
   > Nel tuo elenco vedrai sempre le risorse del flusso di lavoro gestito da Amazon che Image Builder utilizza per i flussi di lavoro predefiniti. Per visualizzare i dettagli di questi flussi di lavoro, puoi selezionare il link Workflow.

3. Per creare un nuovo flusso di lavoro, scegli Crea flusso di lavoro con immagini. Viene visualizzata la pagina Crea flusso di lavoro con immagini.
4. Configura i dettagli per il tuo nuovo flusso di lavoro. Per creare un flusso di lavoro di creazione, seleziona l'opzione Crea nella parte superiore del modulo. Per creare un flusso di lavoro di test, seleziona l'opzione Test nella parte superiore del modulo. Image Builder compila l'elenco dei modelli in base a questa opzione. Tutti gli altri passaggi sono gli stessi per i flussi di lavoro di compilazione e test.

   Generali

La sezione generale include le impostazioni che si applicano alla risorsa del flusso di lavoro, come il nome e la descrizione. Le impostazioni generali includono quanto segue:

- Nome del flusso di lavoro delle immagini (obbligatorio): il nome del flusso di lavoro delle immagini. Il nome deve essere univoco nel tuo account. Il nome può avere una lunghezza massima di 128 caratteri. I caratteri validi includono lettere, numeri-, spazi e_.
- Versione (obbligatoria): la versione semantica della risorsa del flusso di lavoro da creare (major.minor.patch).
- Descrizione (opzionale): aggiungi facoltativamente una descrizione per il tuo flusso di lavoro.
- Chiave KMS (opzionale): puoi crittografare le risorse del flusso di lavoro con una chiave gestita dal cliente. Per ulteriori informazioni, consulta Crittografa i flussi di lavoro relativi alle immagini con una chiave gestita dal cliente.

Documento di definizione

Il documento del flusso di lavoro YAML contiene tutte le configurazioni per il flusso di lavoro.

Inizia a usare

- Per iniziare con un modello predefinito di Image Builder come base per il flusso di lavoro, selezionate l'opzione Inizia dai modelli. Questa opzione è selezionata per impostazione predefinita. Dopo aver scelto il modello da utilizzare dall'elenco Modelli, la configurazione predefinita dal modello selezionato viene copiata nel Contenuto del documento del nuovo flusso di lavoro, dove è possibile apportare modifiche.
- Per definire il documento del flusso di lavoro da zero, selezionate l'opzione Inizia da zero. Questo compila il Contenuto con una breve descrizione di alcune parti importanti del formato del documento per aiutarti a iniziare.

Il pannello Contenuto include una barra di stato nella parte inferiore che mostra gli avvisi o gli errori relativi al documento YAML. Per ulteriori informazioni su come creare un documento di workflow YAML, consulta. Creare un documento di workflow YAML

5. Una volta completato il flusso di lavoro o se desideri salvare lo stato di avanzamento e riprenderlo in un secondo momento, scegli Crea flusso di lavoro.

AWS CLI

Prima di eseguire il create-workflow comando in AWS CLI, devi creare il documento YAML che contiene tutta la configurazione per il tuo flusso di lavoro. Per ulteriori informazioni, consulta Creare un documento di workflow YAML.

L'esempio seguente mostra come creare un flusso di lavoro di compilazione con il comando AWS CLI create-workflow. Il `--data` parametro si riferisce a un documento YAML che contiene la configurazione di build per il flusso di lavoro creato.

Esempio: creazione di un flusso di lavoro

```
aws imagebuilder create-workflow --name example-build-workflow --semantic-
version 1.0.0 --type BUILD --data file://example-build-workflow.yml
```

Output:

```
{
"workflowBuildVersionArn": "arn:aws:imagebuilder:us-west-2:111122223333:workflow/
build/example-build-workflow/1.0.0/1",
"clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

L'esempio seguente mostra come creare un flusso di lavoro di test con il comando create-workflow AWS CLI . Il `--data` parametro si riferisce a un documento YAML che contiene la configurazione di build per il flusso di lavoro creato.

Esempio: creazione di un flusso di lavoro di test

```
aws imagebuilder create-workflow --name example-test-workflow --semantic-
version 1.0.0 --type TEST --data file://example-test-workflow.yml
```

Output:

```
{
"workflowBuildVersionArn": "arn:aws:imagebuilder:us-west-2:111122223333:workflow/
test/example-test-workflow/1.0.0/1",
"clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

Creare un documento di workflow YAML

Il documento di definizione del formato YAML configura le fasi di input, output e flusso di lavoro per le fasi di creazione e test del processo di creazione dell'immagine. Puoi iniziare da modelli che includono passaggi standardizzati oppure puoi iniziare da zero per definire il tuo flusso di lavoro. Sia che utilizzi un modello o inizi da zero, puoi personalizzare il flusso di lavoro in base alle tue esigenze.

Struttura di un documento di workflow YAML

Il documento del flusso di lavoro YAML utilizzato da Image Builder per eseguire azioni di creazione e test delle immagini è strutturato come segue.

- Identificazione del documento di workflow
- Parametri di input del documento del workflow
- Fasi del documento del workflow
- Output dei documenti del flusso di lavoro

Identificazione del documento di workflow

Identifica in modo univoco il flusso di lavoro. Questa sezione può includere i seguenti attributi.

Campo	Descrizione	Tipo	Richiesto
nome	Il nome del documento del flusso di lavoro.	Stringa	No
description	La descrizione del documento.	Stringa	No
schemaVersion	La versione dello schema del documento, attualmente 1.0.	Stringa	Sì

Esempio

```
---
name: sample-test-image
description: Workflow for a sample image, with extra configuration options exposed
 through workflow parameters.
schemaVersion: 1.0
```

Parametri di input del documento del workflow

Questa parte del documento del flusso di lavoro definisce i parametri di input che il chiamante può specificare. Se non disponi di parametri, puoi omettere questa sezione. Se specificate dei parametri, ogni parametro può includere i seguenti attributi.

Campo	Descrizione	Tipo	Richiesto	Vincoli
nome	Nome del parametro.	Stringa	Sì	
description	La descrizione del parametro.	Stringa	No	
default	Il valore predefinito del parametro, se non viene fornito alcun valore. Se non includete un valore predefinito nella definizione del parametro, il valore del parametro è obbligato	Corrisponde al tipo di dati del parametro.	No	

Campo	Descrizione	Tipo	Richiesto	Vincoli
	rio in fase di esecuzione.			
tipo	Il tipo di dati del parametro. Se non includete il tipo di dati nella definizione del parametro, il tipo di parametro per impostazione predefinita è un valore di stringa richiesto in fase di esecuzione.	Stringa	Sì	Il tipo di dati del parametro deve essere uno dei seguenti: • `string` • `integer` • `boolean` • `stringList`

Esempio

Specificate il parametro nel documento del flusso di lavoro.

```
parameters:
  - name: waitForActionAtEnd
    type: boolean
    default: true
    description: "Wait for an external action at the end of the workflow"
```

Utilizzate il valore del parametro nel documento del flusso di lavoro.

```
$.parameters.waitForActionAtEnd
```

Fasi del documento del workflow

Speciifica fino a 15 azioni in fasi per il flusso di lavoro. Le fasi vengono eseguite nell'ordine in cui sono definite nel documento del flusso di lavoro. In caso di errore, il rollback viene eseguito in ordine inverso, iniziando dal passaggio non riuscito e procedendo a ritroso rispetto ai passaggi precedenti.

Ogni passaggio può fare riferimento all'output di qualsiasi azione del passaggio precedente. Questa operazione è nota come concatenamento o riferimento. Per fare riferimento all'output di un'azione precedente, puoi usare un JSONPath selettore. Per esempio:

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

Per ulteriori informazioni, consulta Utilizzate variabili dinamiche nel documento del flusso di lavoro.

> **Note**
>
> Anche se la fase stessa non ha un attributo di output, qualsiasi output di un'azione di passaggio è incluso nella `stepOutput` fase.

Ogni passaggio può includere i seguenti attributi.

Campo	Descrizione	Tipo	Richiesto	Valore predefinito	Vincoli
action	L'azione del flusso di lavoro eseguita da questo passaggio.	Stringa	Sì		Deve essere un'azione graduale supportat a per i documenti del flusso di lavoro di Image Builder.
`if`, seguito da una serie di istruzion i condizion ali che	Le istruzion i condizion ali aggiungon o punti decisiona li relativi al	Dict	No		Image Builder supporta le seguenti istruzioni condizion ali come

Campo	Descrizione	Tipo	Richiesto	Valore predefinito	Vincoli
modificano l'`if`operatore.	flusso di controllo al corpo delle fasi del flusso di lavoro.				modificatori dell'operatore: `if` • Condizioni di ramificazione e modificatori:,,,. `if` `and` `or` `not` Le condizioni di ramificazione vengono specificate singolarmente su una riga. • Operatori di confronto: `booleanEquals` `numberEquals` ,`numberGreaterThan` ,`numberGreaterThanEquals` ,, `numberLessThan` `numberLessThanEqua`

Campo	Descrizione	Tipo	Richiesto	Valore predefinito	Vincoli
					`ls` ,`stringEquals` .
description	La descrizione della fase.	Stringa	No		Le stringhe vuote non sono consentite. Se inclusa, la lunghezza deve essere compresa tra 1 e 1024 caratteri.
inputs	Contiene i parametri necessari per l'esecuzione dell'azione del passaggio. È possibile specificare i valori chiave come valori statici o con una JSONPath variabile che si risolve nel tipo di dati corretto.	Dict	Sì		

Campo	Descrizione	Tipo	Richiesto	Valore predefinito	Vincoli
nome	Il nome della fase. Questo nome deve essere univoco all'interno del documento del flusso di lavoro.	Stringa	Sì		La lunghezza deve essere compresa tra 3 e 128 caratteri. Può includere caratteri alfanumerici e. _ Senza spazi.

Campo	Descrizione	Tipo	Richiesto	Valore predefinito	Vincoli
onFailure	Configura l'azione da eseguire in caso di errore del passaggio, come segue. Comportamento • `Abort`— Non riesce a completare il passaggio, non riesce a completare il flusso di lavoro e non esegue i passaggi rimanenti dopo il passaggio non riuscito. Se il rollback è abilitato	Stringa	No	`Abort`	`Abort` | `Continue`

Campo	Descrizione	Tipo	Richiesto	Valore predefinito	Vincoli
	, il rollback inizia con il passaggio non riuscito e continua fino a quando non vengono ripristinati tutti i passaggi che lo consentono. • `Continue`— Fallisce il passaggio, ma continua a eseguire i passaggi rimanenti dopo il passaggio fallito. In questo caso, non è previsto alcun rollback.				

Campo	Descrizione	Tipo	Richiesto	Valore predefinito	Vincoli
Rollback abilitato	Configura se il passaggio verrà ripristinato in caso di errore. È possibile utilizzare un valore booleano statico o una JSONPath variabile dinamica che si risolve in un valore booleano.	Boolean	No	`true`	`true` | `false` | o una JSONPath variabile che si risolve in vero o falso.
timeoutSeconds	Il tempo massimo, in secondi, di esecuzione del passaggio prima di fallire e riprovare, se si verificano nuovi tentativi.	Numero intero	No	Dipende dall'impostazione predefinita definita per l'azione della fase, se applicabile.	Tra 1-86400 secondi (massimo 24 ore)

Esempio

```
steps:
  - name: LaunchTestInstance
    action: LaunchInstance
    onFailure: Abort
    inputs:
      waitFor: "ssmAgent"

  - name: ApplyTestComponents
    action: ExecuteComponents
    onFailure: Abort
    inputs:
      instanceId.$: "$.stepOutputs.LaunchTestInstance.instanceId"

  - name: TerminateTestInstance
    action: TerminateInstance
    onFailure: Continue
    inputs:
      instanceId.$: "$.stepOutputs.LaunchTestInstance.instanceId"

  - name: WaitForActionAtEnd
    action: WaitForAction
    if:
      booleanEquals: true
      value: "$.parameters.waitForActionAtEnd"
```

Output dei documenti del flusso di lavoro

Definisce gli output per il flusso di lavoro. Ogni output è una coppia di valori chiave che specifica il nome dell'output e il valore. È possibile utilizzare gli output per esportare dati in fase di esecuzione che i flussi di lavoro successivi possono utilizzare. Questa sezione è facoltativa.

Ogni output definito include i seguenti attributi.

Campo	Descrizione	Tipo	Richiesto
nome	Il nome dell'output. Il nome deve essere univoco in tutti i flussi	Stringa	Sì

Campo	Descrizione	Tipo	Richiesto
	di lavoro che includi nella pipeline.		
value	Il valore per l'output. Il valore della stringa può essere una variabile dinamica, ad esempio un file di output di un'azione graduale. Per ulteriori informazioni, consulta Utilizzate variabili dinamiche nel documento del flusso di lavoro.	Stringa	Sì

Esempio

Create un ID dell'immagine di output per il documento del flusso di lavoro con l'output del `createProdImage` passaggio.

```
outputs:
  - name: 'outputImageId'
    value: '$.stepOutputs.createProdImage.imageId'
```

Fate riferimento all'output del flusso di lavoro nel flusso di lavoro successivo.

```
$.workflowOutputs.outputImageId
```

Azioni in fasi supportate per il documento del flusso di lavoro

Questa sezione include dettagli sulle azioni dei passaggi supportate da Image Builder.

Termini utilizzati in questa sezione

AMI

Amazon Machine Image

ARN

Nome della risorsa Amazon

Azioni supportate

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

BootstrapInstanceForContainer

Questa operazione esegue uno script di servizio per avviare l'istanza con i requisiti minimi per eseguire i flussi di lavoro dei container. Image Builder utilizza l'sendCommandAPI Systems Manager per eseguire questo script. Per ulteriori informazioni, vedere AWS Systems Manager Run Command.

> **Note**
>
> Lo script bootstrap installa i pacchetti Docker AWS CLI e Docker che sono i prerequisiti per consentire a Image Builder di creare correttamente contenitori Docker. Se non includi questa operazione, la creazione dell'immagine potrebbe fallire.

Timeout predefinito: 60 minuti

Rollback: non è previsto alcun rollback per questa operazione.

Input: la tabella seguente include gli input supportati per questa operazione.

Nome di input	Descrizione	Tipo	Obbligatorio	Predefinita	Vincoli
instanceId	L'ID dell'istanza da avviare.	Stringa	Sì		Deve essere l'ID dell'istanza di output della fase del flusso di lavoro che ha avviato l'istanza per questo flusso di lavoro.

Output: la tabella seguente include gli output per questa azione.

Nome output	Descrizione	Tipo
runCommandId	L'ID del Systems Manager sendCommand che ha eseguito lo script di bootstrap sull'istanza.	Stringa
status	Lo stato restituito dal Systems ManagersendCommand.	Stringa
output	Output restituito dal Systems ManagersendCommand.	Stringa

Esempio

Specificate l'azione della fase nel documento del flusso di lavoro.

```
- name: ContainerBootstrapStep
```

```
  action: BootstrapInstanceForContainer
  onFailure: Abort
  inputs:
      instanceId.$: $.stepOutputs.LaunchStep.instanceId
```

Utilizzate l'output del valore dell'azione della fase nel documento del flusso di lavoro.

```
$.stepOutputs.ContainerBootstrapStep.status
```

CollectImageMetadata

Questa operazione è valida solo per i flussi di lavoro di creazione.

EC2 Image Builder esegue AWS Systems Manager (Systems Manager) Agent sulle EC2 istanze che avvia per creare e testare l'immagine. Image Builder raccoglie informazioni aggiuntive sull'istanza utilizzata durante la fase di creazione con Systems Manager Inventory. Queste informazioni includono il nome e la versione del sistema operativo (OS), nonché l'elenco dei pacchetti e delle rispettive versioni riportate dal sistema operativo.

> **Note**
>
> Questa operazione funziona solo per le immagini che creano AMIs.

Timeout predefinito: 30 minuti

Rollback: Image Builder ripristina tutte le risorse Systems Manager create durante questa fase.

Input: la tabella seguente include gli input supportati per questa operazione.

Nome di input	Descrizione	Tipo	Obbligatorio	Predefinita	Vincoli
instanceId	L'istanza di build a cui applicare le impostazioni dei metadati.	Stringa	Sì		Deve essere l'ID dell'istanza di output della fase del flusso di lavoro che ha avviato l'istanza di

Nome di input	Descrizione	Tipo	Obbligatorio	Predefinita	Vincoli
					creazione per questo flusso di lavoro.

Output: la tabella seguente include gli output per questa azione.

Nome output	Descrizione	Tipo
osVersion	Il nome e la versione del sistema operativo raccolti dall'istanza di build.	Stringa
ID dell'associazione	L'ID di associazione Systems Manager utilizzato per la raccolta dell'inventario.	Stringa

Esempio

Specificate l'azione della fase nel documento del flusso di lavoro.

```
- name: CollectMetadataStep
  action: CollectImageMetadata
  onFailure: Abort
  inputs:
      instanceId: $.stepOutputs.LaunchStep.instanceId
```

Utilizza l'output dell'azione della fase nel documento del flusso di lavoro.

```
$.stepOutputs.CollectMetadataStep.osVersion
```

CollectImageScanFindings

Se Amazon Inspector è abilitato per il tuo account e la scansione delle immagini è abilitata per la tua pipeline, questa operazione raccoglie i risultati della scansione delle immagini riportati da Amazon Inspector per la tua istanza di test. Questa operazione non è disponibile per i flussi di lavoro di compilazione.

Timeout predefinito: 120 minuti

Rollback: non è previsto alcun rollback per questa operazione.

Input: la tabella seguente include gli input supportati per questa operazione.

Nome di input	Descrizione	Tipo	Obbligatorio	Predefinita	Vincoli
instanceId	L'ID dell'istanza su cui è stata eseguita la scansione.	Stringa	Sì		Deve essere l'ID dell'istanza di output della fase del flusso di lavoro che ha avviato l'istanza per questo flusso di lavoro.

Output: la tabella seguente include gli output per questa azione.

Nome output	Descrizione	Tipo
runCommandId	L'ID del Systems Manager sendCommand che ha eseguito lo script per raccogliere i risultati.	Stringa
status	Lo stato restituito dal Systems ManagersendCommand.	Stringa
output	Output restituito dal Systems ManagersendCommand.	Stringa

Esempio

Specificate l'azione della fase nel documento del flusso di lavoro.

```
- name: CollectFindingsStep
  action: CollectImageScanFindings
  onFailure: Abort
  inputs:
      instanceId.$: $.stepOutputs.LaunchStep.instanceId
```

Utilizzate l'output del valore dell'azione della fase nel documento del flusso di lavoro.

```
$.stepOutputs.CollectFindingsStep.status
```

CreateImage

Questa operazione crea un'immagine da un'istanza in esecuzione con l' EC2 `CreateImage`API Amazon. Durante il processo di creazione, l'azione della fase attende se necessario per verificare che le risorse abbiano raggiunto lo stato corretto prima di continuare.

Timeout predefinito: 720 minuti

Rollback: non è previsto alcun rollback per questa operazione.

Input: la tabella seguente include gli input supportati per questa operazione.

Nome di input	Descrizione	Tipo	Obbligatorio	Predefinita	Vincoli
instanceId	L'istanza da cui creare la nuova immagine.	Stringa	Sì		L'istanza per l'ID di istanza fornito deve trovarsi in uno `running` stato all'inizio di questo passaggio.

Output: la tabella seguente include gli output per questa azione.

Nome output	Descrizione	Tipo
imageId	L'ID AMI dell'immagine creata.	Stringa

Esempio

Specificate l'azione della fase nel documento del flusso di lavoro.

```
- name: CreateImageFromInstance
  action: CreateImage
  onFailure: Abort
  inputs:
      instanceId.$: "i-1234567890abcdef0"
```

Utilizzate l'output del valore dell'azione della fase nel documento del flusso di lavoro.

```
$.stepOutputs.CreateImageFromInstance.imageId
```

ExecuteComponents

Questa operazione esegue i componenti specificati nella ricetta per l'immagine corrente in fase di creazione. I flussi di lavoro di compilazione eseguono i componenti di compilazione sull'istanza di compilazione. I flussi di lavoro di test eseguono solo componenti di test sull'istanza di test.

Image Builder utilizza l'sendCommandAPI Systems Manager per eseguire i componenti. Per ulteriori informazioni, vedere AWS Systems Manager Run Command.

Timeout predefinito: 720 minuti

Rollback: non è previsto alcun rollback per questa operazione.

Input: la tabella seguente include gli input supportati per questa operazione.

Nome di input	Descrizione	Tipo	Obbligatorio	Predefinita	Vincoli
instanceId	L'ID dell'istanza su cui devono essere eseguiti i componenti.	Stringa	Sì		Deve essere l'ID dell'istanza di output della fase del flusso di lavoro che ha avviato l'istanza per

Nome di input	Descrizione	Tipo	Obbligatorio	Predefinita	Vincoli
					questo flusso di lavoro.

Output: la tabella seguente include gli output per questa azione.

Nome output	Descrizione	Tipo
runCommandId	L'ID del Systems Manager sendCommand che ha eseguito i componenti sull'istanza.	Stringa
status	Lo stato restituito dal Systems ManagersendCommand.	Stringa
output	Output restituito dal Systems ManagersendCommand.	Stringa

Esempio

Specificate l'azione della fase nel documento del flusso di lavoro.

```
- name: ExecComponentsStep
  action: ExecuteComponents
  onFailure: Abort
  inputs:
      instanceId: $.stepOutputs.LaunchStep.instanceId
```

Utilizza l'output dell'azione della fase nel documento del flusso di lavoro.

```
$.stepOutputs.ExecComponentsStep.status
```

LaunchInstance

Questa operazione avvia un'istanza nell'istanza Account AWS e attende che l'agente Systems Manager sia in esecuzione sull'istanza prima di passare alla fase successiva. L'azione di avvio

utilizza le impostazioni della ricetta e delle risorse di configurazione dell'infrastruttura associate all'immagine. Ad esempio, il tipo di istanza da avviare deriva dalla configurazione dell'infrastruttura. L'output è l'ID dell'istanza che è stata avviata.

L'`waitFor`input configura la condizione che soddisfa il requisito di completamento della fase.

Timeout predefinito: 60 minuti

Rollback: per le istanze di build, il rollback esegue l'azione che hai configurato nella risorsa di configurazione dell'infrastruttura. Per impostazione predefinita, le istanze di compilazione vengono terminate se la creazione dell'immagine fallisce. Tuttavia, esiste un'impostazione nella configurazione dell'infrastruttura per mantenere l'istanza di build per la risoluzione dei problemi.

Input: la tabella seguente include gli input supportati per questa operazione.

Nome di input	Descrizione	Tipo	Obbligatorio	Predefinita	Vincoli
imageIdOverride	L'immagine da usare per avviare l'istanza	Stringa	No	Fase di creazione: immagine di base della ricetta dell'immagine Fase di test: AMI in uscita dalla fase di costruzione	Deve essere un ID AMI valido
instanceTypesOverride	Image Builder prova ogni tipo di istanza nell'elenco finché non ne trova uno che si avvia correttamente	Elenco di String	No	Tipi di istanze specificati nella configurazione dell'infrastruttura	Devono essere tipi di istanza validi

Nome di input	Descrizione	Tipo	Obbligatorio	Predefinita	Vincoli
Attendi	La condizione da attendere prima di completare la fase del flusso di lavoro e passare alla fase successiva	Stringa	Sì		Image Builder supporta. `ssmAgent`

Output: la tabella seguente include gli output per questa operazione.

Nome output	Descrizione	Tipo
instanceId	L'ID dell'istanza che è stata avviata.	Stringa

Esempio

Specificate l'azione della fase nel documento del flusso di lavoro.

```
- name: LaunchStep
  action: LaunchInstance
  onFailure: Abort
  inputs:
    waitFor: ssmAgent
```

Utilizza l'output dell'azione della fase nel documento del flusso di lavoro.

```
$.stepOutputs.LaunchStep.instanceId
```

RunCommand

Questa operazione esegue un documento di comando per il flusso di lavoro. Image Builder utilizza l'sendCommandAPI Systems Manager per eseguirlo automaticamente. Per ulteriori informazioni, vedere AWS Systems Manager Run Command.

Timeout predefinito: 12 ore

Rollback: non è previsto alcun rollback per questa operazione.

Input: la tabella seguente include gli input supportati per questa operazione.

Nome di input	Descrizione	Tipo	Obbligatorio	Predefinita	Vincoli
instanceId	L'ID dell'istanza su cui eseguire il documento di comando.	Stringa	Sì		Deve essere l'ID dell'istanza di output della fase del flusso di lavoro che ha avviato l'istanza per questo flusso di lavoro.
documentName	Il nome del documento di comando Systems Manager da eseguire.	Stringa	Sì		
parametri	Un elenco di coppie chiave-valore per tutti i parametri richiesti dal	<string>dizionario <string, list >	Condizionale		

Nome di input	Descrizione	Tipo	Obbligatorio	Predefinita	Vincoli
	documento di comando.				
Versione del documento	La versione del documento di comando da eseguire.	Stringa	No	$DEFAULT	

Output: la tabella seguente include gli output per questa azione.

Nome output	Descrizione	Tipo
runCommandId	L'ID del Systems Manager sendCommand che ha eseguito il documento di comando sull'istanza.	Stringa
status	Lo stato restituito dal Systems ManagersendCommand.	Stringa
output	Output restituito dal Systems ManagersendCommand.	Elenco di stringhe

Esempio

Specificate l'azione della fase nel documento del flusso di lavoro.

```
- name: RunCommandDoc
  action: RunCommand
  onFailure: Abort
  inputs:
    documentName: SampleDocument
    parameters:
        osPlatform:
          - "linux"
```

```
    instanceId.$: $.stepOutputs.LaunchStep.instanceId
```

Utilizzate l'output del valore dell'azione della fase nel documento del flusso di lavoro.

```
$.stepOutputs.RunCommandDoc.status
```

RunSysPrep

Questa operazione utilizza l'sendCommandAPI Systems Manager per eseguire il `AWSEC2-RunSysprep` documento per le istanze di Windows prima che l'istanza di compilazione si chiuda per l'istantanea. Queste azioni seguono le AWS migliori pratiche per il rafforzamento e la pulizia dell'immagine.

Timeout predefinito: 60 minuti

Rollback: non è previsto alcun rollback per questa operazione.

Input: la tabella seguente include gli input supportati per questa operazione.

Nome di input	Descrizione	Tipo	Obbligatorio	Predefinita	Vincoli
instanceId	L'ID dell'istanza su cui eseguire il `AWSEC2-RunSysprep` documento.	Stringa	Sì		Deve essere l'ID dell'istanza di output della fase del flusso di lavoro che ha avviato l'istanza per questo flusso di lavoro.

Output: la tabella seguente include gli output per questa azione.

Nome output	Descrizione	Tipo
runCommandId	L'ID del Systems Manager sendCommand che ha	Stringa

Nome output	Descrizione	Tipo
	eseguito il `AWSEC2-RunSysprep` documento sull'istanza.	
status	Lo stato restituito dal Systems ManagersendCommand.	Stringa
output	Output restituito dal Systems ManagersendCommand.	Stringa

Esempio

Specificate l'azione della fase nel documento del flusso di lavoro.

```
- name: RunSysprep
  action: RunSysPrep
  onFailure: Abort
  inputs:
      instanceId.$: $.stepOutputs.LaunchStep.instanceId
```

Utilizzate l'output del valore dell'azione della fase nel documento del flusso di lavoro.

```
$.stepOutputs.RunSysprep.status
```

SanitizeInstance

Questa operazione esegue lo script di sanificazione consigliato per le istanze Linux prima che l'istanza di compilazione si chiuda per l'istantanea. Lo script sanitize aiuta a garantire che l'immagine finale segua le migliori pratiche di sicurezza e che gli artefatti o le impostazioni di build che non devono essere trasferiti all'istantanea vengano rimossi. Per ulteriori informazioni sullo script, consulta. Pulizia post-compilazione richiesta Questa operazione non si applica alle immagini dei contenitori.

Image Builder utilizza l'sendCommandAPI Systems Manager per eseguire questo script. Per ulteriori informazioni, vedere AWS Systems Manager Run Command.

Timeout predefinito: 60 minuti

Rollback: non è previsto alcun rollback per questa operazione.

Input: la tabella seguente include gli input supportati per questa operazione.

Nome di input	Descrizione	Tipo	Obbligatorio	Predefinita	Vincoli
instanceId	L'ID dell'istanza da disinfettare.	Stringa	Sì		Deve essere l'ID dell'istanza di output della fase del flusso di lavoro che ha avviato l'istanza per questo flusso di lavoro.

Output: la tabella seguente include gli output per questa azione.

Nome output	Descrizione	Tipo
runCommandId	L'ID del Systems Manager sendCommand che ha eseguito lo script di sanificazione sull'istanza.	Stringa
status	Lo stato restituito dal Systems ManagersendCommand.	Stringa
output	Output restituito dal Systems ManagersendCommand.	Stringa

Esempio

Specificate l'azione della fase nel documento del flusso di lavoro.

```
- name: SanitizeStep
  action: SanitizeInstance
  onFailure: Abort
```

```
  inputs:
      instanceId: $.stepOutputs.LaunchStep.instanceId
```

Utilizzate l'output del valore dell'azione della fase nel documento del flusso di lavoro.

```
$.stepOutputs.SanitizeStep.status
```

TerminateInstance

Questa operazione termina l'istanza con l'ID dell'istanza passato come input.

Timeout predefinito: 30 minuti

Rollback: non è previsto alcun rollback per questa operazione.

Input: la tabella seguente include gli input supportati per questa operazione.

Nome di input	Descrizione	Tipo	Obbligatorio	Predefinita	Vincoli
instanceId	L'ID dell'istanza da terminare.	Stringa	Sì		

Uscite: non ci sono uscite per questa operazione.

Esempio

Specificate l'azione della fase nel documento del flusso di lavoro.

```
- name: TerminateInstance
  action: TerminateInstance
  onFailure: Continue
  inputs:
      instanceId.$: i-1234567890abcdef0
```

WaitForAction

Questa operazione mette in pausa il flusso di lavoro in esecuzione e attende di ricevere un'azione esterna dall'azione dell'API Image SendWorkflowStepAction Builder. Questo passaggio pubblica un EventBridge evento nel bus degli EventBridge eventi predefinito con il tipo di dettaglio. `EC2 Image`

`Builder Workflow Step Waiting` La fase può anche inviare una notifica SNS se si fornisce un ARN per argomento SNS.

Timeout predefinito: 3 giorni

Rollback: non è previsto alcun rollback per questa operazione.

Input: la tabella seguente include gli input supportati per questa operazione.

Nome di input	Descrizione	Tipo	Obbligatorio	Predefinita	Vincoli
snsTopicArn	Un argomento SNS ARN opzionale a cui inviare una notifica quando la fase del flusso di lavoro è in sospeso.	Stringa	No		

Output: la tabella seguente include gli output per questa azione.

Nome output	Descrizione	Tipo
action	L'azione restituita dall'azione SendWorkflowStepAction API.	Stringa (`RESUMEoSTOP`)
motivo	Il motivo dell'azione restituita.	Stringa

Esempio

Specificate l'azione della fase nel documento del flusso di lavoro.

```
- name: SendEventAndWait
  action: WaitForAction
```

```
  onFailure: Abort
  inputs:
    snsTopicArn: arn:aws:sns:us-west-2:111122223333:ExampleTopic
```

Utilizzate l'output del valore dell'azione della fase nel documento del flusso di lavoro.

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

Utilizzate variabili dinamiche nel documento del flusso di lavoro

È possibile utilizzare variabili dinamiche nei documenti del flusso di lavoro per rappresentare valori che variano in fase di esecuzione del processo di creazione dell'immagine. I valori delle variabili dinamiche sono rappresentati come JSONPath selettori con nodi strutturali che identificano in modo univoco la variabile di destinazione.

JSONPath struttura variabile dinamica del flusso di lavoro

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

Il primo nodo dopo la radice ($) si riferisce alla struttura del documento del flusso di lavoro, ad esempio `ostepOutputs`, nel caso delle variabili di sistema di Image Builder,. `imageBuilder` L'elenco seguente contiene i nodi della struttura dei documenti JSONPath del flusso di lavoro supportati.

Nodi della struttura del documento

- parametri: i parametri del flusso di lavoro
- StepOutputs: restituisce i risultati di una fase dello stesso documento del flusso di lavoro
- WorkflowOutputs: output da un documento del flusso di lavoro già eseguito
- imagebuilder - Variabili di sistema di Image Builder

I `parameters` nodi della struttura `stepOutputs` del documento includono un nodo opzionale per il nome del passaggio. Questo aiuta a garantire nomi di variabili univoci in tutti i passaggi.

L'ultimo nodo di JSONPath è il nome della variabile di destinazione, ad esempio`instanceId`.

Ogni passaggio può fare riferimento all'output di qualsiasi azione del passaggio precedente con queste variabili JSONPath dinamiche. Questa operazione è nota anche come concatenamento o

riferimento. Per fare riferimento all'output di un'azione precedente, è possibile utilizzare la seguente variabile dinamica.

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

Quando un parametro di input fa riferimento a una variabile dinamica, l'indicatore di concatenamento (`.$`) deve essere collegato alla fine del nome del parametro, come illustrato nell'esempio seguente.

Esempio

```
- name: ApplyTestComponents
  action: ExecuteComponents
  onFailure: Abort
  inputs:
    instanceId.$: "$.stepOutputs.LaunchTestInstance.instanceId"
```

Usa le variabili di sistema di Image Builder

Image Builder fornisce le seguenti variabili di sistema che è possibile utilizzare nel documento del flusso di lavoro:

Nome della variabile	Descrizione	Tipo	Valore di esempio
cloudWatchLogGruppo	Il nome del gruppo CloudWatch Logs per i log di output. Formato: `/aws/imagebuilder/`*`<recipe-name>`*	Stringa	`/aws/imagebuilder/`*`sampleImageRecipe`*
cloudWatchLogStream	Il nome del flusso CloudWatch Logs per i log di output.	Stringa	*`1.0.0/1`*
collectImageMetadata	L'impostazione che indica a Image Builder	Booleano	`true` | `false`

Nome della variabile	Descrizione	Tipo	Valore di esempio
	se raccogliere i metadati dell'istanza.		
collectImageScanRisultati	Il valore corrente dell'impostazione che consente a Image Builder di raccogliere i risultati della scansione delle immagini.	Booleano	`true` | `false`
imageBuildNumber	Il numero di versione di build dell'immagine.	Numero intero	*1*
imageId	L'ID AMI dell'immagine di base.	Stringa	*ami-1234567890abcdef1*
Nome dell'immagine	Il nome dell'immagine.	Stringa	*sampleImage*
imageType	Il tipo di output dell'immagine.	Stringa	`AMI` | `Docker`
imageVersionNumber	Il numero di versione dell'immagine.	Stringa	*1.0.0*
instanceProfileName	Il nome del ruolo del profilo dell'istanza utilizzato da Image Builder per avviare le istanze di compilazione e test.	Stringa	*SampleImageBuilderInstanceProfileRole*

Nome della variabile	Descrizione	Tipo	Valore di esempio
platform	La piattaforma del sistema operativo dell'immagine creata.	Stringa	`Linux` | `Windows` | `MacOS`
Registri S3	Un oggetto JSON che contiene la configura zione per i log S3 scritti da Image Builder.	Oggetto JSON	{'S3logs': {'s3 ':' ', 's3BucketName': *sample-bucket* '}} KeyPrefix *ib-logs*
securityGroups	Il gruppo di sicurezza IDs che si applica alla creazione e al test delle istanze.	Elenco [Stringa]	*[sg-12345 67890abcd ef1, sg-111122 22333344445]*
Fonte ImageRan	L'Amazon Resource Name (ARN) della risorsa di immagine Image Builder utilizzata dal flusso di lavoro per le fasi di creazione e test.	Stringa	arn:aws:imagebuild er: ::image//*us-east-1 111122223 333 sampleIma ge 1.0.0/1*
subnetId	L'ID della sottorete in cui avviare le istanze di build e test.	Stringa	*subnet-12 34567890a bcdef1*

Nome della variabile	Descrizione	Tipo	Valore di esempio
terminateInstanceOnFallimento	Il valore corrente dell'impostazione che ordina a Image Builder di terminare l'istanza in caso di errore o di conservarla per la risoluzione dei problemi.	Booleano	`true` | `false`
Fase del flusso di lavoro	La fase corrente in esecuzione per l'esecuzione del flusso di lavoro.	Stringa	`Build` | `Test`
workingDirectory	Il percorso della directory di lavoro.	Stringa	`/tmp`

Utilizza le istruzioni condizionali nelle fasi del flusso di lavoro

Le istruzioni condizionali iniziano con l'attributo `if` statement document. Lo scopo ultimo dell'`if`istruzione è determinare se eseguire l'azione del passaggio o ignorarla. Se l'`if`istruzione viene risolta in`true`, viene eseguita l'azione del passaggio. Se si risolve in`false`, Image Builder salta l'azione del passaggio e registra lo stato `SKIPPED` del passaggio nel registro.

L'`if`istruzione supporta istruzioni ramificate (`and`,`or`) e modificatori condizionali (). `not` Supporta inoltre i seguenti operatori di confronto che eseguono confronti di valori (uguali, minori, maggiori di) in base ai tipi di dati confrontati (stringa o numero).

Operatori di confronto supportati

- `booleanEquals`
- `numberEquals`
- `numberGreaterThan`

- `numberGreaterThanEquals`
- `numberLessThan`
- `numberLessThanEquals`
- `stringEquals`

Regole per le istruzioni di ramificazione e i modificatori condizionali

Le seguenti regole si applicano alle istruzioni di ramificazione (`and`,`or`) e ai modificatori condizionali (). `not`

- Le istruzioni di ramificazione e i modificatori condizionali devono apparire su una riga a sé stanti.
- Le istruzioni di ramificazione e i modificatori condizionali devono seguire le regole di livello.
  - Può esserci una sola dichiarazione a livello principale.
  - Ogni ramo o modificatore figlio inizia un nuovo livello.

  Per ulteriori informazioni sui livelli, vedereLivelli annidati nelle istruzioni condizionali.
- Ogni istruzione ramificata deve avere almeno un'istruzione condizionale secondaria, ma non più di dieci.
- I modificatori condizionali operano solo su un'istruzione condizionale secondaria.

Livelli annidati nelle istruzioni condizionali

Le istruzioni condizionali operano a diversi livelli in una sezione a sé stante. Ad esempio, l'attributo `if` statement appare allo stesso livello nel documento del flusso di lavoro del nome e dell'azione della fase. Questa è la base dell'istruzione condizionale.

È possibile specificare fino a quattro livelli di istruzioni condizionali, ma solo un'istruzione può apparire a livello principale. Tutte le altre istruzioni ramificate, i modificatori condizionali o gli operatori condizionali vengono indentati da lì, un rientro per livello.

Lo schema seguente mostra il numero massimo di livelli annidati per un'istruzione condizionale.

```
base:
  parent:
    - child (level 2)
        - child (level 3)
```

```
            child (level 4)
```

Attributo `if`

L'`if`attributo specifica l'istruzione condizionale come attributo del documento. Questo è il livello zero.

Livello genitore

Questo è il primo livello di annidamento per le istruzioni condizionali. Può esserci una sola istruzione a questo livello. Se non hai bisogno di ramificazioni o modificatori, questo può essere un operatore condizionale senza istruzioni secondarie. Questo livello non utilizza la notazione tratteggiata, ad eccezione degli operatori condizionali.

Livelli per bambini

I livelli da due a quattro sono considerati livelli per bambini. Le istruzioni secondarie possono includere istruzioni ramificate, modificatori condizionali o operatori condizionali.

Esempio: livelli annidati

L'esempio seguente mostra il numero massimo di livelli in un'istruzione condizionale.

```
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"
```

Regole di nidificazione

- Ogni ramo o modificatore a livello di bambino inizia un nuovo livello.
- Ogni livello è rientrato.
- Possono esserci un massimo di quattro livelli, tra cui un'istruzione, un modificatore o un operatore a livello principale, e fino a tre livelli aggiuntivi.

Esempi di istruzioni condizionali

Questo gruppo di esempi mostra vari aspetti delle istruzioni condizionali.

Ramificazione: e

L'istruzione di `and` ramificazione opera su un elenco di espressioni che sono figlie del ramo, ognuna delle quali deve restituire a. `true` Image Builder valuta le espressioni nell'ordine in cui appaiono nell'elenco. Se un'espressione restituisce un risultato pari a`false`, l'elaborazione si interrompe e viene considerato il ramo. `false`

L'esempio seguente restituisce a`true`, perché entrambe le espressioni restituiscono a. `true`

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

Ramificazione: o

L'istruzione di `or` ramificazione opera su un elenco di espressioni che sono figlie del ramo, almeno una delle quali deve restituire a. `true` Image Builder valuta le espressioni nell'ordine in cui appaiono nell'elenco. Se un'espressione restituisce un risultato pari a`true`, l'elaborazione si interrompe e viene considerato il ramo. `true`

L'esempio seguente restituisce a`true`, anche se la prima espressione è. `false`

```
if:
  or:
    - stringEquals: 'test_string'
      value: 'test_string_not_equal'
    - numberEquals: 1
      value: 1
```

Modificatore condizionale: no

Il modificatore `not` condizionale annulla le istruzioni condizionali che sono figlie del ramo.

L'esempio seguente restituisce `true` quando il `not` modificatore annulla l'istruzione condizionale. `stringEquals`

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

Istruzione condizionale: booleanEquals

L'operatore di `booleanEquals` confronto confronta i valori booleani e restituisce true se i valori booleani corrispondono esattamente.

L'esempio seguente determina se è abilitato. `collectImageScanFindings`

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

Dichiarazione condizionale: stringEquals

L'operatore di `stringEquals` confronto confronta due stringhe e restituisce true se le stringhe corrispondono esattamente. Se uno dei due valori non è una stringa, Image Builder lo converte in una stringa prima del confronto.

L'esempio seguente confronta la variabile di sistema della piattaforma per determinare se il flusso di lavoro è in esecuzione su una piattaforma Linux.

```
if:
  - stringEquals: 'Linux'
    value: '$.imagebuilder.Platform'
```

Dichiarazione condizionale: numberEquals

L'operatore di `numberEquals` confronto confronta due numeri e restituisce true se i numeri sono uguali. I numeri da confrontare devono avere uno dei seguenti formati.

- Numero intero
- Float
- Una stringa che corrisponde al seguente schema regex:`^-?[0-9]+(\.)?[0-9]+$`.

Il seguente esempio confronta tutti con. `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
```

Gestisci la creazione di immagini personalizzate in Image Builder tramite un processo di pipeline ripetibile

Le pipeline di immagini di Image Builder forniscono un framework di automazione per la creazione e la manutenzione di immagini personalizzate AMIs e di contenitori. Le pipeline offrono le seguenti funzionalità:

- Assemblate l'immagine di base, i componenti per la creazione e il test, la configurazione dell'infrastruttura e le impostazioni di distribuzione.
- Semplifica la pianificazione dei processi di manutenzione automatizzati utilizzando la procedura guidata `Schedule builder` nella console o inserendo espressioni cron per gli aggiornamenti ricorrenti delle immagini.
- Abilita il rilevamento delle modifiche per l'immagine di base e i componenti, per saltare automaticamente le build pianificate quando non ci sono modifiche.
- Abilita l'automazione basata su regole tramite Amazon. EventBridge

> **Note**
>
> Per ulteriori informazioni sull'utilizzo dell' EventBridge API per visualizzare o modificare le regole, consulta Amazon EventBridge API Reference. Per ulteriori informazioni sull'uso dei EventBridge events comandi nelle regole AWS CLI per visualizzare o modificare, consulta gli eventi nel AWS CLI Command Reference.

Indice

- Elenca e visualizza i dettagli della pipeline
- Crea e aggiorna pipeline di immagini AMI
- Crea e aggiorna le pipeline di immagini dei contenitori
- Configurazione dei flussi di lavoro della pipeline di immagini in Image Builder
- Esegui la tua pipeline di immagini
- Usa le espressioni cron in Image Builder
- Usa EventBridge le regole con le pipeline di Image Builder

Elenca e visualizza i dettagli della pipeline

Questa sezione descrive i vari modi in cui è possibile trovare informazioni e visualizzare i dettagli per le pipeline di EC2 immagini di Image Builder.

Dettagli della pipeline

- Elenca le pipeline di immagini da AWS CLI
- Ottieni i dettagli della pipeline di immagini da AWS CLI

Elenca le pipeline di immagini da AWS CLI

L'esempio seguente mostra come utilizzare il list-image-pipelines comando in AWS CLI per elencare tutte le pipeline di immagini.

```
aws imagebuilder list-image-pipelines
```

Ottieni i dettagli della pipeline di immagini da AWS CLI

L'esempio seguente mostra come utilizzare il get-image-pipeline comando in AWS CLI per ottenere i dettagli su una pipeline di immagini tramite il relativo ARN.

```
aws imagebuilder get-image-pipeline --image-pipeline-arn arn:aws:imagebuilder:us-
west-2:123456789012:image-pipeline/my-example-pipeline
```

Crea e aggiorna pipeline di immagini AMI

È possibile impostare, configurare e gestire le pipeline di immagini AMI dalla console Image Builder, tramite l'API Image Builder o imagebuilder con i comandi in. AWS CLIÈ possibile utilizzare la procedura guidata della console Create image pipeline per guidarvi nei seguenti passaggi:

- Specificate i dettagli della pipeline come nome, descrizione e tag delle risorse.
- Seleziona una ricetta di immagini AMI che includa un'immagine di base da immagini gestite da Amazon ad avvio rapido, immagini che hai creato o che sono state condivise con te o immagini a cui ti iscrivi tramite. Marketplace AWS La ricetta include anche componenti che eseguono le seguenti attività sulle EC2 istanze utilizzate da Image Builder per creare l'immagine:
  - Aggiungere e rimuovere software

  - Personalizza impostazioni e script
  - Esegui test selezionati
- Specificate i flussi di lavoro per configurare le fasi di creazione e test delle immagini eseguite dalla pipeline.
- Definisci la configurazione dell'infrastruttura per la tua pipeline con impostazioni predefinite o impostazioni che configuri tu stesso. La configurazione include il tipo di istanza e la coppia di chiavi da utilizzare per l'immagine, le impostazioni di sicurezza e di rete, le impostazioni di archiviazione dei log e la risoluzione dei problemi e le notifiche SNS.

  Si tratta di un passaggio facoltativo. Image Builder utilizza le impostazioni predefinite per la configurazione dell'infrastruttura se non si definisce la configurazione personalmente.
- Definite le impostazioni di distribuzione per distribuire le immagini alle AWS regioni e agli account di destinazione. Puoi specificare una chiave KMS per la crittografia, configurare la condivisione AMI o la configurazione della licenza o configurare un modello di avvio per la AMIs distribuzione.

  Questo passaggio è facoltativo. Se non definisci tu stesso la configurazione, Image Builder utilizza la denominazione predefinita per l'AMI di output e distribuisce l'AMI nella regione di origine. La regione di origine è la regione in cui viene eseguita la pipeline.

Per ulteriori informazioni e un step-by-step tutorial sull'utilizzo della procedura guidata della console Create image pipeline con i valori predefiniti, se forniti, consulta. Tutorial: crea una pipeline di immagini con l'AMI di output dalla procedura guidata della console Image Builder

Indice

- Crea una pipeline di immagini AMI da AWS CLI
- Aggiorna le pipeline di immagini AMI dalla console
- Aggiorna le pipeline di immagini AMI da AWS CLI

Crea una pipeline di immagini AMI da AWS CLI

È possibile creare una pipeline di immagini AMI con un file JSON che contiene i dettagli di configurazione come input per il create-image-pipeline comando in. AWS CLI

La frequenza con cui la pipeline crea una nuova immagine per incorporare eventuali aggiornamenti in sospeso dell'immagine e dei componenti di base dipende dalla `schedule` configurazione. A `schedule` ha i seguenti attributi:

- `scheduleExpression`— Imposta la pianificazione dell'esecuzione della pipeline per valutare `pipelineExecutionStartCondition` e determinare se avviare una build. La pianificazione è configurata con espressioni cron. Per ulteriori informazioni su come formattare un'espressione cron in Image Builder, vedere. Usa le espressioni cron in Image Builder
- `pipelineExecutionStartCondition`— Determina se la pipeline deve avviare la compilazione. I valori validi includono:
  - `EXPRESSION_MATCH_ONLY`— la pipeline creerà una nuova immagine ogni volta che l'espressione cron corrisponde all'ora corrente.
  - `EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE`— la pipeline non avvierà una nuova creazione di immagini a meno che non vi siano modifiche in sospeso all'immagine o ai componenti di base.

Quando si esegue il create-image-pipeline comando in AWS CLI, molte risorse di configurazione sono opzionali. Tuttavia, alcune risorse hanno requisiti condizionali, a seconda del tipo di immagine creata dalla pipeline. Le seguenti risorse sono necessarie per le pipeline di immagini AMI:

- Immagine: ricetta ARN
- Configurazione dell'infrastruttura ARN

1. Creazione di un file JSON di input nella CLI

   Usa il tuo strumento di modifica dei file preferito per creare un file JSON con le seguenti chiavi, oltre a valori validi per il tuo ambiente. In questo esempio viene utilizzato un file denominato `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**
>
> - È necessario includere l'annotazione `file://` all'inizio del percorso del file JSON.
> - Il percorso del file JSON deve seguire la convenzione appropriata per il sistema operativo di base in cui si esegue il comando. Ad esempio, Windows utilizza la barra rovesciata (\) per fare riferimento al percorso della directory, mentre Linux e macOS utilizzano la barra (/).

2. Esegui il comando seguente, utilizzando il file che hai creato come input.

```
aws imagebuilder create-image-pipeline --cli-input-json file://create-image-
pipeline.json
```

Aggiorna le pipeline di immagini AMI dalla console

Dopo aver creato una pipeline di immagini Image Builder per l'immagine AMI, è possibile apportare modifiche alla configurazione dell'infrastruttura e alle impostazioni di distribuzione dalla console Image Builder.

Per aggiornare una pipeline di immagini con una nuova ricetta di immagini, è necessario utilizzare il. AWS CLI Per ulteriori informazioni sul tagging, consulta Aggiorna le pipeline di immagini AMI da AWS CLIin questa guida.

Scegli una pipeline Image Builder esistente

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/

2. Per visualizzare un elenco delle pipeline di immagini create con il tuo account, scegli Image pipelines dal pannello di navigazione.

> **Note**
>
> L'elenco delle pipeline di immagini include un indicatore per il tipo di immagine di output creata dalla pipeline: AMI o Docker.

3. Per visualizzare i dettagli o modificare una pipeline, scegliete il link al nome della pipeline. Questo apre la vista dettagliata della pipeline.

> **Note**
>
> Puoi anche selezionare la casella di controllo accanto al nome della tubazione, quindi scegliere Visualizza dettagli.

Dettagli della pipeline

La pagina dei dettagli della pipeline include le seguenti sezioni:

Riepilogo

La sezione nella parte superiore della pagina riassume i dettagli chiave della pipeline visibili con una qualsiasi delle schede di dettaglio aperte. I dettagli visualizzati in questa sezione sono modificabili solo nelle rispettive schede di dettaglio.

Schede dei dettagli

- Immagini di output: mostra le immagini di output prodotte dalla pipeline.
- Ricetta nell'immagine: mostra i dettagli della ricetta. Dopo aver creato una ricetta, non è possibile modificarla. È necessario creare una nuova versione della ricetta dalla pagina Ricette di immagini nella console Image Builder o utilizzando i comandi di Image Builder in. AWS CLI Per ulteriori informazioni, consulta Gestire le ricette in Image Builder.
- Configurazione dell'infrastruttura: mostra informazioni modificabili per configurare l'infrastruttura della pipeline di compilazione.
- Impostazioni di distribuzione: mostra informazioni modificabili per la distribuzione AMI.
- EventBridge regole — Per l'Event Bus selezionato, mostra EventBridge le regole che hanno come target la pipeline corrente. Include le azioni Create event bus e Create rule che si collegano

alla EventBridge console. Per ulteriori informazioni su questa scheda, consultaUsa le regole EventBridge .

Modifica la configurazione dell'infrastruttura per la tua pipeline

La configurazione dell'infrastruttura include i seguenti dettagli che è possibile modificare dopo aver creato la pipeline:

- La descrizione della configurazione dell'infrastruttura.
- Il ruolo IAM da associare al profilo dell'istanza.
- AWS infrastruttura, incluso il tipo di istanza e un argomento SNS per le notifiche.
- VPC, sottorete e gruppi di sicurezza.
- Impostazioni per la risoluzione dei problemi, tra cui Terminate Instance in caso di errore, la coppia di chiavi per la connessione e una posizione opzionale del bucket S3 per i log delle istanze.

Per modificare la configurazione dell'infrastruttura dalla pagina dei dettagli della pipeline, segui questi passaggi:

1. Scegli la scheda Configurazione dell'infrastruttura.
2. Scegli Modifica nell'angolo in alto a destra del pannello dei dettagli di configurazione.
3. Quando sei pronto per salvare gli aggiornamenti che hai apportato alla configurazione dell'infrastruttura, scegli Salva modifiche.

Modifica le impostazioni di distribuzione per la tua pipeline

Le impostazioni di distribuzione includono i seguenti dettagli che è possibile modificare dopo aver creato la pipeline:

- La descrizione della configurazione di distribuzione.
- Impostazioni regionali per le regioni in cui distribuisci l'immagine. Per impostazione predefinita, la regione 1 è la regione in cui è stata creata la pipeline. È possibile aggiungere regioni per la distribuzione con il pulsante Aggiungi regione e rimuovere tutte le regioni tranne la regione 1.

  Le impostazioni della regione includono:

  - Regione di destinazione

- Il nome dell'AMI di output
- Autorizzazioni di avvio e account con cui condividerle
- Licenze associate (configurazioni delle licenze associate)

> **Note**
>
> Le impostazioni del License Manager non verranno replicate tra le AWS regioni che devono essere abilitate nell'account, ad esempio tra le regioni `ap-east-1` (Hong Kong) e `me-south-1` (Bahrein).

Per modificare le impostazioni di distribuzione dalla pagina dei dettagli della pipeline, procedi nel seguente modo:

1. Scegli la scheda Impostazioni di distribuzione.
2. Scegli Modifica nell'angolo superiore destro del pannello dei dettagli di distribuzione.
3. Quando sei pronto per salvare gli aggiornamenti, scegli Salva modifiche.

Modifica la pianificazione di costruzione della tua pipeline

La pagina Modifica pipeline include i seguenti dettagli che è possibile modificare dopo aver creato la pipeline:

- La descrizione della pipeline.
- Raccolta di metadati migliorata. Questa opzione è attivata per impostazione predefinita. Per disattivarla, deselezionate la casella di controllo Abilita raccolta avanzata di metadati.
- La pianificazione di creazione per la tua pipeline. Puoi modificare le opzioni di pianificazione e tutte le impostazioni qui.

Per modificare la pipeline dalla pagina dei dettagli della pipeline, procedi nel seguente modo:

1. Nell'angolo in alto a destra della pagina dei dettagli della pipeline, scegli Azioni, quindi Modifica pipeline.
2. Quando sei pronto per salvare gli aggiornamenti, scegli Salva modifiche.

> **Note**
>
> Per ulteriori informazioni sulla pianificazione della build utilizzando le espressioni cron, consulta. Usa le espressioni cron in Image Builder

Aggiorna le pipeline di immagini AMI da AWS CLI

È possibile aggiornare una pipeline di immagini AMI utilizzando un file JSON come input per il update-image-pipeline comando in. AWS CLI Per configurare il file JSON, devi disporre di Amazon Resource Names (ARNs) per fare riferimento alle seguenti risorse esistenti:

- Pipeline di immagini da aggiornare
- Ricetta dell'immagine
- Configurazione dell'infrastruttura
- Distribution Settings (Impostazioni distribuzione)

È possibile aggiornare una pipeline di immagini AMI con il update-image-pipeline comando AWS CLI seguente:

> **Note**
>
> UpdateImagePipeline non supporta gli aggiornamenti selettivi per la pipeline. È necessario specificare tutte le proprietà richieste nella richiesta di aggiornamento, non solo le proprietà che sono state modificate.

1. Creazione di un file JSON di input nella CLI

   Usa il tuo strumento di modifica dei file preferito per creare un file JSON con le seguenti chiavi, oltre a valori validi per il tuo ambiente. In questo esempio viene utilizzato un file denominato `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**
>
> - È necessario includere l'annotazione `file://` all'inizio del percorso del file JSON.
> - Il percorso del file JSON deve seguire la convenzione appropriata per il sistema operativo di base in cui si esegue il comando. Ad esempio, Windows utilizza la barra rovesciata (\) per fare riferimento al percorso della directory, mentre Linux e macOS utilizzano la barra (/).

2. Esegui il comando seguente utilizzando il file creato come input.

```
aws imagebuilder update-image-pipeline --cli-input-json file://update-image-
pipeline.json
```

Crea e aggiorna le pipeline di immagini dei contenitori

È possibile impostare, configurare e gestire le pipeline di immagini dei contenitori utilizzando la console Image Builder, tramite l'API Image Builder o imagebuilder con i comandi in. AWS CLI La procedura guidata della console Create image pipeline fornisce gli elementi iniziali e guida l'utente attraverso i passaggi per:

- Seleziona un'immagine di base dai repository Quick-Start Managed Images, Amazon ECR o Docker Hub
- Aggiungi e rimuovi software
- Personalizza impostazioni e script
- Esegui test selezionati
- Crea un Dockerfile utilizzando variabili di compilazione preconfigurate.
- AWS Distribuisci immagini nelle regioni

Per ulteriori informazioni e un step-by-step tutorial sull'utilizzo della procedura guidata della console Create image pipeline, consulta. Tutorial: crea una pipeline di immagini con l'immagine del contenitore Docker in uscita dalla procedura guidata della console Image Builder

Indice

- Crea una pipeline di immagini di un contenitore da AWS CLI
- Aggiorna una pipeline di immagini del contenitore dalla console
- Aggiorna le pipeline delle immagini dei contenitori da AWS CLI

Crea una pipeline di immagini di un contenitore da AWS CLI

È possibile creare una pipeline di immagini del contenitore utilizzando un file JSON come input per il comando in. create-image-pipeline AWS CLI

La frequenza con cui la pipeline crea una nuova immagine per incorporare eventuali aggiornamenti in sospeso dell'immagine e dei componenti di base dipende dalla `schedule` configurazione. A `schedule` ha i seguenti attributi:

- `scheduleExpression`— Imposta la pianificazione dell'esecuzione della pipeline per valutare `pipelineExecutionStartCondition` e determinare se avviare una build. La pianificazione è configurata con espressioni cron. Per ulteriori informazioni su come formattare un'espressione cron in Image Builder, vedere. Usa le espressioni cron in Image Builder
- `pipelineExecutionStartCondition`— Determina se la pipeline deve avviare la compilazione. I valori validi includono:
  - `EXPRESSION_MATCH_ONLY`— la pipeline creerà una nuova immagine ogni volta che l'espressione cron corrisponde all'ora corrente.

- `EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE`— la pipeline non avvierà una nuova creazione di immagini a meno che non vi siano modifiche in sospeso all'immagine o ai componenti di base.

Quando si esegue il create-image-pipeline comando in AWS CLI, molte risorse di configurazione sono opzionali. Tuttavia, alcune risorse hanno requisiti condizionali, a seconda del tipo di immagine creata dalla pipeline. Le seguenti risorse sono necessarie per le pipeline di immagini dei contenitori:

- Ricetta contenitore ARN
- Configurazione dell'infrastruttura ARN

Se non includi una risorsa di configurazione della distribuzione quando esegui il create-image-pipeline comando, l'immagine di output viene archiviata nell'archivio ECR specificato come repository di destinazione nella ricetta del contenitore nella regione in cui esegui il comando. Se includi una risorsa di configurazione della distribuzione per la tua pipeline, viene utilizzato il repository di destinazione che hai specificato per la prima regione della distribuzione.

1. Creazione di un file JSON di input nella CLI

   Usa il tuo strumento di modifica dei file preferito per creare un file JSON con le seguenti chiavi, oltre a valori validi per il tuo ambiente. In questo esempio viene utilizzato un file denominato `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**
>
> - È necessario includere l'annotazione `file://` all'inizio del percorso del file JSON.
> - Il percorso del file JSON deve seguire la convenzione appropriata per il sistema operativo di base in cui si esegue il comando. Ad esempio, Windows utilizza la barra rovesciata (\) per fare riferimento al percorso della directory, mentre Linux e macOS utilizzano la barra (/).

2. Esegui il comando seguente utilizzando il file creato come input.

```
aws imagebuilder create-image-pipeline --cli-input-json file://create-image-
pipeline.json
```

Aggiorna una pipeline di immagini del contenitore dalla console

Dopo aver creato una pipeline di immagini del contenitore Image Builder per l'immagine Docker, puoi apportare modifiche alla configurazione dell'infrastruttura e alle impostazioni di distribuzione dalla console Image Builder.

Per aggiornare una pipeline di immagini del contenitore con una nuova ricetta di contenitore, è necessario utilizzare il. AWS CLI Per ulteriori informazioni sul tagging, consulta Aggiorna le pipeline delle immagini dei contenitori da AWS CLIin questa guida.

Scegli una pipeline di immagini Docker di Image Builder esistente

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Per visualizzare un elenco delle pipeline di immagini create con il tuo account, scegli Image pipelines dal pannello di navigazione.

> **Note**
>
> L'elenco delle pipeline di immagini include un indicatore per il tipo di immagine di output creata dalla pipeline: AMI o Docker.

3. Per visualizzare i dettagli o modificare una pipeline, scegliete il link al nome della pipeline. Questo apre la vista dettagliata della pipeline.

> **Note**
>
> Puoi anche selezionare la casella di controllo accanto al nome della tubazione, quindi scegliere Visualizza dettagli.

Dettagli della pipeline

La pagina dei dettagli della pipeline di EC2 Image Builder include le seguenti sezioni:

Riepilogo

La sezione nella parte superiore della pagina riassume i dettagli chiave della pipeline che sono visibili con una qualsiasi delle schede di dettaglio aperte. I dettagli visualizzati in questa sezione sono modificabili solo nelle rispettive schede di dettaglio.

Schede dei dettagli

- Immagini di output: mostra le immagini di output prodotte dalla pipeline.
- Ricetta del contenitore: mostra i dettagli della ricetta. Dopo aver creato una ricetta, non è possibile modificarla. È necessario creare una nuova versione della ricetta dalla pagina Container recipes. Per ulteriori informazioni, consulta Crea una nuova versione di una ricetta contenitore.
- Configurazione dell'infrastruttura: mostra informazioni modificabili per configurare l'infrastruttura di build pipeline.
- Impostazioni di distribuzione: mostra informazioni modificabili per la distribuzione delle immagini Docker.
- EventBridge regole — Per l'Event Bus selezionato, mostra EventBridge le regole che hanno come target la pipeline corrente. Include le azioni Create event bus e Create rule che si collegano alla EventBridge console. Per ulteriori informazioni su questa scheda, consultaUsa le regole EventBridge .

Modifica la configurazione dell'infrastruttura per la tua pipeline

La configurazione dell'infrastruttura include i seguenti dettagli che è possibile modificare dopo aver creato la pipeline:

- La descrizione della configurazione dell'infrastruttura.
- Il ruolo IAM da associare al profilo dell'istanza.
- AWS infrastruttura, incluso il tipo di istanza e un argomento SNS per le notifiche.
- VPC, sottorete e gruppi di sicurezza.
- Impostazioni per la risoluzione dei problemi, tra cui Terminate Instance in caso di errore, la coppia di chiavi per la connessione e una posizione opzionale del bucket S3 per i log delle istanze.

Per modificare la configurazione dell'infrastruttura dalla pagina dei dettagli della pipeline, procedi nel seguente modo:

1. Scegli la scheda Configurazione dell'infrastruttura.
2. Scegli Modifica nell'angolo in alto a destra del pannello dei dettagli di configurazione.
3. Quando sei pronto a salvare gli aggiornamenti che hai apportato alla configurazione dell'infrastruttura, scegli Salva modifiche.

Modifica le impostazioni di distribuzione per la tua pipeline

Le impostazioni di distribuzione includono i seguenti dettagli che è possibile modificare dopo aver creato la pipeline:

- La descrizione delle impostazioni di distribuzione.
- Impostazioni regionali per le regioni in cui distribuisci l'immagine. Per impostazione predefinita, la regione 1 è la regione in cui è stata creata la pipeline. È possibile aggiungere regioni per la distribuzione con il pulsante Aggiungi regione e rimuovere tutte le regioni tranne la regione 1.

  Le impostazioni della regione includono:

  - Regione di destinazione
  - L'impostazione predefinita del servizio è «ECR» e non è modificabile.
  - Nome del repository: il nome del repository di destinazione (esclusa la posizione Amazon ECR). Ad esempio, il nome del repository con la posizione avrebbe lo schema seguente:

```
<account-id>.dkr.ecr.<region>.amazonaws.com/<repository-name>
```

> **Note**
>
> Se modificate il nome del Repository, solo le immagini create dopo la modifica del nome verranno aggiunte con il nuovo nome. Tutte le immagini precedenti create dalla pipeline rimangono nel loro repository originale.

Per modificare le impostazioni di distribuzione dalla pagina dei dettagli della pipeline, procedi nel seguente modo:

1. Scegli la scheda Impostazioni di distribuzione.
2. Scegli Modifica nell'angolo superiore destro del pannello dei dettagli di distribuzione.
3. Quando sei pronto per salvare gli aggiornamenti che hai apportato alle impostazioni di distribuzione, scegli Salva modifiche.

Modifica la pianificazione di costruzione della tua pipeline

La pagina Modifica pipeline include i seguenti dettagli che è possibile modificare dopo aver creato la pipeline:

- La descrizione della pipeline.
- Raccolta di metadati migliorata. Questa opzione è attivata per impostazione predefinita. Per disattivarla, deselezionate la casella di controllo Abilita raccolta avanzata di metadati.
- La pianificazione di creazione per la tua pipeline. Puoi modificare le opzioni di pianificazione e tutte le impostazioni in questa sezione.

Per modificare la pipeline dalla pagina dei dettagli della pipeline, procedi nel seguente modo:

1. Nell'angolo in alto a destra della pagina dei dettagli della pipeline, scegli Azioni, quindi Modifica pipeline.
2. Quando sei pronto per salvare gli aggiornamenti, scegli Salva modifiche.

> **Note**
>
> Per ulteriori informazioni sulla pianificazione della build utilizzando le espressioni cron, consulta. Usa le espressioni cron in Image Builder

Aggiorna le pipeline delle immagini dei contenitori da AWS CLI

È possibile aggiornare una pipeline di immagini del contenitore utilizzando un file JSON come input per il comando in. update-image-pipeline AWS CLI Per configurare il file JSON, devi disporre di Amazon Resource Names (ARNs) per fare riferimento alle seguenti risorse esistenti:

- Pipeline di immagini da aggiornare
- Ricetta del contenitore
- Configurazione dell'infrastruttura
- Impostazioni di distribuzione (se incluse nella pipeline corrente)

> **Note**
>
> Se la risorsa delle impostazioni di distribuzione è inclusa, l'archivio ECR specificato come repository di destinazione nelle impostazioni di distribuzione per la regione in cui viene eseguito il comando (Regione 1) ha la precedenza sul repository di destinazione specificato nella ricetta del contenitore.

Segui questi passaggi per aggiornare una pipeline di immagini del contenitore utilizzando il comando contenuto in: update-image-pipeline AWS CLI

> **Note**
>
> UpdateImagePipeline non supporta gli aggiornamenti selettivi per la pipeline. È necessario specificare tutte le proprietà richieste nella richiesta di aggiornamento, non solo le proprietà che sono state modificate.

1. Creazione di un file JSON di input nella CLI

   Usa il tuo strumento di modifica dei file preferito per creare un file JSON con le seguenti chiavi, oltre a valori validi per il tuo ambiente. In questo esempio viene utilizzato un file denominato `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**
   >
   > - È necessario includere l'annotazione `file://` all'inizio del percorso del file JSON.
   > - Il percorso del file JSON deve seguire la convenzione appropriata per il sistema operativo di base in cui si esegue il comando. Ad esempio, Windows utilizza la barra rovesciata (\) per fare riferimento al percorso della directory, mentre Linux e macOS utilizzano la barra (/).

2. Esegui il comando seguente utilizzando il file creato come input.

```
aws imagebuilder update-image-pipeline --cli-input-json file://update-image-pipeline.json
```

Configurazione dei flussi di lavoro della pipeline di immagini in Image Builder

Con i flussi di lavoro basati su immagini, puoi personalizzare i flussi di lavoro eseguiti dalla tua pipeline per creare e testare le immagini in base alle tue esigenze. I flussi di lavoro definiti vengono eseguiti nel contesto del framework del flusso di lavoro Image Builder. Per ulteriori informazioni sulle fasi che compongono il framework del flusso di lavoro, vedere. Gestisci i flussi di lavoro di creazione e test per le immagini di Image Builder

Crea un flusso di lavoro

Crea flussi di lavoro eseguiti durante la `Build` fase del framework del flusso di lavoro. Puoi specificare un solo flusso di lavoro di compilazione per la tua pipeline. Oppure puoi saltare completamente la compilazione per configurare una pipeline di solo test.

Workflow di test

I flussi di lavoro di test vengono eseguiti durante la `Test` fase del framework del flusso di lavoro. Puoi specificare fino a dieci flussi di lavoro di test per la tua pipeline. Puoi anche saltare completamente i test se desideri solo che la pipeline venga creata.

Definisci gruppi di test per i flussi di lavoro di test

I flussi di lavoro di test sono definiti all'interno di gruppi di test. Puoi eseguire fino a dieci flussi di lavoro di test per la tua pipeline. Sei tu a decidere se eseguire i flussi di lavoro di test in un ordine specifico o eseguirne il maggior numero possibile contemporaneamente. Il modo in cui vengono eseguiti dipende da come vengono definiti i gruppi di test. I seguenti scenari illustrano diversi modi in cui è possibile definire i flussi di lavoro di test.

> **Note**
>
> Se utilizzi la console per creare flussi di lavoro, ti consigliamo di dedicare del tempo alla pianificazione del modo in cui desideri eseguire i flussi di lavoro di test prima di definire i

gruppi di test. Nella console, puoi aggiungere o rimuovere flussi di lavoro e gruppi di test, ma non puoi riordinarli.

Scenario 1: Esegui un flusso di lavoro di test alla volta

Per eseguire tutti i flussi di lavoro di test uno alla volta, puoi configurare fino a dieci gruppi di test, ciascuno con un unico flusso di lavoro di test. I gruppi di test vengono eseguiti uno alla volta, nell'ordine in cui vengono aggiunti alla pipeline. Questo è un modo per garantire che i flussi di lavoro di test vengano eseguiti uno alla volta in un ordine specifico.

Scenario 2: Esegui più flussi di lavoro di test contemporaneamente

Se l'ordine non è importante e desideri eseguire il maggior numero possibile di flussi di lavoro di test contemporaneamente, puoi configurare un singolo gruppo di test e inserire il numero massimo di flussi di lavoro di test. Image Builder avvia fino a cinque flussi di lavoro di test contemporaneamente e avvia flussi di lavoro di test aggiuntivi man mano che gli altri vengono completati. Se il vostro obiettivo è quello di eseguire i flussi di lavoro di test il più velocemente possibile, questo è un modo per farlo.

Scenario 3: Mescola e abbina

Se hai uno scenario misto, con alcuni flussi di lavoro di test che possono essere eseguiti contemporaneamente e altri che dovrebbero essere eseguiti uno alla volta, puoi configurare i gruppi di test per raggiungere questo obiettivo. L'unico limite alla configurazione dei gruppi di test è il numero massimo di flussi di lavoro di test che è possibile eseguire per la pipeline

Impostare i parametri del flusso di lavoro in una pipeline di Image Builder dalla console

I parametri del flusso di lavoro funzionano allo stesso modo per i flussi di lavoro di creazione e i flussi di lavoro di test. Quando crei o aggiorni una pipeline, selezioni i flussi di lavoro di creazione e test che desideri includere. Se nel documento del workflow sono stati definiti dei parametri per un workflow selezionato, Image Builder li visualizza nel pannello Parametri. Il pannello è nascosto per i flussi di lavoro che non hanno parametri definiti.

Ogni parametro mostra i seguenti attributi definiti dal documento del flusso di lavoro:

- Nome (non modificabile): il nome del parametro.
- Tipo (non modificabile): il tipo di dati per il valore del parametro.

- Valore: il valore del parametro. È possibile modificare il valore del parametro per impostarlo per la pipeline.

Specificare il ruolo del servizio IAM utilizzato da Image Builder per eseguire le azioni del flusso di lavoro

Per eseguire flussi di lavoro di immagini, Image Builder necessita dell'autorizzazione per eseguire azioni del flusso di lavoro. È possibile specificare il ruolo AWSServiceRoleForImageBuildercollegato al servizio oppure specificare il proprio ruolo personalizzato per l'accesso al servizio, come segue.

- Console: nella procedura guidata alla pipeline, fase 3, definisci il processo di creazione dell'immagine, seleziona il ruolo collegato al servizio o il tuo ruolo personalizzato dall'elenco dei ruoli IAM nel pannello di accesso al servizio.
- API Image Builder: nella richiesta di CreateImageazione, specifica il ruolo collegato al servizio o il tuo ruolo personalizzato come valore per il parametro. `executionRole`

Per ulteriori informazioni su come creare un ruolo di servizio, consulta Creazione di un ruolo per delegare le autorizzazioni a un AWS servizio nella Guida per l'utente.AWS Identity and Access Management

Esegui la tua pipeline di immagini

Se hai scelto l'opzione di pianificazione manuale per la tua pipeline, questa verrà eseguita solo quando avvierai manualmente la build. Se hai scelto una delle opzioni di pianificazione automatica, puoi anche eseguirla manualmente, tra le esecuzioni pianificate regolarmente. Ad esempio, se disponi di una pipeline che normalmente viene eseguita una volta al mese, ma devi incorporare un aggiornamento a uno dei componenti due settimane dopo l'esecuzione precedente, puoi scegliere di eseguire la pipeline manualmente.

Console

Per eseguire la pipeline dalla pagina dei dettagli della pipeline nella console Image Builder, scegliete Esegui pipeline dal menu Azioni nella parte superiore della pagina. Nella parte superiore della pagina viene visualizzato un messaggio di stato per avvisarti che la pipeline è iniziata o se c'è un errore.

1. Nell'angolo in alto a sinistra della pagina dei dettagli della pipeline, scegli Esegui pipeline dal menu Azioni.
2. Puoi vedere lo stato attuale della pipeline nella scheda Immagini di output, nella colonna Stato.

AWS CLI

L'esempio seguente mostra come utilizzare il start-image-pipeline-execution comando in AWS CLI per avviare manualmente una pipeline di immagini. Quando si esegue questo comando, la pipeline crea e distribuisce una nuova immagine.

```
aws imagebuilder start-image-pipeline-execution --image-pipeline-arn
 arn:aws:imagebuilder:us-west-2:111122223333:image-pipeline/my-example-pipeline
```

Per vedere quali risorse vengono create durante l'esecuzione della pipeline di compilazione, consulta. Risorse create

Usa le espressioni cron in Image Builder

Usa le espressioni cron per EC2 Image Builder per impostare una finestra temporale per aggiornare l'immagine con aggiornamenti che si applicano all'immagine di base e ai componenti della pipeline. La finestra temporale per l'aggiornamento della pipeline inizia con l'ora impostata nell'espressione cron. Puoi impostare l'ora nell'espressione cron fino al minuto. La build della pipeline può essere eseguita a partire dall'ora di inizio o dopo.

A volte possono essere necessari alcuni secondi o fino a un minuto prima che la build inizi a funzionare.

> **Note**
>
> Le espressioni Cron utilizzano il fuso orario UTC (Universal Coordinated Time) per impostazione predefinita, oppure è possibile specificare il fuso orario. Per ulteriori informazioni sull'ora UTC e per trovare lo scostamento del fuso orario, vedere Abbreviazioni dei fusi orari — Elenco mondiale.

Valori supportati per le espressioni cron in Image Builder

EC2 Image Builder utilizza un formato cron composto da sei campi obbligatori. Ciascuno è separato dagli altri da uno spazio intermedio, senza spazi iniziali o finali:

<Minute> <Hour> <Day> <Month> <Day of the week> <Year>

La tabella seguente mostra i valori supportati per le voci Cron richieste.

Valori supportati per le espressioni Cron

Campo	Valori	Caratteri jolly
Minuto	`0-59`	, - * /
Ora	`0-23`	, - * /
Day (Giorno)	`1-31`	`, - * ? / L W`
Mese	`1-12` o `jan-dec`	`, - * /`
Giorno della settimana	`1-7` o `sun-sat`	`, - * ? L #`
Anno	`1970-2199`	`, - * /`

Caratteri jolly

La tabella seguente descrive come Image Builder utilizza i caratteri jolly per le espressioni cron. Tieni presente che l'avvio della compilazione può richiedere fino a un minuto dopo il tempo specificato.

Caratteri jolly supportati per le espressioni Cron

Carattere jolly	Descrizione
,	Il carattere jolly , (virgola) include valori aggiuntivi. Nel campo Mese, `jan,feb,mar` include gennaio, febbraio e marzo.
-	Il carattere jolly - (trattino) specifica gli intervalli. Nel campo giorno del mese, `1-15` include i giorni da 1 a 15 del mese specificato.

Carattere jolly	Descrizione
*	Il carattere jolly * (asterisco) include tutti i valori validi per il campo.
?	Il carattere jolly ? (punto interrogativo) specifica che il valore del campo dipende da un'altra impostazione. Nel caso del Giorno e dei Day-of-week campi, quando uno è specificato o include tutti i valori possibili (*), l'altro deve essere un. ? Non è possibile specificare entrambi. Ad esempio, se si immette a 7 nel campo Giorno (si esegue la build il settimo giorno del mese), la Day-of-week posizione deve contenere un?.
/	Il carattere jolly / (barra) specifica gli increment i. Ad esempio, se desideri che la build venga eseguita a giorni alterni, inserisci */2 nel campo giorno.
L	Il carattere jolly L in uno dei campi relativi al giorno specifica l'ultimo giorno: 28-31 per il giorno del mese, a seconda del mese, o domenica, per il giorno della settimana.
W	Il carattere jolly W nel campo Day-of-month specifica un giorno feriale. Nel Day-of-month campo, se inserisci un numero precedente alW, significa che vuoi scegliere come target il giorno della settimana più vicino a quel giorno. Ad esempio, se lo specifichi3W, desideri che la build venga eseguita il giorno della settimana più vicino al terzo giorno del mese.

Carattere jolly	Descrizione
#	Il # (hash) è consentito solo per il campo del giorno della settimana e deve essere seguito da un numero compreso tra 1 e 5. Il numero specifica in quali settimane di un determina to mese valgono per l'esecuzione della build. Ad esempio, se desideri che la build venga eseguita il secondo venerdì di ogni mese, utilizzala `fri#2` per il campo giorno della settimana.

Restrizioni

- Non puoi specificare i campi Day-of-month e Day-of-week nella stessa espressione cron. Se specifichi un valore o `*` in uno di questi campi, devi utilizzare un `?` nell'altro.
- Le espressioni Cron che indicano frequenze più rapide di un minuto non sono supportate.

Esempi di espressioni cron in Image Builder

Le espressioni Cron vengono inserite in modo diverso per la console Image Builder rispetto all'API o alla CLI. Per vedere degli esempi, scegli la scheda che fa per te.

Image Builder console

Gli esempi seguenti mostrano le espressioni cron che puoi inserire nella console per la pianificazione della compilazione. L'ora UTC viene specificata utilizzando un orologio a 24 ore.

Esegui ogni giorno alle 10:00 (UTC)

```
0 10 * * ? *
```

Corri ogni giorno alle 12:15 (UTC)

```
15 12 * * ? *
```

Corri ogni giorno a mezzanotte (UTC)

```
0 0 * * ? *
```

Corri alle 10:00 (UTC) ogni mattina nei giorni feriali

```
0 10 ? * 2-6 *
```

Esegui alle 18:00 (UTC) ogni sera nei giorni feriali

```
0 18 ? * mon-fri *
```

Esegui alle 8:00 (UTC) il primo giorno di ogni mese

```
0 8 1 * ? *
```

Esegui il secondo martedì di ogni mese alle 22:30 (UTC)

```
30 22 ? * tue#2 *
```

> **Tip**
>
> Se non vuoi che il processo di pipeline si protragga fino al giorno successivo mentre è in esecuzione, assicurati di tenere conto del tempo di compilazione quando specifichi l'ora di inizio.

API/CLI

Gli esempi seguenti mostrano le espressioni cron che è possibile inserire per la pianificazione della compilazione utilizzando i comandi CLI o le richieste API. Viene mostrata solo l'espressione cron.

Esegui ogni giorno alle 10:00 (UTC)

```
cron(0 10 * * ? *)
```

Corri ogni giorno alle 12:15 (UTC)

```
cron(15 12 * * ? *)
```

Corri ogni giorno a mezzanotte (UTC)

```
cron(0 0 * * ? *)
```

Corri alle 10:00 (UTC) ogni mattina nei giorni feriali

`cron(0 10 ? * 2-6 *)`

Esegui alle 18:00 (UTC) ogni sera nei giorni feriali

`cron(0 18 ? * mon-fri *)`

Esegui alle 8:00 (UTC) il primo giorno di ogni mese

`cron(0 8 1 * ? *)`

Esegui il secondo martedì di ogni mese alle 22:30 (UTC)

`cron(30 22 ? * tue#2 *)`

> **Tip**
>
> Se non vuoi che il processo di pipeline si protragga fino al giorno successivo mentre è in esecuzione, assicurati di tenere conto del tempo di compilazione quando specifichi l'ora di inizio.

Valuta le espressioni in Image Builder

Un espressione rate inizia quando crei la regola di evento pianificato e successivamente la esegui nella relativa pianificazione definita.

Le espressioni rate hanno due campi obbligatori. I campi sono separati da uno spazio vuoto.

Sintassi

```
rate(value unit)
```

value

Un numero positivo.

unità

L'unità di tempo. Per i valori di 1, ad esempio `minute`, e i valori maggiori di 1, ad esempio `minutes`, sono necessarie unità diverse.

Valori validi: minuto | minuti | ora | ore | giorno | giorni

Restrizioni

Se il valore è uguale a 1, l'unità deve essere singolare. Analogamente, per valori maggiori di 1, l'unità deve essere plurale. Ad esempio, `rate(1 hours)` e `rate(5 hour)` non sono validi, mentre lo sono `rate(1 hour)` e `rate(5 hours)`.

Usa EventBridge le regole con le pipeline di Image Builder

Gli eventi di un'ampia gamma di AWS servizi partner vengono trasmessi in streaming ai bus di EventBridge eventi di Amazon quasi in tempo reale. Puoi anche generare eventi personalizzati e inviare eventi dalle tue applicazioni a EventBridge. I bus degli eventi utilizzano regole per determinare dove indirizzare i dati degli eventi.

Le pipeline di Image Builder sono disponibili come obiettivi delle EventBridge regole, il che significa che è possibile eseguire una pipeline di Image Builder in base a regole create per rispondere agli eventi sul bus o in base a una pianificazione.

Per un riepilogo degli eventi generati dal sistema a cui viene inviato Image Builder EventBridge, vedere. Messaggi di eventi inviati da Image Builder

> **Note**
>
> I bus degli eventi sono specifici di una regione. La regola e l'obiettivo devono trovarsi nella stessa regione.

Indice

- EventBridge termini
- Visualizza EventBridge le regole per la tua pipeline di Image Builder
- Usa EventBridge le regole per pianificare la costruzione di una pipeline

EventBridge termini

Questa sezione contiene un riepilogo dei termini per aiutarti a capire come si EventBridge integra con le tue pipeline di Image Builder.

Evento

Descrive una modifica in un ambiente che potrebbe influire su una o più risorse dell'applicazione. L'ambiente può essere un AWS ambiente, un servizio o un'applicazione partner SaaS o una delle tue applicazioni o servizi. Puoi anche impostare eventi programmati su una sequenza temporale.

Router di eventi

Una pipeline che riceve dati sugli eventi da applicazioni e servizi.

Origine

Il servizio o l'applicazione che ha inviato l'evento al bus degli eventi.

Target

Una risorsa o un endpoint che EventBridge richiama quando corrisponde a una regola, inviando i dati dall'evento alla destinazione.

Regola

Una regola trova gli eventi in entrata corrispondenti e li invia alle destinazioni per l'elaborazione. Una singola regola può inviare un evento a più destinazioni, che possono quindi essere eseguite in parallelo. Le regole si basano su un modello di eventi o su una pianificazione.

Pattern

Un modello di evento definisce la struttura dell'evento e i campi a cui una regola corrisponde per avviare l'azione di destinazione.

Pianificazione

Le regole di pianificazione eseguono un'azione in base a una pianificazione, ad esempio l'esecuzione di una pipeline di Image Builder per aggiornare un'immagine su base trimestrale. Esistono due tipi di espressioni di pianificazione:

- Espressioni cron: soddisfano criteri di pianificazione specifici utilizzando la sintassi cron che può delineare criteri semplici; ad esempio, vengono eseguite settimanalmente in un giorno specifico. Puoi anche stabilire criteri più complessi, come l'esecuzione trimestrale il quinto giorno del mese, tra le 2:00 e le 4:00.
- Espressioni di frequenza: specifica un intervallo regolare quando viene richiamato l'obiettivo, ad esempio ogni 12 ore.

Visualizza EventBridge le regole per la tua pipeline di Image Builder

La scheda EventBridge regole nella pagina dei dettagli delle pipeline di immagini di Image Builder mostra i bus di EventBridge eventi a cui l'account ha accesso e le regole per il bus di eventi selezionato che si applicano alla pipeline corrente. Questa scheda si collega anche direttamente alla EventBridge console per la creazione di nuove risorse.

Azioni che si collegano alla EventBridge console

- Crea un bus per eventi
- Creare una regola

Per ulteriori informazioni EventBridge, consulta i seguenti argomenti nella Amazon EventBridge User Guide.

- Che cos'è Amazon EventBridge
- Autobus per EventBridge eventi Amazon
- EventBridge Eventi Amazon
- EventBridge Regole di Amazon

Usa EventBridge le regole per pianificare la costruzione di una pipeline

Per questo esempio, creiamo una nuova regola di pianificazione per il bus degli eventi predefinito, utilizzando un'espressione di velocità. La regola in questo esempio genera un evento sul bus degli eventi ogni 90 giorni. L'evento avvia una creazione di pipeline per aggiornare l'immagine.

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Per visualizzare un elenco delle pipeline di immagini create con il tuo account, scegli Image pipelines dal pannello di navigazione.

   > **Note**
   >
   > L'elenco delle pipeline di immagini include un indicatore per il tipo di immagine di output creata dalla pipeline: AMI o Docker.

3. Per visualizzare i dettagli o modificare una pipeline, scegliete il link al nome della pipeline. Questo apre la vista dettagliata della pipeline.

> **Note**
>
> Puoi anche selezionare la casella di controllo accanto al nome della tubazione, quindi scegliere Visualizza dettagli.

4. Apri la scheda EventBridge delle regole.
5. Mantieni il bus degli eventi predefinito preselezionato nel pannello Event Bus.
6. Scegli Crea regola. Verrai indirizzato alla pagina Crea regola nella EventBridge console Amazon.
7. Inserire un nome e una descrizione per la regola. Il nome della regola deve essere univoco all'interno del bus degli eventi per la regione selezionata.
8. Nel pannello Definisci modello, scegliete l'opzione Pianificazione. Questo espande il pannello, con l'opzione Tariffa fissa selezionata per ogni opzione.
9. Inserisci `90` nella prima casella e seleziona Giorni dall'elenco a discesa.
10. Esegui le seguenti azioni nel pannello Seleziona obiettivi:
    a. Effettua una selezione `EC2 Image Builder` dall'elenco a discesa Target.
    b. Per applicare la regola a una pipeline Image Builder, selezionate la pipeline di destinazione dall'elenco a discesa Image Pipeline.
    c. EventBridge necessita dell'autorizzazione per avviare una compilazione per la pipeline selezionata. Per questo esempio, mantieni l'opzione predefinita Create a new role per questa risorsa specifica.
    d. Scegliere Add target (Aggiungi target).
11. Scegli Crea

> **Note**
>
> Per ulteriori informazioni sulle impostazioni per le regole di espressione delle tariffe non trattate in questo esempio, consulta Rate expression nella Amazon EventBridge User Guide.

Integra prodotti e servizi in Image Builder

EC2 Image Builder si integra con Marketplace AWS altre applicazioni per aiutarti a creare immagini di macchine personalizzate robuste Servizi AWS e sicure.

Prodotti

Le ricette di Image Builder possono incorporare prodotti di immagini Marketplace AWS e componenti gestiti da Image Builder per fornire funzionalità di compilazione e test specializzate, come segue.

- Marketplace AWS prodotti di immagini: utilizzate un prodotto di immagine Marketplace AWS come immagine di base nella ricetta per soddisfare gli standard organizzativi, come CIS Hardening. Quando crei una ricetta dalla console Image Builder, puoi scegliere tra gli abbonamenti esistenti o cercare un prodotto specifico da. Marketplace AWS Quando crei una ricetta dall'API Image Builder, dalla CLI o dall'SDK, puoi specificare un prodotto di immagine Amazon Resource Name (ARN) da utilizzare come immagine di base.
- Componenti Image Builder: i componenti specificati nelle ricette possono eseguire azioni di compilazione e test, ad esempio per installare software o eseguire la convalida della conformità. Alcuni prodotti di immagini a cui ti abboni Marketplace AWS potrebbero includere un componente complementare che puoi utilizzare nelle tue ricette. Le immagini CIS Hardened includono un AWSTOE componente corrispondente che puoi utilizzare nella tua ricetta per applicare le linee guida CIS Benchmarks Level 1 per la tua configurazione.

> **Note**
>
> Per ulteriori informazioni sui prodotti correlati alla conformità, consulta. Prodotti di conformità per le immagini di Image Builder

Servizi

Image Builder si integra con quanto segue Servizi AWS per fornire metriche dettagliate degli eventi, registrazione e monitoraggio. Queste informazioni consentono di tenere traccia delle attività, risolvere i problemi di creazione delle immagini e creare automazioni basate sulle notifiche degli eventi.

- AWS Organizations— AWS Organizations consente di applicare le politiche di controllo dei servizi (SCP) agli account dell'organizzazione. È possibile creare, gestire, abilitare e disabilitare

singole politiche. Analogamente a tutti gli altri AWS artefatti e servizi, Image Builder rispetta le politiche definite in. AWS Organizations AWS fornisce un modello SCPs per scenari comuni, come l'imposizione di vincoli sugli account dei membri per avviare istanze solo approvate. AMIs

- AWS CloudTrail— Monitora gli eventi di Image Builder inviati a. CloudTrail Per ulteriori informazioni sull' CloudTrail integrazione con Image Builder, vedere. Registra le chiamate API Image Builder utilizzando CloudTrail

  Per ulteriori informazioni CloudTrail, incluso come attivarlo e trovare i file di registro, consulta la Guida per l'AWS CloudTrail utente.

- Amazon CloudWatch Logs: monitora, archivia e accedi ai file di registro di Image Builder con. CloudWatch Facoltativamente, puoi salvare i log in un bucket S3. Per ulteriori informazioni sull' CloudWatch integrazione con Image Builder, vedere. Monitora i log di Image Builder con Amazon Logs CloudWatch

  Per ulteriori informazioni sui CloudWatch log, consulta What is Amazon CloudWatch Logs? nella Amazon CloudWatch Logs User Guide.

- Amazon Elastic Container Registry (Amazon ECR) — Amazon ECR è AWS un servizio di registro di immagini di container gestito sicuro, scalabile e affidabile. Le immagini dei container create con Image Builder vengono archiviate in Amazon ECR nella regione di origine (dove viene eseguita la build) e in tutte le regioni in cui distribuisci l'immagine del contenitore. Per ulteriori informazioni su Amazon ECR, consulta la Amazon Elastic Container Registry User Guide.

- Amazon EventBridge: connettiti a un flusso di dati sugli eventi in tempo reale dalle attività di Image Builder nel tuo account. Per ulteriori informazioni su EventBridge, consulta What Is Amazon EventBridge? nella Amazon EventBridge User Guide.

- Amazon Inspector: scopri le vulnerabilità del tuo software e delle impostazioni di rete con scansioni automatiche dell'istanza di EC2 test lanciata da Image Builder per creare una nuova immagine. Image Builder salva i risultati per la risorsa di immagini di output in modo da poter analizzare e correggere dopo la chiusura dell'istanza di test. Per ulteriori informazioni sulle scansioni e sui prezzi, consulta Cos'è Amazon Inspector? nella Guida per l'utente di Amazon Inspector.

  Amazon Inspector può anche scansionare i tuoi repository ECR se configuri una scansione avanzata. Per ulteriori informazioni, consulta Scansione delle immagini dei container Amazon ECR nella Guida per l'utente di Amazon Inspector.

> **Note**
>
> Amazon Inspector è una funzionalità a pagamento.

- AWS License Manager— È possibile allegare una licenza autogestita di License Manager a un'AMI di output durante il processo di distribuzione. La licenza specificata per la regione di destinazione deve già esistere in quella regione. Per ulteriori informazioni sulle licenze autogestite, consulta Licenze autogestite in License Manager.
- Marketplace AWS— Visualizza un elenco dei tuoi attuali abbonamenti ai Marketplace AWS prodotti e cerca prodotti illustrati direttamente da Image Builder. Puoi anche utilizzare un prodotto di immagine a cui ti sei abbonato come immagine di base per una ricetta di Image Builder. Per ulteriori informazioni sulla gestione degli Marketplace AWS abbonamenti, consulta la sezione Acquisto di prodotti nella Guida all'Marketplace AWS acquisto.
- AWS Resource Access Manager (AWS RAM) — Con AWS RAM, puoi condividere risorse con chiunque Account AWS o tramite AWS Organizations. Se ne hai più Account AWS, puoi creare risorse centralmente e AWS RAM utilizzarle per condividerle con altri account. EC2 Image Builder consente la condivisione delle seguenti risorse: componenti, immagini e ricette di immagini. Per ulteriori informazioni in merito AWS RAM, consulta la Guida per l'AWS Resource Access Manager utente. Per informazioni sulla condivisione delle risorse di Image Builder, vedere. Condividi le risorse di Image Builder con AWS RAM
- Amazon Simple Notification Service (Amazon SNS): se configurato, pubblica messaggi dettagliati sullo stato dell'immagine su un argomento SNS a cui sei abbonato. Per ulteriori informazioni su Amazon SNS, consulta Che cos'è Amazon SNS? nella Guida per gli sviluppatori di Amazon Simple Notification Service.

Argomenti sull'integrazione di prodotti e servizi

- EventBridge Integrazione di Amazon in Image Builder
- Integrazione di Amazon Inspector in Image Builder
- Marketplace AWS integrazione in Image Builder
- Integrazione di Amazon SNS in Image Builder
- Prodotti di conformità per le immagini di Image Builder

EventBridge Integrazione di Amazon in Image Builder

Amazon EventBridge è un servizio di bus eventi senza server che puoi utilizzare per connettere la tua applicazione Image Builder con dati correlati provenienti da altri. Servizi AWS In EventBridge, una regola abbina gli eventi in arrivo e li invia alle destinazioni per l'elaborazione. Una singola regola può inviare un evento a più destinazioni e questi eventi vengono quindi eseguiti in parallelo.

Con EventBridge, puoi automatizzare Servizi AWS e rispondere automaticamente a eventi di sistema come problemi di disponibilità delle applicazioni o modifiche delle risorse. Gli eventi di Servizi AWS vengono trasmessi quasi EventBridge in tempo reale. È possibile impostare regole che reagiscono agli eventi in arrivo per avviare azioni. Ad esempio, l'invio di un evento a una funzione Lambda quando lo stato di un' EC2 istanza passa da pendente a in esecuzione. Questi sono chiamati modelli. Per creare una regola basata su un pattern di eventi, consulta Creazione di EventBridge regole Amazon che reagiscono agli eventi nella Amazon EventBridge User Guide.

Le azioni che possono essere avviate automaticamente includono quanto segue:

- Richiamare una funzione AWS Lambda
- Richiama il comando Amazon EC2 Run
- Inoltro dell'evento a flusso di dati Amazon Kinesis
- Attiva una macchina a AWS Step Functions stati
- Notifica un argomento Amazon SNS o una coda Amazon SQS

È inoltre possibile impostare regole di pianificazione per il bus eventi predefinito per eseguire un'azione a intervalli regolari, ad esempio eseguire una pipeline di Image Builder per aggiornare un'immagine su base trimestrale. Esistono due tipi di espressioni di pianificazione:

- cron expressions — Il seguente esempio di espressione cron pianifica l'esecuzione di un'attività ogni giorno a mezzogiorno UTC+0:

  ```
  cron(0 12 * * ? *)
  ```

  Per ulteriori informazioni sull'utilizzo delle espressioni cron con EventBridge, consulta le espressioni Cron nella Amazon EventBridge User Guide.
- rate expression — Il seguente esempio di espressione di velocità pianifica l'esecuzione di un'attività ogni 12 ore:

  ```
  rate(12 hour)
  ```

Per ulteriori informazioni sull'utilizzo delle espressioni di frequenza con EventBridge, consulta Rate expression nella Amazon EventBridge User Guide.

Per ulteriori informazioni su come le EventBridge regole si integrano con le pipeline di immagini di Image Builder, vedere. Usa EventBridge le regole con le pipeline di Image Builder

Messaggi di eventi inviati da Image Builder

Image Builder invia messaggi di eventi in EventBridge caso di cambiamenti significativi nello stato delle risorse di Image Builder. Ad esempio, quando c'è un cambiamento di stato per un'immagine. Gli esempi seguenti mostrano i tipici messaggi di eventi JSON che Image Builder potrebbe inviare.

EC2 Modifica dello stato dell'immagine di Image Builder

Image Builder invia questo evento quando lo stato di una risorsa immagine cambia durante la creazione dell'immagine. Ad esempio, quando lo stato dell'immagine cambia da uno stato all'altro, come segue:

- Da `building` a `testing`
- Da `testing` a `distribution`
- Da `testing` a `failed`
- Da `integrating` a `available`

```
{
    "version": "0",
    "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "detail-type": "EC2 Image Builder Image State Change",
    "source": "aws.imagebuilder",
    "account": "111122223333",
    "time": "2024-01-18T17:50:56Z",
    "region": "us-west-2",
    "resources": ["arn:aws:imagebuilder:us-west-2:111122223333:image/
cmkencryptedworkflowtest-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222/1.0.0/1"],
    "detail": {
        "previous-state": {
            "status": "TESTING"
        },
        "state": {
            "status": "AVAILABLE"
        }
```

```
    }
}
```

EC2 Rilevato CVE di Image Builder

Se hai abilitato il rilevamento CVE per l'immagine, Image Builder invia un messaggio con i risultati ogni volta che viene completata una scansione dell'immagine.

```
{
    "version": "0",
    "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "detail-type": "EC2 Image Builder CVE Detected",
    "source": "aws.imagebuilder",
    "account": "111122223333",
    "time": "2023-03-01T16:59:09Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:imagebuilder:us-east-1:111122223333:image/test-image/1.0.0/1",
        "arn:aws:imagebuilder:us-east-1:111122223333:image-pipeline/test-pipeline"
    ],
    "detail": {
        "resource-id": "i-1234567890abcdef0",
        "finding-severity-counts": {
            "all": 0,
            "critical": 0,
            "high": 0,
            "medium": 0
        }
    }
}
```

EC2 Fase in attesa del flusso di lavoro di Image Builder

Image Builder invia un messaggio quando una fase del `WaitForAction` flusso di lavoro si interrompe in attesa del completamento di un'azione asincrona.

```
{
    "version": "0",
    "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "detail-type": "EC2 Image Builder Workflow Step Waiting",
    "source": "aws.imagebuilder",
    "account": "111122223333",
    "time": "2024-01-18T16:54:44Z",
```

```
    "region": "us-west-2",
    "resources": ["arn:aws:imagebuilder:us-west-2:111122223333:image/
workflowstepwaitforactionwithvalidsnstopictest-a1b2c3d4-5678-90ab-cdef-
EXAMPLE22222/1.0.0/1", "arn:aws:imagebuilder:us-west-2:111122223333:workflow/build/
build-workflow-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333/1.0.0/1"],
    "detail": {
        "workflow-execution-id": "wf-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "workflow-step-execution-id": "step-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "workflow-step-name": "TestAutoSNSStop"
    }
}
```

Integrazione di Amazon Inspector in Image Builder

Quando attivi la scansione di sicurezza con Amazon Inspector, esegue una scansione continua delle immagini delle macchine e delle istanze in esecuzione nel tuo account alla ricerca di vulnerabilità del sistema operativo e del linguaggio di programmazione. Se attivata, la scansione di sicurezza è automatica e Image Builder può salvare un'istantanea dei risultati dall'istanza di test quando si crea una nuova immagine. Amazon Inspector è un servizio a pagamento.

Quando Amazon Inspector rileva vulnerabilità nel software o nelle impostazioni di rete, intraprende le seguenti azioni:

- Ti avvisa che c'è stata una scoperta.
- Valuta la gravità del risultato. La classificazione di gravità classifica le vulnerabilità per aiutarvi a dare priorità ai risultati e include i seguenti valori:
  - Non classificato
  - Messaggio informativo
  - Bassa
  - Media
  - Elevata
  - Critico
- Fornisce informazioni sulla scoperta e collegamenti a risorse aggiuntive per ulteriori dettagli.
- Offre indicazioni sulla correzione per aiutarvi a risolvere i problemi che hanno generato il risultato.

Configura le scansioni di sicurezza

Se hai attivato Amazon Inspector per il tuo account, Amazon Inspector analizza automaticamente EC2 le istanze avviate da Image Builder per creare e testare una nuova immagine. Queste istanze hanno una durata breve durante il processo di creazione e test e i loro risultati normalmente scadono non appena tali istanze vengono chiuse. Per aiutarti a esaminare e correggere i risultati della tua nuova immagine, Image Builder può facoltativamente salvare come snapshot tutti i risultati identificati da Amazon Inspector sull'istanza di test durante il processo di creazione.

Per configurare le scansioni di sicurezza per la tua pipeline, consulta. Configurare le scansioni di sicurezza per le immagini di Image Builder in AWS Management Console

Esamina i risultati sulla sicurezza

Nella console Image Builder, è possibile visualizzare i risultati di sicurezza per tutte le risorse di Image Builder in un unico posto. È possibile visualizzare tutti i risultati nella pagina Risultati di sicurezza nella sezione Panoramica sulla sicurezza oppure raggruppare i risultati per vulnerabilità, per pipeline di immagini o per immagine. L'impostazione predefinita della console mostra tutti i risultati di sicurezza. Il pannello di riepilogo dell'opzione Tutti i risultati di sicurezza mostra il numero di risultati disponibili per ogni livello di gravità. Per ulteriori informazioni, consulta Gestisci i risultati di sicurezza per le immagini di Image Builder in AWS Management Console.

Per ulteriori informazioni sui risultati delle vulnerabilità di Amazon Inspector, consulta la sezione Comprendere i risultati in Amazon Inspector nella Amazon Inspector User Guide.

Marketplace AWS integrazione in Image Builder

Marketplace AWS è un catalogo digitale curato in cui puoi trovare e abbonarti a software, dati e servizi di terze parti che ti aiutano a creare soluzioni adatte alle tue esigenze aziendali. Marketplace AWS offre agli acquirenti autenticati e ai venditori registrati offerte di software appartenenti a categorie popolari come sicurezza, rete, archiviazione, apprendimento automatico e altro ancora.

Un Marketplace AWS venditore può essere un fornitore di software indipendente (ISV), un rivenditore o un individuo che ha qualcosa da offrire relativo a prodotti e servizi. AWS Quando il venditore invia un prodotto Marketplace AWS, definisce il prezzo del prodotto e i termini e le condizioni d'uso. Gli acquirenti accettano i prezzi, i termini e le condizioni stabiliti per l'offerta. Per saperne di più Marketplace AWS, consulta Cos'è Marketplace AWS?

Marketplace AWS funzionalità di integrazione

Image Builder si integra con Marketplace AWS per fornire le seguenti funzionalità direttamente dalla console Image Builder:

- Cerca i prodotti di immagine disponibili in. Marketplace AWS
- Cerca prodotti di Marketplace AWS immagini che forniscono componenti.
- Visualizza un elenco dei tuoi attuali abbonamenti ai Marketplace AWS prodotti.
- Usa un prodotto Marketplace AWS illustrativo a cui ti sei abbonato come immagine di base per una ricetta di Image Builder.
- Usa Marketplace AWS i componenti a cui ti sei abbonato in una ricetta di Image Builder.

Image Builder si integra con i prodotti e i componenti di immagini Marketplace AWS a cui ti sei abbonato. È inoltre possibile cercare prodotti e componenti di Marketplace AWS immagini dalla pagina Scopri i prodotti senza uscire dalla console Image Builder.

L'AMI di output creata da Image Builder include i codici prodotto dei prodotti e dei Marketplace AWS componenti di immagini. È possibile avere fino a quattro codici di prodotto per l'immagine personalizzata finale.

Marketplace AWS abbonamenti in Image Builder

La pagina Abbonamenti nella Marketplace AWS sezione della console Image Builder mostra un elenco Marketplace AWS dei prodotti a cui sei attualmente abbonato. Ogni prodotto sottoscritto mostra i seguenti dettagli:

- Il nome del prodotto. Questo è collegato alla pagina dei dettagli del prodotto in Marketplace AWS. La pagina dei dettagli del prodotto per il quale hai sottoscritto l'abbonamento si apre in una nuova scheda del browser.
- L'editore. Questo è collegato alla pagina dei dettagli dell'editore in Marketplace AWS. La pagina dei dettagli dell'editore si apre in una nuova scheda del browser.
- La versione a cui ti sei abbonato.
- Se nel prodotto sottoscritto sono inclusi componenti associati, Image Builder visualizza un collegamento ai dettagli del componente.

Nella parte superiore della pagina, puoi cercare un prodotto specifico per nome oppure puoi sfogliare i risultati con i controlli di impaginazione. Per utilizzare un prodotto immagine sottoscritto in una nuova ricetta, seleziona un prodotto sottoscritto e scegli Crea nuova ricetta. Image Builder preseleziona il primo prodotto dell'elenco per impostazione predefinita.

> **Note**
>
> Se stai cercando un prodotto a cui ti sei appena abbonato e non lo vedi nell'elenco, usa il pulsante di aggiornamento nella parte superiore della scheda per aggiornare i risultati. Potrebbero essere necessari alcuni minuti prima che un nuovo abbonamento compaia nell'elenco.

Scopri i prodotti di Marketplace AWS immagini dalla console Image Builder

Questa sezione si concentra sui Marketplace AWS prodotti illustrati da utilizzare come immagine di base nella ricetta. Per i prodotti che includono componenti software associati, puoi filtrare in base al proprietario del prodotto nella console e nell'API, nell'SDK e nella CLI. Per ulteriori informazioni, consulta Elenca i componenti di Image Builder. Per ulteriori informazioni sulla ricerca, la sottoscrizione e l'utilizzo Marketplace AWS dei componenti, consulta. Usa Marketplace AWS i componenti per personalizzare l'immagine

Scopri i prodotti

Per trovare un prodotto di Marketplace AWS immagini dalla console Image Builder, procedi nel seguente modo:

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Dal pannello di navigazione, scegli Scopri i prodotti nella Marketplace AWSsezione.
3. Puoi cercare prodotti illustrati nella scheda Prodotti immagine della pagina Scopri i prodotti.

   Image Builder prefiltra i prodotti Marketplace AWS per concentrarsi sulle immagini delle macchine da utilizzare nelle ricette di Image Builder. Per ulteriori informazioni sull' Marketplace AWS integrazione con Image Builder, scegli la scheda corrispondente a ciò che desideri vedere.

   Questa scheda contiene due pannelli. A sinistra, il pannello Perfeziona i risultati ti aiuta a filtrare i risultati per trovare i prodotti a cui desideri abbonarti. A destra, il pannello Cerca prodotti mostra i prodotti che soddisfano i criteri di filtro e ti dà anche la possibilità di cercare per nome del prodotto.

   Perfeziona i risultati

   L'elenco seguente mostra solo alcuni dei filtri che puoi applicare alla ricerca dei prodotti:

- Seleziona una o più categorie di prodotti, come software di infrastruttura o apprendimento automatico.
- Scegliete i sistemi operativi per il vostro prodotto di immagine o scegliete tutti i prodotti per una piattaforma di sistema operativo specifica, ad esempio All Linux/Unix.
- Scegli uno o più editori per mostrare i prodotti disponibili. Seleziona il link Mostra tutto per visualizzare tutti gli editori che hanno prodotti che soddisfano i filtri che hai applicato.

> **Note**
>
> I nomi degli editori non sono in ordine alfabetico. Se stai cercando un editore specifico, ad esempio`Center for Internet Security`, puoi inserire parte del nome nella casella di ricerca nella parte superiore della finestra di dialogo Tutti gli editori. Dovresti pronunciare il nome per esteso, come abbreviazione, per evitare che produca i risultati che stai cercando. `CIS`
> Puoi anche sfogliare i nomi degli editori pagina per pagina.

Le scelte di filtro sono dinamiche. Ogni scelta effettuata influisce sulle opzioni disponibili per tutte le altre categorie. Sono disponibili migliaia di prodotti Marketplace AWS, quindi più puoi filtrare, più è probabile che trovi quello che desideri.

Cerca prodotti

Per trovare un prodotto specifico per nome, puoi inserire parte del nome nella barra di ricerca nella parte superiore di questo pannello. Ogni risultato di prodotto include i seguenti dettagli:

- Il nome e il logo del prodotto. Entrambi sono collegati alla pagina dei dettagli del prodotto in Marketplace AWS. La pagina dei dettagli si apre in una nuova scheda del browser. Da lì, puoi abbonarti al prodotto illustrato se desideri utilizzarlo in una ricetta di Image Builder. Per ulteriori informazioni, consulta la sezione Acquisto di prodotti nella Guida Marketplace AWS all'acquisto.

  Se ti abboni al prodotto di immagini in Marketplace AWS, torna alla scheda Image Builder del browser e aggiorna l'elenco dei prodotti di immagini sottoscritti per visualizzarlo.

> **Note**
>
> Potrebbero essere necessari alcuni minuti prima che il nuovo abbonamento sia disponibile.

- Il nome dell'editore. È collegato alla pagina dei dettagli dell'editore in Marketplace AWS. La pagina dei dettagli dell'editore si apre in una nuova scheda del browser.
- La versione del prodotto.
- La valutazione in stelle del prodotto e i link diretti alla sezione delle recensioni della pagina dei dettagli del prodotto in Marketplace AWS. La pagina dei dettagli si apre in una nuova scheda del browser.
- Le prime righe della descrizione del prodotto.

Direttamente sotto la barra di ricerca, puoi vedere quanti risultati ha prodotto la tua ricerca e quale sottoinsieme di tali risultati è attualmente visualizzato. Puoi utilizzare controlli aggiuntivi sul lato destro del pannello per regolare le impostazioni relative al numero di prodotti da visualizzare contemporaneamente e all'ordinamento da applicare ai risultati. Puoi anche usare il controllo di impaginazione per sfogliare i risultati.

Usa un prodotto di Marketplace AWS immagini nelle ricette di Image Builder

Apri la pagina Crea ricetta e seleziona un prodotto di Marketplace AWS immagine da utilizzare come immagine di base, come segue.

1. Aprire la console EC2 Image Builder all'indirizzo. https://console.aws.amazon.com/imagebuilder/
2. Dal pannello di navigazione, scegli Ricette di immagini nella Marketplace AWSsezione. Questo mostra un elenco di ricette di immagini che hai creato.
3. Scegli Crea ricetta di immagine. Si apre la pagina Crea ricetta.
4. Inserisci il nome e la versione della ricetta nella sezione Dettagli della ricetta come di consueto.
5. Nella sezione Immagine di base, scegli l'opzione Marketplace AWS Immagini. Questo ti mostra un elenco dei prodotti Marketplace AWS illustrati a cui ti sei abbonato nella scheda Abbonamenti. Puoi scegliere l'immagine di base dall'elenco.

   Puoi anche cercare altri prodotti di immagine disponibili Marketplace AWS direttamente dalla Marketplace AWSscheda. Scegli Aggiungi prodotti o apri direttamente la Marketplace

AWSscheda. Per ulteriori informazioni su come impostare i filtri e effettuare ricerche in Marketplace AWS, consultaScopri i prodotti di Marketplace AWS immagini dalla console Image Builder.

6. Inserisci i dettagli rimanenti come al solito. Se alcuni degli abbonamenti ai prodotti includono componenti di compilazione, puoi selezionarli dall'elenco Build components. Selezionateli `Marketplace AWS` dall'elenco dei tipi di proprietari dei componenti per visualizzarli oppure selezionate `Third party managed` per il componente CIS.
7. Scegli Crea ricetta.

   L'immagine finale può contenere fino a quattro codici prodotto relativi a prodotti e componenti di Marketplace AWS immagini. Se l'immagine di base e i componenti selezionati contengono più di quattro codici prodotto, Image Builder restituisce un errore quando si tenta di creare la ricetta.

Integrazione di Amazon SNS in Image Builder

Amazon Simple Notification Service (Amazon SNS) è un servizio gestito che fornisce il recapito asincrono dei messaggi dagli editori agli abbonati (noti anche come produttori e consumatori).

Puoi specificare un argomento SNS nella configurazione dell'infrastruttura. Quando create un'immagine o eseguite una pipeline, Image Builder può pubblicare messaggi dettagliati sullo stato dell'immagine su questo argomento. Quando lo stato dell'immagine raggiunge uno dei seguenti stati, Image Builder pubblica un messaggio:

- `AVAILABLE`
- `FAILED`

Per un messaggio SNS di esempio da Image Builder, vedere. Formato del messaggio SNS Se desideri creare un nuovo argomento SNS, consulta la sezione Guida introduttiva ad Amazon SNS nella Amazon Simple Notification Service Developer Guide.

Argomenti SNS crittografati

Se l'argomento SNS è crittografato, è necessario concedere l'autorizzazione nella AWS KMS key politica per il ruolo del servizio Image Builder per eseguire le seguenti azioni:

- `kms:Decrypt`
- `kms:GenerateDataKey`

> **Note**
>
> Se l'argomento SNS è crittografato, la chiave che crittografa questo argomento deve risiedere nell'account in cui viene eseguito il servizio Image Builder. Image Builder non può inviare notifiche ad argomenti SNS crittografati con chiavi di altri account.

Esempio di aggiunta della policy chiave KMS

L'esempio seguente mostra la sezione aggiuntiva che aggiungi alla politica delle chiavi KMS. Usa Amazon Resource Name (ARN) per il ruolo collegato al servizio IAM che Image Builder ha creato con il tuo account quando hai creato per la prima volta un'immagine Image Builder. Per ulteriori informazioni sul ruolo collegato al servizio Image Builder, vedere. Usa i ruoli collegati ai servizi IAM per 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": "*"
  }]
}
```

È possibile utilizzare uno dei seguenti metodi per ottenere l'ARN.

AWS Management Console

Per ottenere l'ARN per il ruolo collegato al servizio che Image Builder ha creato con il tuo account da, procedi nel seguente modo: AWS Management Console

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione a sinistra, seleziona Ruoli.

3. Cerca`ImageBuilder`, e scegli il seguente nome del ruolo tra i risultati:. `AWSServiceRoleForImageBuilder` Viene visualizzata la pagina dei dettagli del ruolo.
4. Per copiare l'ARN negli appunti, scegli l'icona accanto al nome ARN.

AWS CLI

Per ottenere l'ARN per il ruolo collegato al servizio che Image Builder ha creato con il tuo account da AWS CLI, usa il comando IAM get-role, come segue.

```
aws iam get-role --role-name AWSServiceRoleForImageBuilder
```

Esempio di output parziale:

```
{
    "Role": {
        "Path": "/aws-service-role/imagebuilder.amazonaws.com/",
        "RoleName": "AWSServiceRoleForImageBuilder",
        ...
        "Arn": "arn:aws:iam::123456789012:role/aws-service-role/
imagebuilder.amazonaws.com/AWSServiceRoleForImageBuilder",
        ...
}
```

Formato del messaggio SNS

Dopo che Image Builder ha pubblicato un messaggio sul tuo argomento Amazon SNS, gli altri servizi che sottoscrivono l'argomento possono filtrare in base al formato del messaggio e determinare se soddisfa i criteri per ulteriori azioni. Ad esempio, un messaggio di successo potrebbe avviare un'attività per aggiornare un archivio di AWS Systems Manager parametri o avviare un flusso di lavoro di test di conformità esterno per l'AMI di output.

L'esempio seguente mostra il payload JSON per un messaggio tipico che Image Builder pubblica quando una build di pipeline viene completata e crea un'immagine Linux.

```
{
  "versionlessArn": "arn:aws:imagebuilder:us-west-1:123456789012:image/example-linux-
image",
  "semver": 12379400392853802748991242227,
```

```
  "arn": "arn:aws:imagebuilder:us-west-1:123456789012:image/example-linux-
image/1.0.0/3",
  "name": "example-linux-image",
  "version": "1.0.0",
  "type": "AMI",
  "buildVersion": 3,
  "state": {
    "status": "AVAILABLE"
  },
  "platform": "Linux",
  "imageRecipe": {
    "arn": "arn:aws:imagebuilder:us-west-1:123456789012:image-recipe/example-linux-
image/1.0.0",
    "name": "amjule-barebones-linux",
    "version": "1.0.0",
    "components": [
      {
        "componentArn": "arn:aws:imagebuilder:us-west-1:123456789012:component/update-
linux/1.0.2/1"
      }
    ],
    "platform": "Linux",
    "parentImage": "arn:aws:imagebuilder:us-west-1:987654321098:image/amazon-linux-2-
x86/2022.6.14/1",
    "blockDeviceMappings": [
      {
        "deviceName": "/dev/xvda",
        "ebs": {
          "encrypted": false,
          "deleteOnTermination": true,
          "volumeSize": 8,
          "volumeType": "gp2"
        }
      }
    ],
    "dateCreated": "Feb 24, 2021 12:31:54 AM",
    "tags": {
      "internalId": "1a234567-8901-2345-bcd6-ef7890123456",
      "resourceArn": "arn:aws:imagebuilder:us-west-1:123456789012:image-recipe/example-
linux-image/1.0.0"
    },
    "workingDirectory": "/tmp",
    "accountId": "462045008730"
  },
```

```
  "sourcePipelineArn": "arn:aws:imagebuilder:us-west-1:123456789012:image-pipeline/
example-linux-pipeline",
  "infrastructureConfiguration": {
    "arn": "arn:aws:imagebuilder:us-west-1:123456789012:infrastructure-configuration/
example-linux-infra-config-uswest1",
    "name": "example-linux-infra-config-uswest1",
    "instanceProfileName": "example-linux-ib-baseline-admin",
    "tags": {
      "internalId": "234abc56-d789-0123-a4e5-6b789d012c34",
      "resourceArn": "arn:aws:imagebuilder:us-west-1:123456789012:infrastructure-
configuration/example-linux-infra-config-uswest1"
    },
    "logging": {
      "s3Logs": {
        "s3BucketName": "amzn-s3-demo-bucket"
      }
    },
    "keyPair": "example-linux-key-pair-uswest1",
    "terminateInstanceOnFailure": true,
    "snsTopicArn": "arn:aws:sns:us-west-1:123456789012:example-linux-ibnotices-
uswest1",
    "dateCreated": "Feb 24, 2021 12:31:55 AM",
    "accountId": "123456789012"
  },
  "imageTestsConfigurationDocument": {
    "imageTestsEnabled": true,
    "timeoutMinutes": 720
  },
  "distributionConfiguration": {
    "arn": "arn:aws:imagebuilder:us-west-1:123456789012:distribution-configuration/
example-linux-distribution",
    "name": "example-linux-distribution",
    "dateCreated": "Feb 24, 2021 12:31:56 AM",
    "distributions": [
      {
        "region": "us-west-1",
        "amiDistributionConfiguration": {}
      }
    ],
    "tags": {
      "internalId": "345abc67-8910-12d3-4ef5-67a8b90c12de",
      "resourceArn": "arn:aws:imagebuilder:us-west-1:123456789012:distribution-
configuration/example-linux-distribution"
    },
```

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

L'esempio seguente mostra il payload JSON per un messaggio tipico pubblicato da Image Builder per un errore di compilazione della pipeline per un'immagine Linux.

```
{
  "versionlessArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-
image",
  "semver": 123794003928538027489912423 1,
  "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-image/1.0.0/7",
  "name": "My Example Image",
  "version": "1.0.0",
  "type": "AMI",
  "buildVersion": 7,
  "state": {
    "status": "FAILED",
    "reason": "Image Failure reason."
```

```
  },
  "platform": "Linux",
  "imageRecipe": {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-example-
image/1.0.0",
    "name": "My Example Image",
    "version": "1.0.0",
    "description": "Testing Image recipe",
    "components": [
      {
        "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/my-
example-image-component/1.0.0/1"
      }
    ],
    "platform": "Linux",
    "parentImage": "ami-0cd12345db678d90f",
    "dateCreated": "Jun 21, 2022 11:36:14 PM",
    "tags": {
      "internalId": "1a234567-8901-2345-bcd6-ef7890123456",
      "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-
example-image/1.0.0"
    },
    "accountId": "123456789012"
  },
  "sourcePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/my-
example-image-pipeline",
  "infrastructureConfiguration": {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/
my-example-infra-config",
    "name": "SNS topic Infra config",
    "description": "An example that will retain instances of failed builds",
    "instanceTypes": [
      "t2.micro"
    ],
    "instanceProfileName": "EC2InstanceProfileForImageBuilder",
    "tags": {
      "internalId": "234abc56-d789-0123-a4e5-6b789d012c34",
      "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-
configuration/my-example-infra-config"
    },
    "terminateInstanceOnFailure": true,
    "snsTopicArn": "arn:aws:sns:us-west-2:123456789012:example-pipeline-notification-
topic",
    "dateCreated": "Jul 5, 2022 7:31:53 PM",
```

```
    "accountId": "123456789012"
  },
  "imageTestsConfigurationDocument": {
    "imageTestsEnabled": true,
    "timeoutMinutes": 720
  },
  "distributionConfiguration": {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:distribution-configuration/my-
example-distribution-config",
    "name": "New distribution config",
    "dateCreated": "Dec 3, 2021 9:24:22 PM",
    "distributions": [
      {
        "region": "us-west-2",
        "amiDistributionConfiguration": {},
        "fastLaunchConfigurations": [
          {
            "enabled": true,
            "snapshotConfiguration": {
              "targetResourceCount": 2
            },
            "maxParallelLaunches": 2,
            "launchTemplate": {
              "launchTemplateId": "lt-01234567890"
            },
            "accountId": "123456789012"
          }
        ]
      }
    ],
    "tags": {
      "internalId": "1fecd23a-4f56-7f89-01e2-345678abbe90",
      "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:distribution-
configuration/my-example-distribution-config"
    },
    "accountId": "123456789012"
  },
  "dateCreated": "Jul 5, 2022 7:40:15 PM",
  "outputResources": {
    "amis": []
  },
  "accountId": "123456789012",
  "enhancedImageMetadataEnabled": true,
  "buildType": "SCHEDULED",
```

```
  "tags": {
    "internalId": "456c78b9-0e12-3f45-afb6-7e89b0f1a23b",
    "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-
image/1.0.0/7"
  }
}
```

Prodotti di conformità per le immagini di Image Builder

Con standard di sicurezza in continua evoluzione, mantenere la conformità e salvaguardare l'organizzazione dalle minacce informatiche può essere una sfida. Per garantire la conformità delle immagini personalizzate e mantenerle invariate grazie agli aggiornamenti automatici quando gli editori rilasciano nuove versioni, Image Builder si integra con i prodotti Marketplace AWS di conformità e i componenti di Image Builder.

Image Builder si integra con i seguenti prodotti di conformità:

- Rafforzamento dei benchmark di Center for Internet Security (CIS)

  È possibile utilizzare CIS Hardened Images e i relativi componenti di protezione avanzata CIS per creare immagini personalizzate conformi alle più recenti linee guida CIS Benchmarks Level 1. Le immagini CIS Hardened sono disponibili in. Marketplace AWS Per ulteriori informazioni su come configurare e utilizzare le immagini protette e i componenti di protezione avanzata CIS, consultate le guide di avvio rapido nel portale di supporto del sito Web CIS.

  > **Note**
  >
  > Quando ti abboni a un'immagine protetta CIS, hai anche accesso al componente di compilazione associato che esegue uno script per applicare le linee guida CIS Benchmark Level 1 per la tua configurazione. Per ulteriori informazioni, consulta Componenti di tempra CIS.

- Guide tecniche all'implementazione della sicurezza (STIG)

  Per la conformità STIG, puoi utilizzare componenti STIG gestiti da Amazon AWS Task Orchestrator and Executor (AWSTOE) nelle tue ricette di Image Builder. I componenti STIG scansionano l'istanza di build alla ricerca di configurazioni errate ed eseguono uno script di correzione per correggere i problemi rilevati. Non possiamo garantire la conformità STIG per le immagini create con Image Builder. È necessario collaborare con il team di conformità della propria organizzazione

per verificare che l'immagine finale sia conforme. Per un elenco completo dei componenti AWSTOE STIG che è possibile utilizzare nelle ricette di Image Builder, vedere. Amazon ha gestito i componenti di protezione STIG per Image Builder

Monitora eventi e registri in Image Builder

Per mantenere l'affidabilità, la disponibilità e le prestazioni delle pipeline di EC2 Image Builder, è importante monitorare eventi e registri. Gli eventi e i registri ti aiutano a vedere il quadro generale e ad approfondire i dettagli quando una chiamata API fallisce. Image Builder si integra con servizi in grado di inviare avvisi e avviare risposte automatiche quando gli eventi soddisfano i criteri che hai configurato.

I seguenti argomenti descrivono le tecniche di monitoraggio che è possibile utilizzare tramite servizi che si integrano con Image Builder.

Monitora eventi e registri

- Registra le chiamate API Image Builder utilizzando CloudTrail
- Monitora i log di Image Builder con Amazon Logs CloudWatch

Registra le chiamate API Image Builder utilizzando CloudTrail

EC2 Image Builder è integrato con AWS CloudTrail, un servizio che fornisce una registrazione delle azioni intraprese da un utente, ruolo o AWS servizio in Image Builder. CloudTrail acquisisce tutte le chiamate API per Image Builder come eventi. Le chiamate acquisite includono chiamate dalla console Image Builder e chiamate di codice alle operazioni dell'API Image Builder. Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per Image Builder. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare la richiesta effettuata a Image Builder, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e ulteriori dettagli.

Per ulteriori informazioni CloudTrail, consulta la Guida per l'AWS CloudTrail utente.

Informazioni su Image Builder in CloudTrail

CloudTrail è abilitato sul tuo Account AWS quando crei l'account. Quando si verifica un'attività in Image Builder, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio nella cronologia degli eventi. È possibile visualizzare, cercare e scaricare eventi recenti in. Account AWS Per ulteriori informazioni, consulta Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi nel tuo Account AWS, compresi gli eventi per Image Builder, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando si crea un percorso nella console, questo sarà valido in tutte le Regioni AWS. Il trail registra gli eventi di tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta gli argomenti seguenti:

- Panoramica della creazione di un percorso
- CloudTrail servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Tutte le azioni di Image Builder vengono registrate CloudTrail e documentate nell'Image Builder EC2 API Reference. Ad esempio, le chiamate alle `StartImagePipelineExecution` azioni `CreateImagePipelineUpdateInfrastructureConfiguration`, e generano voci nei file di registro. CloudTrail

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con credenziali utente root o AWS Identity and Access Management (IAM).
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, consulta Elemento CloudTrail userIdentity.

Monitora i log di Image Builder con Amazon Logs CloudWatch

CloudWatch Il supporto per i log è attivato per impostazione predefinita. I log vengono conservati sull'istanza durante il processo di compilazione e trasmessi in streaming a Logs. CloudWatch I log delle istanze vengono rimossi dall'istanza prima della creazione dell'immagine.

I log di compilazione vengono trasmessi in streaming seguendo il gruppo e lo stream Image CloudWatch Builder Logs:

LogGroup:

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

LogStream (x.x.x/x):

```
ImageVersion/ImageBuildVersion
```

Puoi disattivare lo streaming dei CloudWatch log rimuovendo le seguenti autorizzazioni associate al profilo dell'istanza.

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

Per una risoluzione avanzata dei problemi, puoi eseguire comandi e script predefiniti utilizzando Run Command.AWS Systems Manager Per ulteriori informazioni, consulta Risolvere i problemi relativi a Image Builder.

Sicurezza in Image Builder

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS te e te. Il modello di responsabilità condivisa descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che funziona Servizi AWS nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei AWS Programmi di AWS conformità dei Programmi di conformità dei di . Per ulteriori informazioni sui programmi di conformità che si applicano a EC2 Image Builder, vedere Servizi AWS in Scope by Compliance Program AWS Services in Scope Program.
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

Questa documentazione aiuta a capire come applicare il modello di responsabilità condivisa quando si utilizza Image Builder. I seguenti argomenti mostrano come configurare Image Builder per soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche a usarne altri Servizi AWS che ti aiutano a monitorare e proteggere le tue risorse di Image Builder.

Argomenti

- Protezione dei dati e modello di responsabilità AWS condivisa in Image Builder
- Integrazione di Identity and Access Management per Image Builder
- Risorse per la convalida della conformità per Image Builder
- Ridondanza e resilienza dei dati in Image Builder
- Sicurezza dell'infrastruttura in Image Builder
- Gestione delle patch per le immagini di Image Builder
- Best practice di sicurezza per Image Builder

Protezione dei dati e modello di responsabilità AWS condivisa in Image Builder

Il modello di responsabilità AWS condivisa modello di di si applica alla protezione dei dati in EC2 Image Builder. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, vedi le Domande frequenti sulla privacy dei dati. Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al Modello di responsabilità condivisa AWS e GDPR nel Blog sulla sicurezza AWS .

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire AWS le attività, consulta Lavorare con i CloudTrail percorsi nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-3 per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-3.

Ti consigliamo di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori con Image Builder o altro Servizi AWS utilizzando la console, l'API o. AWS CLI AWS SDKs I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere

utilizzati per i la fatturazione o i log di diagnostica. Quando fornisci un URL a un server esterno, ti suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.

Crittografia e gestione delle chiavi in Image Builder

Image Builder crittografa i dati in transito e a riposo per impostazione predefinita con una chiave KMS di proprietà del servizio, ad eccezione di quanto segue:

- Componenti personalizzati: Image Builder crittografa i componenti personalizzati con la chiave KMS predefinita o una chiave KMS di proprietà del servizio.
- Flussi di lavoro di immagini: Image Builder può crittografare i flussi di lavoro di immagini con una chiave gestita dal cliente se si specifica la chiave durante la creazione del flusso di lavoro. Image Builder gestisce la crittografia e la decrittografia con la tua chiave per eseguire i flussi di lavoro che hai configurato per le tue immagini.

È possibile gestire le proprie chiavi tramite. AWS KMS Tuttavia, non sei autorizzato a gestire la chiave KMS di Image Builder di proprietà di Image Builder. Per ulteriori informazioni sulla gestione delle chiavi KMS con AWS Key Management Service, consulta Getting Started nella Developer Guide. AWS Key Management Service

Contesto di crittografia

Per fornire un ulteriore controllo di integrità e autenticità dei dati crittografati, hai la possibilità di includere un contesto di crittografia quando crittografi i dati. Quando una risorsa viene crittografata con un contesto di AWS KMS crittografia, associa crittograficamente il contesto al testo cifrato. La risorsa può essere decrittografata solo se il richiedente fornisce una corrispondenza esatta con distinzione tra maiuscole e minuscole per il contesto.

Gli esempi di policy in questa sezione utilizzano un contesto di crittografia simile all'Amazon Resource Name (ARN) di una risorsa del flusso di lavoro Image Builder.

Crittografa i flussi di lavoro relativi alle immagini con una chiave gestita dal cliente

Per aggiungere un livello di protezione, puoi crittografare le risorse del flusso di lavoro di Image Builder con la tua chiave gestita dal cliente. Se si utilizza la chiave gestita dal cliente per crittografare i flussi di lavoro di Image Builder creati, è necessario concedere l'accesso nella politica chiave affinché Image Builder possa utilizzare la chiave quando crittografa e decrittografa le risorse del

flusso di lavoro. Puoi revocare l'accesso in qualsiasi momento. Tuttavia, Image Builder non avrà accesso a nessun flusso di lavoro già crittografato se si revoca l'accesso alla chiave.

Il processo per concedere a Image Builder l'accesso all'utilizzo della chiave gestita dal cliente prevede due passaggi, come segue:

Fase 1: Aggiungere le autorizzazioni chiave per i flussi di lavoro di Image Builder

Per consentire a Image Builder di crittografare e decrittografare le risorse del flusso di lavoro quando crea o utilizza tali flussi di lavoro, è necessario specificare le autorizzazioni nella politica delle chiavi KMS.

Questo esempio di policy chiave concede l'accesso alle pipeline di Image Builder per crittografare le risorse del flusso di lavoro durante il processo di creazione e decrittografare le risorse del flusso di lavoro per utilizzarle. La politica concede inoltre l'accesso agli amministratori chiave. Il contesto di crittografia e le specifiche delle risorse utilizzano un carattere jolly per coprire tutte le regioni in cui sono disponibili risorse per il flusso di lavoro.

Come prerequisito per l'utilizzo dei flussi di lavoro di immagini, hai creato un ruolo di esecuzione del flusso di lavoro IAM che concede l'autorizzazione a Image Builder per eseguire azioni di flusso di lavoro. Il principio della prima istruzione mostrata nell'esempio di policy chiave riportato qui deve specificare il ruolo di esecuzione del flusso di lavoro IAM.

Per ulteriori informazioni sulle chiavi gestite dai clienti, consulta Gestire l'accesso alle chiavi gestite dal cliente nella Guida per gli AWS Key Management Service sviluppatori.

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

Passaggio 2: concedi l'accesso chiave al tuo ruolo di esecuzione del flusso di lavoro

Il ruolo IAM che Image Builder assume per eseguire i flussi di lavoro richiede l'autorizzazione per utilizzare la chiave gestita dal cliente. Senza l'accesso alla tua chiave, Image Builder non sarà in grado di crittografare o decrittografare le risorse del flusso di lavoro con essa.

Modifica la politica per il tuo ruolo di esecuzione del flusso di lavoro per aggiungere la seguente dichiarazione di politica.

```
{
 "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 eventi per flussi di lavoro basati su immagini

Gli esempi seguenti mostrano le AWS CloudTrail voci tipiche per la crittografia e la decrittografia dei flussi di lavoro di immagini archiviate con una chiave gestita dal cliente.

Esempio: GenerateDataKey

Questo esempio mostra come potrebbe apparire un CloudTrail evento quando Image Builder richiama l'azione API dall'azione dell' AWS KMS GenerateDataKeyAPI Image Builder. CreateWorkflow Image Builder deve crittografare un nuovo flusso di lavoro prima di creare la risorsa del flusso di lavoro.

```
{
 "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"
}
```


Esempio: Decrypt

Questo esempio mostra come potrebbe apparire un CloudTrail evento quando Image Builder richiama l'azione API dall'azione dell' AWS KMS DecryptAPI Image Builder. GetWorkflow Le pipeline di Image Builder devono decrittografare una risorsa del flusso di lavoro prima di poterla utilizzare.


```
{
 "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"
}
```

Archiviazione dei dati in Image Builder

Image Builder non memorizza nessuno dei tuoi log nel servizio. Tutti i log vengono salvati sull' EC2 istanza Amazon utilizzata per creare l'immagine o nei log di automazione di Systems Manager.

Privacy del traffico tra reti in Image Builder

Le connessioni sono protette tra Image Builder e le sedi locali, AZs tra l'interno di AWS una regione e AWS tra le regioni tramite HTTPS. Non esistono connessioni dirette tra gli account.

Integrazione di Identity and Access Management per Image Builder

Argomenti

- Destinatari
- Autenticazione con identità
- Come Image Builder funziona con le politiche e i ruoli IAM
- Gestisci i perimetri dei dati per l'accesso al download dei bucket S3 in Image Builder
- Politiche basate sull'identità di Image Builder
- Politiche basate sulle risorse di Image Builder
- Usa politiche AWS gestite per EC2 Image Builder
- Usa i ruoli collegati ai servizi IAM per Image Builder
- Risolvi i problemi di IAM in Image Builder

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia a seconda del lavoro svolto in Image Builder.

Utente del servizio: se si utilizza il servizio Image Builder per svolgere il proprio lavoro, l'amministratore fornisce le credenziali e le autorizzazioni necessarie. Man mano che utilizzi più funzionalità di Image Builder per svolgere il tuo lavoro, potresti aver bisogno di autorizzazioni aggiuntive. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non è possibile accedere a una funzionalità in Image Builder, vedere. Risolvi i problemi di IAM in Image Builder

Amministratore del servizio: se sei responsabile delle risorse di Image Builder presso la tua azienda, probabilmente hai pieno accesso a Image Builder. Il tuo compito è determinare a quali funzionalità e risorse di Image Builder devono accedere gli utenti del servizio. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per ulteriori informazioni su come la tua azienda può utilizzare IAM con Image Builder, consulta. Come Image Builder funziona con le politiche e i ruoli IAM

Amministratore IAM: se sei un amministratore IAM, potresti voler conoscere i dettagli su come scrivere policy per gestire l'accesso a Image Builder. Per visualizzare esempi di policy basate sull'identità di Image Builder che è possibile utilizzare in IAM, vedere. Politiche basate sull'identità di Image Builder

Autenticazione con identità

Per informazioni dettagliate su come fornire l'autenticazione a persone e processi in uso Account AWS, consulta Identities in the IAM User Guide.

Come Image Builder funziona con le politiche e i ruoli IAM

Prima di utilizzare IAM per gestire l'accesso a Image Builder, scopri quali funzionalità IAM sono disponibili per l'uso con Image Builder.

Per avere una visione di alto livello di come Image Builder e AWS altri servizi funzionano con la maggior parte delle funzionalità IAM, AWS consulta i servizi che funzionano con IAM nella IAM User Guide.

Politiche basate sull'identità per Image Builder

Supporta le policy basate su identità: sì

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e

ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente IAM.

Con le policy basate su identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta Guida di riferimento agli elementi delle policy JSON IAM nella Guida per l'utente di IAM.

Esempi di policy basate sull'identità per Image Builder

Per visualizzare esempi di politiche basate sull'identità di Image Builder, vedere. Politiche basate sull'identità di Image Builder

Politiche basate sulle risorse all'interno di Image Builder

Supporta le policy basate sulle risorse: sì

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, puoi specificare un intero account o entità IAM in un altro account come principale in una policy basata sulle risorse. L'aggiunta di un principale multi-account a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono diversi Account AWS, un amministratore IAM dell'account affidabile deve inoltre concedere all'entità principale (utente o ruolo) l'autorizzazione ad accedere alla risorsa. L'autorizzazione viene concessa collegando all'entità una policy basata sull'identità. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate su identità. Per ulteriori informazioni, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Azioni politiche per Image Builder

Supporta le operazioni di policy: si

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Action`di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni politiche in genere hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le operazioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco delle azioni di Image Builder, vedere Azioni definite da Image EC2 Builder nel Service Authorization Reference.

Le azioni politiche in Image Builder utilizzano il seguente prefisso prima dell'azione:

```
imagebuilder
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

```
"Action": [
 "imagebuilder:action1",
 "imagebuilder:action2"
 ]
```

Per visualizzare esempi di politiche basate sull'identità di Image Builder, vedere. Politiche basate sull'identità di Image Builder

Risorse politiche per Image Builder

Supporta le risorse di policy: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource`della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Le istruzioni devono includere un elemento `Resource`o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). È possibile eseguire questa operazione per operazioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

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

Per visualizzare un elenco dei tipi di risorse di Image Builder e relativi ARNs, vedere Risorse definite da Image EC2 Builder nel Service Authorization Reference. Per sapere con quali azioni è possibile specificare l'ARN di ogni risorsa, vedere Azioni definite da Image EC2 Builder.

Per visualizzare esempi di politiche basate sull'identità di Image Builder, vedere. Politiche basate sull'identità di Image Builder

Chiavi delle condizioni dei criteri per Image Builder

Supporta le chiavi di condizione delle policy specifiche del servizio: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Condition`(o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition`è facoltativo. È possibile compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition`in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione `AND`logica. Se si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica. `OR` Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

È possibile anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, è possibile autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta Elementi delle policy IAM: variabili e tag nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

Per visualizzare un elenco delle chiavi di condizione di Image Builder, vedere Chiavi di condizione per Image EC2 Builder nel Service Authorization Reference. Per sapere con quali azioni e risorse è possibile utilizzare una chiave di condizione, vedere Azioni definite da EC2 Image Builder.

Per visualizzare esempi di politiche basate sull'identità di Image Builder, vedere. Politiche basate sull'identità di Image Builder

ACLs in Image Builder

Supporti ACLs: no

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

ABAC con Image Builder

Supporta ABAC (tag nelle policy): parzialmente

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. Puoi allegare tag a entità IAM (utenti o ruoli) e a molte AWS risorse. L'assegnazione di tag alle entità e alle risorse è il primo passaggio di ABAC. In seguito, vengono progettate policy ABAC per consentire operazioni quando il tag dell'entità principale corrisponde al tag sulla risorsa a cui si sta provando ad accedere.

La strategia ABAC è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa impegnativa.

Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Yes (Sì). Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni su ABAC, consulta Definizione delle autorizzazioni con autorizzazione ABAC nella Guida per l'utente IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta Utilizzo del controllo degli accessi basato su attributi (ABAC) nella Guida per l'utente di IAM.

Utilizzo di credenziali temporanee con Image Builder

Supporta le credenziali temporanee: sì

Alcune Servizi AWS non funzionano quando si accede utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che Servizi AWS funzionano con credenziali temporanee, consulta la sezione relativa alla Servizi AWS compatibilità con IAM nella IAM User Guide.

Stai utilizzando credenziali temporanee se accedi AWS Management Console utilizzando qualsiasi metodo tranne nome utente e password. Ad esempio, quando accedete AWS utilizzando il link Single Sign-On (SSO) della vostra azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sullo scambio dei ruoli, consulta Passaggio da un ruolo utente a un ruolo IAM (console) nella Guida per l'utente IAM.

È possibile creare manualmente credenziali temporanee utilizzando l'API or. AWS CLI AWS È quindi possibile utilizzare tali credenziali temporanee per accedere. AWS AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta Credenziali di sicurezza provvisorie in IAM.

Autorizzazioni principali per diversi servizi per Image Builder

Supporta l'inoltro delle sessioni di accesso (FAS): sì

Quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta Forward access sessions.

Ruoli di servizio per Image Builder

Supporta i ruoli di servizio: sì

Un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Create a role to delegate permissions to an Servizio AWS nella Guida per l'utente IAM.

> **Warning**
>
> La modifica delle autorizzazioni per un ruolo di servizio potrebbe interrompere la funzionalità di Image Builder. Modifica i ruoli di servizio solo quando Image Builder fornisce indicazioni in tal senso.

Ruoli collegati ai servizi per Image Builder

Supporta ruoli collegati ai servizi: Sì

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

Per informazioni dettagliate sul ruolo collegato al servizio Image Builder, vedere. Usa i ruoli collegati ai servizi IAM per Image Builder

Politiche basate sull'identità di Image Builder

Con le policy IAM basate su identità puoi specificare azioni e risorse consentite o rifiutate, nonché le condizioni in base alle quali le azioni sono consentite o rifiutate. Image Builder supporta azioni, risorse e chiavi di condizione specifiche. Per informazioni su tutti gli elementi utilizzati in una policy JSON, consulta Actions, Resources and Condition Keys for Amazon EC2 Image Builder nella IAM User Guide.

Azioni

Le azioni politiche in Image Builder utilizzano il seguente prefisso prima dell'azione:. `imagebuilder:` Le istruzioni della policy devono includere un elemento `Action` o `NotAction`. Image Builder definisce il proprio set di azioni che descrivono le attività che è possibile eseguire con questo servizio.

Per specificare più azioni in una sola istruzione, separa ciascuna di esse con una virgola come mostrato di seguito:

```
"Action": [
 "imagebuilder:action1",
 "imagebuilder:action2"
]
```

È possibile specificare più azioni tramite caratteri jolly (*). Ad esempio, per specificare tutte le azioni che iniziano con la parola `List`, includi la seguente azione:

```
"Action": "imagebuilder:List*"
```

Per visualizzare un elenco delle azioni di Image Builder, consulta Actions, Resources and Condition Keys Servizi AWS nella IAM User Guide.

Gestione dell'accesso con policy

Per informazioni dettagliate su come gestire l'accesso creando policy e collegandole a identità o AWS risorse IAM, consulta Policies and permissions nella IAM User Guide. AWS

Il ruolo IAM che associ al tuo profilo di istanza deve disporre delle autorizzazioni per eseguire i componenti di compilazione e test inclusi nell'immagine. Le seguenti policy di ruolo IAM devono essere collegate al ruolo IAM associato al profilo dell'istanza:

- EC2InstanceProfileForImageBuilder
- EC2InstanceProfileForImageBuilderECRContainerBuilds
- AmazonSSMManagedInstanceCore

Risorse

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource`della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Le istruzioni devono includere un elemento `Resource`o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). È possibile eseguire questa operazione per operazioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

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

L'ARN è composto da più nodi che aiutano a identificare la risorsa e garantiscono che il nome sia univoco. Gli ultimi nodi del nome includono diverse varianti di formattazione per il tipo, il nome e l'ID della risorsa. Quando Image Builder crea una risorsa, utilizza il seguente formato:

`arn:aws:imagebuilder:`*`region`*`:`*`owner`*`:`*`resource-type/resource-name/version/build-version`*

> **Note**
>
> La versione build non è sempre inclusa nell'ARN della risorsa. Tuttavia, alcune operazioni API, ad esempio GetComponent, richiedono la versione di build per identificare in modo univoco una risorsa da recuperare.

Per le risorse utilizzate da Image Builder nelle sue ricette, come l'immagine di base o i componenti, il nodo proprietario può essere uno dei seguenti:

- Il numero di account del proprietario della risorsa
- Per le risorse gestite da Amazon: `aws`
- Per Marketplace AWS le risorse: `aws-marketplace`

L'esempio seguente mostra l'ARN per un componente gestito per installare l' CloudWatch agente Amazon su Linux:

```
arn:aws:imagebuilder:us-east-1:aws:component/amazon-cloudwatch-agent-linux/1.0.1/1
```

Questo esempio mostra l'ARN per un componente gestito fittizio da: Marketplace AWS

```
arn:aws:imagebuilder:us-east-1:aws-marketplace:component/example-linux-software-component/1.0.1
```

Per ulteriori informazioni su come ottenere un elenco di componenti, incluso l'uso di un filtro di proprietà, vedere. Elenca i componenti di Image Builder

Esempio ARNs

Di seguito sono riportati alcuni esempi di risorse ARNs che è possibile specificare in una policy IAM:

- ARN istanza

```
"Resource": "arn:aws:imagebuilder:us-
east-1:111122223333:instance/i-1234567890abcdef0"
```

- Esempio Wildcard (*) per specificare tutte le istanze di un determinato account

```
"Resource": "arn:aws:imagebuilder:us-east-1:111122223333:instance/*"
```

- Esempio con Wildcard (*) per specificare tutte le versioni di un flusso di lavoro con immagini gestite

```
"Resource": "arn:aws:imagebuilder:us-east-1:aws:workflow/build/build-image/*"
```

- ARN dell'immagine gestita

```
"Resource": "arn:aws:imagebuilder:us-east-1:aws:image/amazon-linux-2-
arm64/2024.12.17/1"
```

- Esempio di Wildcard (*) per specificare tutte le versioni di un'immagine gestita

```
"Resource": "arn:aws:imagebuilder:us-east-1:aws:image/amazon-linux-2-arm64/x.x.x"
```

Molte azioni dell'API EC2 Image Builder coinvolgono più risorse. Per specificare più risorse in una singola istruzione, separale ARNs con virgole.

```
"Resource": [
   "resource1",
   "resource2"
]
```

Chiavi di condizione

Image Builder fornisce chiavi di condizione specifiche del servizio e supporta l'utilizzo di alcune chiavi di condizione globali. Per visualizzare tutte le chiavi di condizione AWS globali, consulta AWS Global Condition Context Keys nella IAM User Guide. Vengono fornite le seguenti chiavi di condizione specifiche del servizio.

generatore di immagini: CreatedResourceTagKeys

Lavora con operatori stringa.

Usa questa chiave per filtrare l'accesso in base alla presenza di chiavi tag nella richiesta. Ciò consente di gestire le risorse create da Image Builder.

Disponibilità: questa chiave è disponibile solo per `CreateInfrastrucutreConfiguration` e `UpdateInfrastructureConfiguration` APIs.

imagebuilder:/CreatedResourceTag<key>

Lavora con operatori stringa.

Utilizzate questa chiave per filtrare l'accesso in base alle coppie chiave-valore dei tag allegate alla risorsa creata da Image Builder. Ciò consente di gestire le risorse di Image Builder tramite tag definiti.

Disponibilità: questa chiave è disponibile solo per `CreateInfrastrucutreConfiguration` e `UpdateInfrastructureConfiguration` APIs.

generatore di immagini: LifecyclePolicyResourceType

Lavora con operatori stringa.

Usa questa chiave per filtrare l'accesso in base al tipo di risorsa Lifecycle specificato nella richiesta.

Il valore di questa chiave può essere o`AMI_IMAGE`. `CONTAINER_IMAGE`

Disponibilità: questa chiave è disponibile solo per il `CreateLifecyclePolicy` e `UpdateLifecyclePolicy`APIs.

Generatore di immagini: EC2 MetadataHttpTokens

Lavora con operatori stringa.

Utilizza questa chiave per filtrare l'accesso in base al requisito del token HTTP dei metadati dell' EC2 istanza specificato nella richiesta.

Questo valore per questa chiave può essere `optional` o`required`.

Disponibilità: questa chiave è disponibile solo per il `CreateInfrastrucutreConfiguration` e `UpdateInfrastructureConfiguration` APIs.

generatore di immagini: StatusTopicArn

Lavora con operatori stringa.

Utilizzate questa chiave per filtrare l'accesso in base all'ARN dell'argomento SNS nella richiesta in cui verranno pubblicate le notifiche sullo stato del terminale.

Disponibilità: questa chiave è disponibile solo per e. `CreateInfrastrucutreConfiguration` `UpdateInfrastructureConfiguration` APIs

Esempi

Per visualizzare esempi di politiche basate sull'identità di Image Builder, vedere. Politiche basate sull'identità di Image Builder

Politiche basate sulle risorse di Image Builder

Le politiche basate sulle risorse specificano quali azioni un determinato principale può eseguire sulla risorsa Image Builder e in quali condizioni. Image Builder supporta politiche di autorizzazione basate sulle risorse per componenti, immagini e ricette di immagini. Le policy basate su risorse consentono di concedere l'autorizzazione all'utilizzo ad altri account per ogni risorsa. È inoltre possibile utilizzare una politica basata sulle risorse per consentire a un AWS servizio di accedere ai componenti, alle immagini e alle ricette di immagini.

Per informazioni su come allegare una policy basata sulle risorse a un componente, un'immagine o una ricetta di immagini, consulta. Condividi le risorse di Image Builder con AWS RAM

> **Note**
>
> Quando si aggiorna una politica delle risorse utilizzando Image Builder, l'aggiornamento verrà visualizzato nella console RAM.

Autorizzazione basata sui tag Image Builder

È possibile allegare tag alle risorse di Image Builder o passare i tag in una richiesta a Image Builder. Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `imagebuilder:ResourceTag/`*`key-name`*,

`aws:RequestTag/key-name` o `aws:TagKeys`. Per ulteriori informazioni sull'etichettatura delle risorse di Image Builder, vedere. Assegna un tag a una risorsa dal AWS CLI

Ruoli IAM di Image Builder

Un ruolo IAM è un'entità interna all'utente Account AWS che dispone di autorizzazioni specifiche.

Utilizzo di credenziali temporanee con Image Builder

È possibile utilizzare credenziali temporanee per effettuare l'accesso con la federazione, assumere un ruolo IAM o un ruolo multi-account. È possibile ottenere credenziali di sicurezza temporanee chiamando operazioni AWS STS API come o. AssumeRoleGetFederationToken

Ruoli collegati ai servizi

I ruoli collegati ai servizi consentono di accedere Servizi AWS alle risorse di altri servizi per completare un'azione per conto dell'utente. I ruoli collegati ai servizi sono visualizzati nell'account IAM e sono di proprietà del servizio. Un utente con accesso amministrativo può visualizzare ma non modificare le autorizzazioni per i ruoli collegati ai servizi.

Image Builder supporta i ruoli collegati ai servizi. Per informazioni sulla creazione o la gestione dei ruoli collegati ai servizi di Image Builder, vedere. Usa i ruoli collegati ai servizi IAM per Image Builder

Ruoli dei servizi

Questa caratteristica consente a un servizio di assumere un ruolo di servizio per conto dell'utente. Questo ruolo consente al servizio di accedere alle risorse in altri servizi per completare un'azione per conto dell'utente. I ruoli dei servizi sono visualizzati nell'account IAM e sono di proprietà dell'account. Ciò significa che un utente con accesso amministrativo può modificare le autorizzazioni per questo ruolo. Tuttavia, il farlo potrebbe pregiudicare la funzionalità del servizio.

Gestisci i perimetri dei dati per l'accesso al download dei bucket S3 in Image Builder

EC2 Image Builder gestisce due classi di bucket S3 di AWS proprietà del servizio che contengono le risorse scaricabili necessarie per eseguire i carichi di lavoro di Image Builder nel tuo account. Se utilizzi perimetri di dati per controllare l'accesso ad Amazon S3 nel tuo ambiente, potresti dover consentire esplicitamente l'accesso a questi bucket. Puoi utilizzare l'ARN del bucket o l'URL del bucket per inserire questi bucket nell'elenco consentito, a seconda di come controlli l'accesso ad Amazon S3.

Script di bootstrap per la gestione dei componenti (obbligatori)

Questo bucket S3 contiene script di bootstrap per configurare l' AWSTOE applicazione sulle istanze utilizzate per creare immagini. EC2 Image Builder richiede l'accesso per scaricare gli script per supportare la creazione e il test di nuove immagini.

- ARN del secchio S3: `arn:<AWS partition>:s3:::ec2imagebuilder-managed-resources-<AWS Region>-prod`
- URL del bucket S3: `https://ec2imagebuilder-managed-resources-<AWS Region>.s3.<AWS Region>.<AWS partition-specific domain name>`

Componenti gestiti

Questo bucket S3 contiene payload di pacchetti per componenti gestiti da Amazon. Image Builder richiede l'accesso per scaricare tutti i componenti gestiti configurati nelle ricette.

- ARN del secchio S3: `arn:<AWS partition>:s3:::ec2imagebuilder-toe-<AWS Region>-prod`
- URL del bucket S3: `https://ec2imagebuilder-toe-<AWS Region>.s3.<AWS Region>.<AWS partition-specific domain name>`

Politiche basate sull'identità di Image Builder

Argomenti

- Le migliori pratiche di policy basate sull'identità
- Utilizzo della console Image Builder

Le migliori pratiche di policy basate sull'identità

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare le risorse di Image Builder nel tuo account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni.AWS Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta Policy gestite da AWSo Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente IAM.

- Applica le autorizzazioni con privilegio minimo: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso a operazioni e risorse è possibile aggiungere una condizione alle tue policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente IAM.
- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta Convalida delle policy per il Sistema di analisi degli accessi IAM nella Guida per l'utente IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta Protezione dell'accesso API con MFA nella Guida per l'utente IAM.

Per maggiori informazioni sulle best practice in IAM, consulta Best practice di sicurezza in IAM nella Guida per l'utente di IAM.

Utilizzo della console Image Builder

Per accedere alla console EC2 Image Builder, è necessario disporre di un set minimo di autorizzazioni. Queste autorizzazioni consentono di elencare e visualizzare i dettagli sulle risorse Image Builder presenti nel tuo Account AWS Se crei una policy basata su identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti e ruoli IAM) associate a tale policy.

Per garantire che le entità IAM possano utilizzare la console Image Builder, è necessario allegare ad esse una delle seguenti policy AWS gestite:

- Policy AWSImageBuilderReadOnlyAccess
- Policy AWSImageBuilderFullAccess

Per ulteriori informazioni sulle politiche gestite da Image Builder, vedere. Usa politiche AWS gestite per EC2 Image Builder

> **Important**
>
> La AWSImageBuilderFullAccesspolicy è necessaria per creare il ruolo collegato al servizio Image Builder. Quando colleghi questa policy a un'entità IAM, devi anche allegare la seguente policy personalizzata e includere le risorse che desideri utilizzare che non sono `imagebuilder` presenti nel nome della risorsa:

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

Non è necessario consentire autorizzazioni minime per la console agli utenti che effettuano chiamate solo verso l' AWS API AWS CLI o la stessa. Al contrario, è possibile accedere solo alle operazioni che soddisfano l'operazione API che stai cercando di eseguire.

Politiche basate sulle risorse di Image Builder

Per informazioni su come creare un componente, vedere. Usa i componenti per personalizzare l'immagine di Image Builder

Limitazione dell'accesso dei componenti Image Builder a indirizzi IP specifici

L'esempio seguente concede a qualsiasi utente le autorizzazioni per eseguire qualsiasi operazione di Image Builder sui componenti. La richiesta deve, tuttavia, avere origine dall'intervallo di indirizzi IP specificati nella condizione.

La condizione di questa istruzione identifica l'intervallo 54.240.143.* di indirizzi IP consentiti per la versione 4 (IPv4) del protocollo Internet, con un'eccezione: 54.240.143.188.

Il `Condition` blocco utilizza le `NotIpAddress` condizioni `IpAddress` and e la chiave condition, che è una chiave di condizione -wide. `aws:SourceIp` AWS Per ulteriori informazioni su queste chiavi di condizione, vedere Specificazione delle condizioni in una politica. I `aws:sourceIp` IPv4 valori utilizzano la notazione CIDR standard. Per ulteriori informazioni, consulta Operatori di condizione con indirizzo IP nella Guida per l'utente di 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"}
      }
    }
  ]
}
```

Usa politiche AWS gestite per EC2 Image Builder

Una policy AWS gestita è una policy autonoma creata e amministrata da. AWS AWS le politiche gestite sono progettate per fornire autorizzazioni per molti casi d'uso comuni, in modo da poter iniziare ad assegnare autorizzazioni a utenti, gruppi e ruoli.

Tieni presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con il privilegio minimo per i tuoi casi d'uso specifici, poiché sono disponibili per tutti i clienti. AWS Ti consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo policy gestite dal cliente specifiche per i tuoi casi d'uso.

Non è possibile modificare le autorizzazioni definite nelle politiche gestite. AWS Se AWS aggiorna le autorizzazioni definite in una politica AWS gestita, l'aggiornamento ha effetto su tutte le identità principali (utenti, gruppi e ruoli) a cui è associata la politica. AWS è più probabile che aggiorni una policy AWS gestita quando ne Servizio AWS viene lanciata una nuova o quando diventano disponibili nuove operazioni API per i servizi esistenti.

Per ulteriori informazioni, consultare Policy gestite da AWSnella Guida per l'utente di IAM.

Policy AWSImageBuilderFullAccess

La AWSImageBuilderFullAccesspolitica garantisce l'accesso completo alle risorse di Image Builder per il ruolo a cui è associato, consentendo al ruolo di elencare, descrivere, creare, aggiornare ed eliminare le risorse di Image Builder. La politica concede inoltre autorizzazioni mirate alle risorse correlate Servizi AWS necessarie, ad esempio, per verificare le risorse o per visualizzare le risorse correnti per l'account in. AWS Management Console

Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- Image Builder: viene concesso l'accesso amministrativo, in modo che il ruolo possa elencare, descrivere, creare, aggiornare ed eliminare le risorse di Image Builder.
- Amazon EC2: viene concesso l'accesso alle azioni di Amazon EC2 Descrivi necessarie per verificare l'esistenza delle risorse o ottenere elenchi di risorse appartenenti all'account.
- IAM: viene concesso l'accesso per ottenere e utilizzare profili di istanza il cui nome contiene «imagebuilder», per verificare l'esistenza del ruolo collegato al servizio Image Builder tramite l'azione `iam:GetRole` API e per creare il ruolo collegato al servizio Image Builder.
- License Manager: viene concesso l'accesso per elencare le configurazioni di licenza o le licenze per una risorsa.
- Amazon S3: è consentito l'accesso ai bucket di elenco appartenenti all'account e anche ai bucket Image Builder con «imagebuilder» nel nome.
- Amazon SNS: le autorizzazioni di scrittura vengono concesse ad Amazon SNS per verificare la proprietà degli argomenti contenenti «imagebuilder».

Per vedere le autorizzazioni per questa policy, consulta AWSImageBuilderFullAccess nella Guida di riferimento sulle policy gestite da AWS .

Policy AWSImageBuilderReadOnlyAccess

La AWSImageBuilderReadOnlyAccesspolicy fornisce l'accesso in sola lettura a tutte le risorse di Image Builder. Vengono concesse le autorizzazioni per verificare l'esistenza del ruolo collegato al servizio Image Builder tramite l'azione API. `iam:GetRole`

Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- Image Builder: l'accesso è concesso per l'accesso in sola lettura alle risorse di Image Builder.
- IAM: viene concesso l'accesso per verificare l'esistenza del ruolo collegato al servizio Image Builder tramite l'azione API. `iam:GetRole`

Per vedere le autorizzazioni per questa policy, consulta AWSImageBuilderReadOnlyAccess nella Guida di riferimento sulle policy gestite da AWS .

Policy AWSServiceRoleForImageBuilder

La AWSServiceRoleForImageBuilderpolicy consente a Image Builder di effettuare chiamate per Servizi AWS conto dell'utente.

Dettagli dell'autorizzazione

Questa policy è associata al ruolo collegato al servizio Image Builder quando il ruolo viene creato tramite Systems Manager. Per ulteriori informazioni sul ruolo collegato al servizio Image Builder, vedere. Usa i ruoli collegati ai servizi IAM per Image Builder

La politica include le seguenti autorizzazioni:

- CloudWatch Registri: l'accesso è concesso per creare e caricare CloudWatch registri in qualsiasi gruppo di log il cui nome inizia con. `/aws/imagebuilder/`
- Amazon EC2 — Image Builder consente di creare, scattare istantanee e registrare le immagini (AMIs) create e avviare EC2 istanze nel tuo account. Image Builder utilizza istantanee, volumi, interfacce di rete, sottoreti, gruppi di sicurezza, configurazione delle licenze e coppie di chiavi correlati come richiesto, purché l'immagine, l'istanza e i volumi che vengono creati o utilizzati siano etichettati con o. `CreatedBy: EC2 Image Builder` `CreatedBy: EC2 Fast Launch`

  Image Builder può ottenere informazioni sulle EC2 immagini di Amazon, gli attributi delle istanze, lo stato delle istanze, i tipi di istanza disponibili per il tuo account, i modelli di avvio, le sottoreti, gli host e i tag sulle tue risorse Amazon. EC2

  Image Builder può aggiornare le impostazioni dell'immagine per abilitare o disabilitare l'avvio più rapido delle istanze di Windows nell'account, dove l'immagine è contrassegnata con. `CreatedBy: EC2 Image Builder`

  Inoltre, Image Builder può avviare, arrestare e terminare le istanze in esecuzione nel tuo account, condividere istantanee di Amazon EBS, creare e aggiornare immagini e avviare modelli, annullare la registrazione di immagini esistenti, aggiungere tag e replicare immagini tra gli account a cui hai concesso le autorizzazioni tramite la policy. Ec2ImageBuilderCrossAccountDistributionAccess Il tagging di Image Builder è necessario per tutte queste azioni, come descritto in precedenza.
- Amazon ECR — Image Builder ha accesso per creare un repository, se necessario, per le scansioni delle vulnerabilità delle immagini dei container e contrassegnare le risorse che crea per limitare l'ambito delle sue operazioni. A Image Builder viene inoltre concesso l'accesso per eliminare le immagini del contenitore che ha creato per le scansioni dopo aver scattato istantanee delle vulnerabilità.

- EventBridge— A Image Builder viene concesso l'accesso per creare e gestire EventBridge regole.
- IAM: Image Builder consente di trasferire qualsiasi ruolo del tuo account ad Amazon e a VM EC2 Import/Export.
- Amazon Inspector: a Image Builder viene concesso l'accesso per determinare quando Amazon Inspector completa le scansioni delle istanze di build e per raccogliere i risultati delle immagini configurate per consentirlo.
- AWS KMS— L'accesso è concesso ad Amazon EBS per crittografare, decrittografare o ricrittografare i volumi Amazon EBS. Questo è fondamentale per garantire il funzionamento dei volumi crittografati quando Image Builder crea un'immagine.
- License Manager: Image Builder consente di aggiornare le specifiche del License Manager tramite. `license-manager:UpdateLicenseSpecificationsForResource`
- Amazon SNS: le autorizzazioni di scrittura sono concesse per qualsiasi argomento di Amazon SNS nel tuo account.
- Systems Manager: a Image Builder viene concesso l'accesso per elencare i comandi di Systems Manager e le relative chiamate, le voci di inventario, descrivere le informazioni sull'istanza e gli stati di esecuzione dell'automazione, descrivere gli host per il supporto del posizionamento delle istanze e ottenere dettagli sulla chiamata dei comandi. Image Builder può anche inviare segnali di automazione e interrompere le esecuzioni di automazione per qualsiasi risorsa del tuo account.

  Image Builder è in grado di emettere chiamate di comando run a qualsiasi istanza contrassegnata `"CreatedBy": "EC2 Image Builder"` per i seguenti file di script:`AWS-RunPowerShellScript`,, o. `AWS-RunShellScript` `AWSEC2-RunSysprep` Image Builder è in grado di avviare un'esecuzione di automazione di Systems Manager nell'account per i documenti di automazione il cui nome inizia con. `ImageBuilder`

  Image Builder è anche in grado di creare o eliminare associazioni di State Manager per qualsiasi istanza dell'account, purché il documento di associazione sia presente`AWS-GatherSoftwareInventory`, e di creare il ruolo collegato al servizio Systems Manager nell'account.
- AWS STS— A Image Builder viene concesso l'accesso per assumere i ruoli denominati EC2ImageBuilderDistributionCrossAccountRoledall'account a qualsiasi account in cui la politica di attendibilità relativa al ruolo lo consenta. Viene utilizzato per la distribuzione di immagini tra più account.

Per vedere le autorizzazioni per questa policy, consulta AWSServiceRoleForImageBuilder nella Guida di riferimento sulle policy gestite da AWS .

Policy Ec2ImageBuilderCrossAccountDistributionAccess

La Ec2ImageBuilderCrossAccountDistributionAccesspolitica concede a Image Builder le autorizzazioni per distribuire immagini tra account nelle regioni di destinazione. Inoltre, Image Builder può descrivere, copiare e applicare tag a qualsiasi EC2 immagine Amazon nell'account. La policy garantisce inoltre la possibilità di modificare le autorizzazioni AMI tramite l'azione `ec2:ModifyImageAttribute` API.

Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- Amazon EC2: Amazon ha accesso per EC2 descrivere, copiare e modificare gli attributi di un'immagine e per creare tag per qualsiasi EC2 immagine Amazon presente nell'account.

Per vedere le autorizzazioni per questa policy, consulta Ec2ImageBuilderCrossAccountDistributionAccess nella Guida di riferimento sulle policy gestite da AWS .

Policy EC2ImageBuilderLifecycleExecutionPolicy

La EC2ImageBuilderLifecycleExecutionPolicypolitica concede a Image Builder le autorizzazioni per eseguire azioni come deprecare, disabilitare o eliminare le risorse di immagine di Image Builder e le relative risorse sottostanti AMIs (istantanee) per supportare regole automatizzate per le attività di gestione del ciclo di vita delle immagini.

Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- Amazon EC2: ad Amazon viene concesso l'accesso EC2 per eseguire le seguenti azioni per Amazon Machine Images (AMIs) nell'account contrassegnato`CreatedBy: EC2 Image Builder`.
  - Abilita e disabilita un AMI.
  - Abilita e disabilita la deprecazione delle immagini.
  - Descrivi e annulla la registrazione di un AMI.

  - Descrivi e modifica gli attributi delle immagini AMI.
  - Elimina le istantanee del volume associate all'AMI.
  - Recupera i tag per una risorsa.
  - Aggiungi o rimuovi tag da un'AMI per renderli obsoleti.
- Amazon ECR: Amazon ECR consente ad Amazon ECR di eseguire le seguenti azioni batch sui repository ECR con il tag. `LifecycleExecutionAccess: EC2 Image Builder` Le azioni Batch supportano le regole automatizzate del ciclo di vita delle immagini dei container.
  - `ecr:BatchGetImage`
  - `ecr:BatchDeleteImage`

  L'accesso è concesso a livello di repository per gli archivi ECR contrassegnati con. `LifecycleExecutionAccess: EC2 Image Builder`
- AWS Gruppi di risorse: a Image Builder viene concesso l'accesso per ottenere risorse basate sui tag.
- EC2 Image Builder: Image Builder ha accesso per eliminare le risorse di immagini di Image Builder.

Per vedere le autorizzazioni per questa policy, consulta EC2ImageBuilderLifecycleExecutionPolicy nella Guida di riferimento sulle policy gestite da AWS .

Policy EC2InstanceProfileForImageBuilder

La EC2InstanceProfileForImageBuilderpolicy concede le autorizzazioni minime necessarie affinché un' EC2 istanza funzioni con Image Builder. Ciò non include le autorizzazioni necessarie per utilizzare l'agente Systems Manager.

Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- CloudWatch Registri: l'accesso è concesso per creare e caricare CloudWatch registri in qualsiasi gruppo di log il cui nome inizia con. `/aws/imagebuilder/`
- Amazon EC2 — Access è concesso per descrivere volumi e istantanee, per creare istantanee di volumi o risorse snapshot create da Image Builder e per creare tag per le risorse Image Builder.
- Image Builder: l'accesso è concesso per ottenere qualsiasi Image Marketplace AWS Builder o componente.

- AWS KMS— L'accesso è concesso per decrittografare un componente Image Builder, se è stato crittografato tramite. AWS KMS
- Amazon S3: l'accesso è concesso per ottenere oggetti archiviati in un bucket Amazon S3 il cui nome inizia `ec2imagebuilder-` con o risorse con estensione di file ISO.

Per vedere le autorizzazioni per questa policy, consulta EC2InstanceProfileForImageBuilder nella Guida di riferimento sulle policy gestite da AWS .

Policy EC2InstanceProfileForImageBuilderECRContainerBuilds

La EC2InstanceProfileForImageBuilderECRContainerBuildspolicy concede le autorizzazioni minime richieste per un' EC2 istanza quando si lavora con Image Builder per creare immagini Docker e quindi registrare e archiviare le immagini in un repository di contenitori Amazon ECR. Ciò non include le autorizzazioni necessarie per utilizzare l'agente Systems Manager.

Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- CloudWatch Registri: l'accesso è concesso per creare e caricare CloudWatch registri in qualsiasi gruppo di log il cui nome inizia con. `/aws/imagebuilder/`
- Amazon ECR: Amazon ECR consente di ottenere, registrare e archiviare un'immagine del contenitore e ottenere un token di autorizzazione.
- Image Builder: è consentito l'accesso per ottenere un componente Image Builder o una ricetta del contenitore.
- AWS KMS— L'accesso è concesso per decrittografare un componente Image Builder o una ricetta contenitore, se è stato crittografato tramite. AWS KMS
- Amazon S3: l'accesso è concesso per archiviare oggetti in un bucket Amazon S3 il cui nome inizia con. `ec2imagebuilder-`

Per vedere le autorizzazioni per questa policy, consulta EC2InstanceProfileForImageBuilderECRContainerBuilds nella Guida di riferimento sulle policy gestite da AWS .

Image Builder aggiorna le politiche gestite AWS

Questa sezione fornisce informazioni sugli aggiornamenti delle politiche AWS gestite per Image Builder da quando questo servizio ha iniziato a tenere traccia di queste modifiche. Per ricevere avvisi

automatici sulle modifiche a questa pagina, abbonatevi al feed RSS nella pagina della cronologia dei documenti di Image Builder.

Modifica	Descrizione	Data
AWSServiceRoleForImageBuilder: aggiornamento a una policy esistente	Image Builder ha apportato le seguenti modifiche al ruolo di servizio per supportare l'importazione di file ISO del sistema operativo client Microsoft come immagine di base. • Aggiunto ec2: RegisterImage per consentire a Image Builder di creare un'istantanea e creare e registrare un AMI il cui sistema operativo di base è stato importato da file disco ISO verificati.	30 dicembre 2024
EC2InstanceProfileForImageBuilder: aggiornamento a una policy esistente	Image Builder ha apportato le seguenti modifiche alla politica del profilo dell'istanza per supportare la creazione di immagini dai file di immagine del disco. • Sono state aggiunte le seguenti azioni ec2: DescribeVolumes e DescribeSnapshots su tutte le risorse per recuperare i dettagli. Sono state inoltre aggiunte le seguenti azioni	30 dicembre 2024

Modifica	Descrizione	Data
	per le risorse create da Image Builder: CreateSnapshot per le risorse di volume e istantanee e. CreateTags Aggiunto s3: GetObject per le risorse con estensione di file «ISO».	
EC2InstanceProfileForImageBuilder— Politica aggiornata	Image Builder ha aggiornato la `EC2InstanceProfileForImageBuilder` politica per consentire a Image Builder di ottenere componenti. Marketplace AWS	2 dicembre 2024
EC2ImageBuilderLifecycleExecutionPolicy: nuova policy	Image Builder ha aggiunto la nuova `EC2ImageBuilderLifecycleExecutionPolicy` politica che contiene le autorizzazioni per la gestione del ciclo di vita delle immagini.	17 novembre 2023

Modifica	Descrizione	Data
AWSServiceRoleForImageBuilder: aggiornamento a una policy esistente	Image Builder ha apportato le seguenti modifiche al ruolo di servizio per fornire supporto per il posizionamento delle istanze. • Aggiunto ec2: DescribeHosts consente a Image Builder di eseguire il polling dell'HostID per determinare quando è in uno stato valido per avviare un'istanza. • Aggiunta l'azione ssm:GetCommandInvocation, API per migliorare il metodo utilizzato da Image Builder per ottenere dettagli sulla chiamata del comando.	19 ottobre 2023

Modifica	Descrizione	Data
AWSServiceRoleForImageBuilder: aggiornamento a una policy esistente	Image Builder ha apportato le seguenti modifiche al ruolo di servizio per fornire supporto per il posizionamento delle istanze. • Aggiunto ec2: DescribeHosts abilita Image Builder per eseguire il polling dell'HostID per determinare quando è in uno stato valido per avviare un'istanza. • Aggiunta l'azione ssm:GetCommandInvocation, API per migliorare il metodo utilizzato da Image Builder per ottenere dettagli sulla chiamata del comando.	28 settembre 2023

Modifica	Descrizione	Data
AWSServiceRoleForImageBuilder: aggiornamento a una policy esistente	Image Builder ha apportato le seguenti modifiche al ruolo di servizio per consentire ai flussi di lavoro di Image Builder di raccogliere i risultati delle vulnerabilità per le build di immagini dei contenitori AMI ed ECR. Le nuove autorizzazioni supportano la funzionalità di rilevamento e segnalazione CVE. • Sono stati aggiunti inspector2: ListCoverage e inspector2: per consentire a ListFindings Image Builder di determinare quando Amazon Inspector completa le scansioni delle istanze di test e per raccogliere i risultati per le immagini configurate per consentirlo. • Aggiunto ecr:CreateRepository, con l'obbligo per Image Builder di etichettare il repository `CreatedBy: EC2 Image Builder` con (). tag-on-create È stato inoltre aggiunto ecr: TagResource (obbligatorio per tag-on-create) con lo stesso vincolo di CreatedBy tag e un vincolo	30 marzo 2023

Modifica	Descrizione	Data
	aggiuntivo che richiede il nome del repository con cui iniziare. `image-builder-*` Il vincolo del nome impedisce l'aumento dei privilegi e impedisce le modifiche ai repository non creati da Image Builder. • Aggiunto ecr: per i repository ECR contrassegnati con. BatchDeleteImage `CreatedBy: EC2 Image Builder` Questa autorizzazione richiede il nome del repository con cui iniziare. `image-builder-*` • Sono state aggiunte le autorizzazioni agli eventi per Image Builder per creare e gestire le regole gestite da EventBridge Amazon `ImageBuilder-*` incluse nel nome.	

Modifica	Descrizione	Data
AWSServiceRoleForImageBuilder: aggiornamento a una policy esistente	Image Builder ha apportato le seguenti modifiche al ruolo di servizio: • Aggiunte le licenze License Manager come risorsa per la RunInstance chiamata ec2: per consentire ai clienti di utilizzare l'immagine AMIs di base associata a una configurazione di licenza.	22 marzo 2022
AWSServiceRoleForImageBuilder: aggiornamento a una policy esistente	Image Builder ha apportato le seguenti modifiche al ruolo di servizio: • Sono state aggiunte le autorizzazioni per l'azione delle EC2 EnableFastLaunch API, per abilitare e disabilitare l'avvio più rapido per le istanze Windows. • Ambito di applicazione più ristretto per ec2: condizioni per i tag delle CreateTags azioni e delle risorse.	21 febbraio 2022

Modifica	Descrizione	Data
AWSServiceRoleForImageBuilder: aggiornamento a una policy esistente	Image Builder ha apportato le seguenti modifiche al ruolo di servizio: • Sono state aggiunte le autorizzazioni per chiamare il servizio VMIE per importare una VM e creare da essa un'AMI di base. • Ambito di applicazione più ristretto per ec2: condizioni per i tag delle CreateTags azioni e delle risorse.	20 novembre 2021
AWSServiceRoleForImageBuilder: aggiornamento a una policy esistente	Image Builder ha aggiunto nuove autorizzazioni per risolvere i problemi in cui più di un'associazione di inventario causa il blocco della creazione dell'immagine.	11 agosto 2021

Modifica	Descrizione	Data
AWSImageBuilderFullAccess: aggiornamento a una policy esistente	Image Builder ha apportato le seguenti modifiche al ruolo di accesso completo: • Autorizzazioni aggiunte da consentire. `ec2:DescribeInstanceTypeOfferings` • Sono state aggiunte le autorizzazioni `ec2:DescribeInstanceTypeOfferings` alla chiamata per consentire alla console Image Builder di riflettere accuratamente i tipi di istanza disponibili nell'account.	13 aprile 2021
Image Builder ha iniziato a tracciare le modifiche	Image Builder ha iniziato a tenere traccia delle modifiche per le sue politiche AWS gestite.	02 aprile 2021

Usa i ruoli collegati ai servizi IAM per Image Builder

EC2 Image Builder utilizza ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato ai servizi è un tipo unico di ruolo IAM collegato direttamente a Image Builder. I ruoli collegati ai servizi sono predefiniti da Image Builder e includono tutte le autorizzazioni richieste dal servizio per chiamare altri utenti per conto dell'utente. Servizi AWS

Un ruolo collegato al servizio rende la configurazione di Image Builder più efficiente, poiché non è necessario aggiungere manualmente le autorizzazioni necessarie. Image Builder definisce le autorizzazioni dei suoi ruoli collegati ai servizi e, se non diversamente definito, solo Image Builder

può assumerne i ruoli. Le autorizzazioni definite includono policy di trust e di autorizzazioni. Le policy di autorizzazioni non possono essere collegate a nessun'altra entità IAM.

Per informazioni su altri servizi che supportano i ruoli collegati ai servizi, consulta Servizi AWS That Work with IAM e cerca i servizi con Sì nella colonna Service-Linked Role. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Autorizzazioni di ruolo collegate ai servizi per Image Builder

Image Builder utilizza il ruolo AWSServiceRoleForImageBuildercollegato al servizio per consentire a Image Builder di accedere alle EC2 risorse per tuo conto. AWS Il ruolo collegato al servizio si fida che il servizio imagebuilder.amazonaws.com assuma il ruolo.

Non è necessario creare manualmente questo ruolo collegato ai servizi. Quando create la vostra prima immagine di Image Builder nella AWS Management Console, o nell' AWS API AWS CLI, Image Builder crea automaticamente il ruolo collegato al servizio.

Le seguenti azioni creano una nuova immagine:

- Esegui la procedura guidata per la pipeline nella console Image Builder per creare un'immagine personalizzata.
- Utilizza una delle seguenti azioni API o il comando corrispondente: AWS CLI
  - L'azione CreateImage API (create-imagein AWS CLI).
  - L'azione ImportVmImage API (import-vm-imagein AWS CLI).
  - L'azione StartImagePipelineExecution API (start-image-pipeline-executionin AWS CLI).

> **Important**
>
> Se il ruolo collegato al servizio viene eliminato dal tuo account, puoi utilizzare la stessa procedura per crearlo nuovamente. Quando crei la tua prima risorsa EC2 Image Builder, Image Builder crea nuovamente il ruolo collegato al servizio per te.

Per visualizzare le autorizzazioni per la, consulta la pagina AWSServiceRoleForImageBuilder. Policy AWSServiceRoleForImageBuilder Per ulteriori informazioni sulla configurazione delle autorizzazioni per un ruolo collegato al servizio, consulta Service-Linked Role Permissions nella IAM User Guide.

Rimuovere un ruolo collegato al servizio Image Builder dall'account

Puoi utilizzare la console IAM AWS CLI, o l' AWS API per rimuovere manualmente il ruolo collegato al servizio per Image Builder dal tuo account. Tuttavia, prima di eseguire questa operazione, è necessario assicurarsi che non vi siano risorse Image Builder abilitate che fanno riferimento ad esso.

> **Note**
>
> Se il servizio Image Builder utilizza il ruolo quando si tenta di eliminare le risorse, l'eliminazione potrebbe non riuscire. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Pulisci le risorse di Image Builder utilizzate dal ruolo **`AWSServiceRoleForImageBuilder`**

1. Verificate che nessuna build di pipeline sia in esecuzione prima di iniziare. Per annullare una build in esecuzione, utilizzate il `cancel-image-creation` comando di. AWS CLI

   ```
   aws imagebuilder cancel-image-creation --image-build-version-
   arn arn:aws:imagebuilder:us-east-1:123456789012:image-pipeline/sample-pipeline
   ```

2. Modifica tutte le pianificazioni della pipeline per utilizzare un processo di compilazione manuale o eliminale se non le utilizzerai più. Per ulteriori informazioni sull'eliminazione delle risorse, consulta. Eliminare risorse Image Builder obsolete o inutilizzate

Elimina il ruolo collegato al servizio utilizzando IAM

Puoi utilizzare la console IAM AWS CLI, o l' AWS API per eliminare il `AWSServiceRoleForImageBuilder` ruolo dal tuo account. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato ai servizi nella Guida per l'utente IAM.

Regioni supportate per i ruoli collegati ai servizi di Image Builder

Image Builder supporta l'utilizzo di ruoli collegati ai servizi in tutte le AWS regioni in cui il servizio è disponibile. Per l'elenco delle AWS regioni supportate, vedere. AWS Regioni ed endpoint

Risolvi i problemi di IAM in Image Builder

Argomenti

- Non sono autorizzato a eseguire un'azione in Image Builder

- Non sono autorizzato a eseguire iam: PassRole
- Voglio consentire a persone esterne a me di accedere Account AWS alle mie risorse di Image Builder

Non sono autorizzato a eseguire un'azione in Image Builder

Se ricevi un errore che indica che non sei autorizzato a eseguire un'operazione, le tue policy devono essere aggiornate per poter eseguire l'operazione.

L'errore di esempio seguente si verifica quando l'utente IAM `mateojackson` prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa *`my-example-widget`* fittizia ma non dispone di autorizzazioni `imagebuilder:`*`GetWidget`* fittizie.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 imagebuilder:GetWidget on resource: my-example-widget
```

In questo caso, la policy per l'utente `mateojackson` deve essere aggiornata per consentire l'accesso alla risorsa *`my-example-widget`* utilizzando l'azione `imagebuilder:`*`GetWidget`*.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Non sono autorizzato a eseguire iam: PassRole

Se ricevi un messaggio di errore indicante che non sei autorizzato a eseguire l'`iam:PassRole`azione, le tue politiche devono essere aggiornate per consentirti di passare un ruolo a Image Builder.

Alcuni Servizi AWS consentono di trasferire un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

Il seguente errore di esempio si verifica quando un utente IAM denominato `marymajor` tenta di utilizzare la console per eseguire un'azione in Image Builder. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Voglio consentire a persone esterne a me di accedere Account AWS alle mie risorse di Image Builder

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per i servizi che supportano politiche basate sulle risorse o elenchi di controllo degli accessi (ACLs), è possibile utilizzare tali criteri per concedere alle persone l'accesso alle proprie risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se Image Builder supporta queste funzionalità, vedere. Come Image Builder funziona con le politiche e i ruoli IAM
- Per scoprire come fornire l'accesso alle tue risorse su tutto Account AWS ciò che possiedi, consulta Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà nella IAM User Guide.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta Fornire l'accesso a soggetti Account AWS di proprietà di terze parti nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta Fornire l'accesso a utenti autenticati esternamente (Federazione delle identità) nella Guida per l'utente IAM.
- Per informazioni sulle differenze di utilizzo tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Risorse per la convalida della conformità per Image Builder

EC2 Image Builder non rientra nell'ambito di alcun programma di AWS conformità.

Per un elenco dei Servizi AWS programmi di conformità specifici, vedere AWS Servizi nell'ambito del programma Servizi AWS di conformità in Ambito di . Per informazioni generali, vedere Programmi di AWS conformità Programmi di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact Scaricamento dei . AWS

La responsabilità della conformità quando si utilizza Image Builder è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- Security and Compliance Quick Start Guides (Guide Quick Start Sicurezza e compliance): queste guide alla distribuzione illustrano considerazioni relative all'architettura e forniscono procedure per la distribuzione di ambienti di base incentrati sulla sicurezza e sulla conformità su AWS.
- AWS Risorse per la per la conformità: questa raccolta di cartelle di lavoro e guide potrebbe riguardare il settore e la località in cui operi.
- Valutazione delle risorse con le regole nella Guida per gli AWS Config sviluppatori: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.
- AWS Security Hub— Questo AWS servizio offre una visione completa dello stato di sicurezza dell'utente, AWS che consente di verificare la conformità agli standard e alle best practice del settore della sicurezza.

È possibile incorporare prodotti Marketplace AWS o componenti di AWS Task Orchestrator and Executor (AWSTOE) nelle immagini di Image Builder per garantire la conformità delle immagini. Per ulteriori informazioni, consulta Prodotti di conformità per le immagini di Image Builder.

Ridondanza e resilienza dei dati in Image Builder

L'infrastruttura AWS globale è costruita attorno AWS a regioni e zone di disponibilità. AWS Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, puoi progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Il servizio EC2 Image Builder consente di distribuire immagini create in una regione tra altre regioni, offrendo loro una resilienza multiregionale per. AMIs Non esiste alcun meccanismo per «eseguire il backup» di pipeline, ricette o componenti di immagini. Puoi archiviare i documenti delle ricette e dei componenti all'esterno del servizio Image Builder, ad esempio in un bucket Amazon S3.

L' EC2 Image Builder non può essere configurato per l'High Availability (HA). È possibile distribuire le immagini in più regioni per renderle più disponibili.

Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta AWS Global Infrastructure.

Sicurezza dell'infrastruttura in Image Builder

La rete AWS globale offre funzionalità di sicurezza e controlla l'accesso alla rete per servizi come EC2 Image Builder. Per ulteriori informazioni sulla sicurezza dell'infrastruttura che AWS fornisce i suoi servizi, consulta la sezione Sicurezza dell'infrastruttura nel white paper Introduzione alla AWS sicurezza.

Per inviare richieste attraverso la rete AWS globale per le azioni dell'API Image Builder, il software client deve rispettare le seguenti linee guida di sicurezza:

- Per inviare richieste di azioni API Image Builder, il software client deve utilizzare una versione supportata di Transport Layer Security (TLS).

  > **Note**
  >
  > AWS sta eliminando gradualmente il supporto per le versioni TLS 1.0 e 1.1. Ti consigliamo vivamente di aggiornare il software client per utilizzare la versione TLS 1.2 o successiva in modo da poter continuare a connetterti. Per ulteriori informazioni, consulta questo post sul blog AWS sulla sicurezza.

- Il software client deve supportare suite di crittografia con Perfect Forward Secrecy (PFS), come Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi attuali, come Java 7 e versioni successive, supporta queste modalità.
- È necessario firmare le richieste API con un ID di chiave di accesso e una chiave di accesso segreta associata a un principale AWS Identity and Access Management (IAM). Oppure puoi usare il AWS Security Token Service(AWS STS) per generare credenziali di sicurezza temporanee per le tue richieste.

Inoltre, le EC2 istanze utilizzate da Image Builder per creare e testare le immagini devono avere accesso. AWS Systems Manager

Gestione delle patch per le immagini di Image Builder

AWS fornisce aggiornamenti gestiti AMIs ogni mese con gli ultimi aggiornamenti e patch di sicurezza applicati per i seguenti sistemi operativi. È possibile utilizzarli AMIs come immagine di base per le personalizzazioni. Per ulteriori informazioni, consulta Sistemi operativi supportati.

- Distribuzioni Linux tra cui Amazon Linux 2, AL2 023, Red Hat Enterprise Linux (RHEL), CentOS, Ubuntu, SUSE Linux Enterprise Server
- Windows Server 2016 e versioni successive
- macOS 10.14.x e versioni successive

Dopo aver creato un'immagine personalizzata, sei responsabile dell'applicazione delle patch al EC2 sistema Amazon, secondo il modello di responsabilità condivisa. Se le EC2 istanze del carico di lavoro dell'applicazione possono essere sostituite facilmente, potrebbe essere più efficiente aggiornare l'AMI di base e ridistribuire tutti i nodi di calcolo in base a questa immagine.

> **Note**
>
> Per l'applicazione delle patch su macOS, ti consigliamo di creare una nuova versione della ricetta che utilizzi l'ultima AMI gestita per l'immagine di base, quindi di creare un'immagine personalizzata aggiornata dalla ricetta e dalle altre risorse di creazione dell'immagine. Se le istanze Mac non possono essere sostituite facilmente, consulta la pagina Aggiornamento del sistema operativo e del software sulle istanze Mac nella Amazon EC2 User Guide per ulteriori informazioni.

Di seguito sono riportati due modi per mantenere aggiornato Image Builder AMIs .

- AWS-provided patching components — Image EC2 Builder fornisce i seguenti componenti di build che installano tutti gli aggiornamenti in sospeso del sistema operativo:
  - `update-linux`
  - `update-windows`

  Questi componenti utilizzano il modulo di azione. `UpdateOS` Per ulteriori informazioni, consulta Aggiorna il sistema operativo. I componenti possono essere aggiunti alle pipeline di creazione dell'immagine selezionandoli dall'elenco dei componenti AWS forniti.

- Componenti di compilazione personalizzati con operazioni di patching: per installare o aggiornare selettivamente le patch sui sistemi operativi supportati AMIs, è possibile creare un componente Image Builder per installare le patch richieste. Un componente personalizzato può installare le patch utilizzando script di shell (Bash o PowerShell) oppure può utilizzare il modulo di `UpdateOS` azione per specificare le patch per l'installazione o l'esclusione. Per ulteriori informazioni, consulta Moduli di azione supportati dal gestore AWSTOE dei componenti.

  Componente che utilizza il modulo di `UpdateOS` azione (solo Linux e Windows). Il modulo di `UpdateOS` azione non è supportato per macOS.)

```
schemaVersion: 1.0
phases:
  - name: build
 steps:
   - name: UpdateOS
  action: UpdateOS
```

  Componente che utilizza Bash per installare gli aggiornamenti yum

```
schemaVersion: 1.0
phases:
  - name: build
 steps:
   - name: InstallYumUpdates
  action: ExecuteBash
  inputs:
    commands:
   - sudo yum update -y
```

Best practice di sicurezza per Image Builder

EC2 Image Builder offre una serie di funzionalità di sicurezza da prendere in considerazione durante lo sviluppo e l'implementazione delle proprie politiche di sicurezza. Le seguenti best practice sono linee guida generali e non rappresentano una soluzione di sicurezza completa. Poiché queste best practice potrebbero non essere appropriate o sufficienti per l'ambiente, gestiscile come considerazioni utili anziché prescrizioni.

- Non utilizzare gruppi di sicurezza eccessivamente permissivi nelle ricette di Image Builder.

- Non condividete immagini con account di cui non vi fidate.
- Non rendere pubbliche le immagini che contengono dati privati o sensibili.
- Applica tutte le patch di sicurezza disponibili per Windows o Linux durante la creazione delle immagini.
- Applica periodicamente gli aggiornamenti delle AMI gestite alle tue ricette macOS e crea nuove immagini per avviare istanze con le patch di sicurezza più recenti.

Ti consigliamo vivamente di testare le immagini per convalidare il livello di sicurezza e i livelli di conformità di sicurezza applicabili. Soluzioni come Amazon Inspector possono aiutare a convalidare il livello di sicurezza e conformità delle immagini.

IMDSv2 per le pipeline di Image Builder

Quando viene eseguita la pipeline di Image Builder, invia richieste HTTP alle EC2 istanze di avvio utilizzate da Image Builder per creare e testare l'immagine. Per configurare la versione di IMDS utilizzata dalla pipeline per le richieste di avvio, imposta il `httpTokens` parametro nelle impostazioni dei metadati dell'istanza di configurazione dell'infrastruttura Image Builder.

> **Note**
>
> Si consiglia di configurare tutte le EC2 istanze che Image Builder avvia da una build di pipeline da IMDSv2 utilizzare in modo che le richieste di recupero dei metadati dell'istanza richiedano un'intestazione token firmata.

Per ulteriori informazioni sulla configurazione dell'infrastruttura Image Builder, vedere. Gestione della configurazione dell'infrastruttura Image Builder Per ulteriori informazioni sulle opzioni dei metadati delle EC2 istanze per le immagini Linux, consulta Configurare le opzioni dei metadati dell'istanza nella Amazon EC2 User Guide. Per le immagini Windows, consulta Configurare le opzioni dei metadati dell'istanza nella Amazon EC2 User Guide.

Pulizia post-compilazione richiesta

Dopo che Image Builder ha completato tutti i passaggi di creazione dell'immagine personalizzata, Image Builder prepara l'istanza di compilazione per il test e la creazione di immagini. Prima di chiudere l'istanza di compilazione per creare l'istantanea, Image Builder esegue la seguente pulizia per garantire la sicurezza dell'immagine:

Linux

La pipeline Image Builder esegue uno script di pulizia per garantire che l'immagine finale segua le migliori pratiche di sicurezza e per rimuovere eventuali artefatti o impostazioni di build che non devono essere trasferiti all'istantanea. Tuttavia, è possibile saltare alcune sezioni dello script o sovrascrivere completamente i dati dell'utente. Pertanto, le immagini prodotte dalle pipeline di Image Builder non sono necessariamente conformi a criteri normativi specifici.

Quando la pipeline completa le fasi di compilazione e test, Image Builder esegue automaticamente il seguente script di pulizia appena prima di creare l'immagine di output.

> **Important**
>
> Se sovrascrivi i dati utente nella tua ricetta, lo script non viene eseguito. In tal caso, assicurati di includere nei dati utente un comando che crei un file vuoto denominato`perform_cleanup`. Image Builder rileva questo file ed esegue lo script di pulizia prima di creare la nuova immagine.

```
#!/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

Dopo che la pipeline Image Builder ha personalizzato le immagini di Windows, esegue l'utilità Microsoft Sysprep. Queste azioni seguono le AWS migliori pratiche per il rafforzamento e la pulizia dell'immagine.

macOS

La pipeline Image Builder esegue uno script di pulizia per garantire che l'immagine finale segua le migliori pratiche di sicurezza e per rimuovere eventuali artefatti o impostazioni di build che non devono essere trasferiti all'istantanea. Tuttavia, è possibile saltare alcune sezioni dello script o sovrascrivere completamente i dati dell'utente. Pertanto, le immagini prodotte dalle pipeline di Image Builder non sono necessariamente conformi a criteri normativi specifici.

Quando la pipeline completa le fasi di compilazione e test, Image Builder esegue automaticamente il seguente script di pulizia appena prima di creare l'immagine di output.

> **Important**
>
> Se sovrascrivi i dati utente nella tua ricetta, lo script non viene eseguito. In tal caso, assicurati di includere nei dati utente un comando che crei un file vuoto denominato`perform_cleanup`. Image Builder rileva questo file ed esegue lo script di pulizia prima di creare la nuova immagine.

```
#!/bin/bash
if [[ ! -f {{workingDirectory}}/perform_cleanup ]]; then
  echo "Skipping cleanup"
  exit 0
else
  sudo rm -f {{workingDirectory}}/perform_cleanup
```

```
fi

function cleanup() {
  FILES=("$@")
  for FILE in "${FILES[@]}"; do
      if [[ -f "$FILE" ]]; then
          echo "Deleting $FILE";
          sudo rm -f $FILE;
      fi;
      if [[ -f $FILE ]]; then
          echo "Failed to delete '$FILE'. Failing."
          exit 1
      fi;
  done
};

# Clean up for cloud-init files
CLOUD_INIT_FILES=(
  "/etc/sudoers.d/90-cloud-init-users"
  "/etc/locale.conf"
  "/var/log/cloud-init.log"
  "/var/log/cloud-init-output.log"
)
if [[ -f {{workingDirectory}}/skip_cleanup_cloudinit_files ]]; then
  echo "Skipping cleanup of cloud init files"
else
  echo "Cleaning up cloud init files"
  cleanup "${CLOUD_INIT_FILES[@]}"
  if [[ $( sudo find /var/lib/cloud -type f | sudo wc -l ) -gt 0 ]]; then
      echo "Deleting files within /var/lib/cloud/*"
      sudo find /var/lib/cloud -type f -exec rm -f {} \;
  fi;

  if [[ $( sudo ls /var/lib/cloud | sudo wc -l ) -gt 0 ]]; then
      echo "Deleting /var/lib/cloud/*"
      sudo rm -rf /var/lib/cloud/* || true
  fi;
fi;


# Clean up for temporary instance files
INSTANCE_FILES=(
  "/etc/.updated"
  "/etc/aliases.db"
```

```
  "/etc/hostname"
  "/var/lib/misc/postfix.aliasesdb-stamp"
  "/var/lib/postfix/master.lock"
  "/var/spool/postfix/pid/master.pid"
  "/var/.updated"
  "/var/cache/yum/x86_64/2/.gpgkeyschecked.yum"
)
if [[ -f {{workingDirectory}}/skip_cleanup_instance_files ]]; then
  echo "Skipping cleanup of instance files"
else
  echo "Cleaning up instance files"
  cleanup "${INSTANCE_FILES[@]}"
fi;


# Clean up for ssh files
SSH_FILES=(
  "/etc/ssh/ssh_host_rsa_key"
  "/etc/ssh/ssh_host_rsa_key.pub"
  "/etc/ssh/ssh_host_ecdsa_key"
  "/etc/ssh/ssh_host_ecdsa_key.pub"
  "/etc/ssh/ssh_host_ed25519_key"
  "/etc/ssh/ssh_host_ed25519_key.pub"
  "/root/.ssh/authorized_keys"
)
if [[ -f {{workingDirectory}}/skip_cleanup_ssh_files ]]; then
  echo "Skipping cleanup of ssh files"
else
  echo "Cleaning up ssh files"
  cleanup "${SSH_FILES[@]}"
  USERS=$(ls /home/)
  for user in $USERS; do
      echo Deleting /home/"$user"/.ssh/authorized_keys;
      sudo find /home/"$user"/.ssh/authorized_keys -type f -exec rm -f {} \;
  done
  for user in $USERS; do
      if [[ -f /home/"$user"/.ssh/authorized_keys ]]; then
          echo Failed to delete /home/"$user"/.ssh/authorized_keys;
          exit 1
      fi;
  done;
fi;
```

```
# Clean up for instance log files
INSTANCE_LOG_FILES=(
  "/var/log/audit/audit.log"
  "/var/log/boot.log"
  "/var/log/dmesg"
  "/var/log/cron"
  "/var/log/amazon/ec2/ec2-macos-init.log"
  "/var/log/amazon/ec2/ena-ethernet.log"
  "/var/log/amazon/ec2/system-monitoring.log"
)
if [[ -f {{workingDirectory}}/skip_cleanup_instance_log_files ]]; then
  echo "Skipping cleanup of instance log files"
else
  echo "Cleaning up instance log files"
  cleanup "${INSTANCE_LOG_FILES[@]}"
fi;

# Clean up for TOE files
if [[ -f {{workingDirectory}}/skip_cleanup_toe_files ]]; then
  echo "Skipping cleanup of TOE files"
else
  echo "Cleaning TOE files"
  if [[ $( sudo find {{workingDirectory}}/TOE_* -type f | sudo wc -l) -gt 0 ]]; then
      echo "Deleting files within {{workingDirectory}}/TOE_*"
      sudo find {{workingDirectory}}/TOE_* -type f -exec rm -f {} \;
  fi
  if [[ $( sudo find {{workingDirectory}}/TOE_* -type f | sudo wc -l) -gt 0 ]]; then
      echo "Failed to delete {{workingDirectory}}/TOE_*"
      exit 1
  fi
  if [[ $( sudo find {{workingDirectory}}/TOE_* -type d | sudo wc -l) -gt 0 ]]; then
      echo "Deleting {{workingDirectory}}/TOE_*"
      sudo rm -rf {{workingDirectory}}/TOE_*
  fi
  if [[ $( sudo find {{workingDirectory}}/TOE_* -type d | sudo wc -l) -gt 0 ]]; then
      echo "Failed to delete {{workingDirectory}}/TOE_*"
      exit 1
  fi
fi

# Clean up for ssm log files
if [[ -f {{workingDirectory}}/skip_cleanup_ssm_log_files ]]; then
  echo "Skipping cleanup of ssm log files"
else
```

```
  echo "Cleaning up ssm log files"
  if [[ $( sudo find /var/log/amazon/ssm -type f | sudo wc -l) -gt 0 ]]; then
      echo "Deleting files within /var/log/amazon/ssm/*"
      sudo find /var/log/amazon/ssm -type f -exec rm -f {} \;
  fi
  if [[ $( sudo find /var/log/amazon/ssm -type f | sudo wc -l) -gt 0 ]]; then
      echo "Failed to delete /var/log/amazon/ssm"
      exit 1
  fi
  if [[ -d "/var/log/amazon/ssm" ]]; then
      echo "Deleting /var/log/amazon/ssm/*"
      sudo rm -rf /var/log/amazon/ssm
  fi
  if [[ -d "/var/log/amazon/ssm" ]]; then
      echo "Failed to delete /var/log/amazon/ssm"
      exit 1
  fi
fi


if [[ $( sudo find /var/log/sa/sa* -type f | sudo wc -l ) -gt 0 ]]; then
  echo "Deleting /var/log/sa/sa*"
  sudo rm -f /var/log/sa/sa*
fi
if [[ $( sudo find /var/log/sa/sa* -type f | sudo wc -l ) -gt 0 ]]; then
  echo "Failed to delete /var/log/sa/sa*"
  exit 1
fi

if [[ $( sudo find /var/lib/dhclient/dhclient*.lease -type f | sudo wc -l ) -gt
 0 ]]; then
      echo "Deleting /var/lib/dhclient/dhclient*.lease"
      sudo rm -f /var/lib/dhclient/dhclient*.lease
fi
if [[ $( sudo find /var/lib/dhclient/dhclient*.lease -type f | sudo wc -l ) -gt
 0 ]]; then
      echo "Failed to delete /var/lib/dhclient/dhclient*.lease"
      exit 1
fi

if [[ $( sudo find /var/tmp -type f | sudo wc -l) -gt 0 ]]; then
      echo "Deleting files within /var/tmp/*"
      sudo find /var/tmp -type f -exec rm -f {} \;
fi
```

```
if [[ $( sudo find /var/tmp -type f | sudo wc -l) -gt 0 ]]; then
      echo "Failed to delete /var/tmp"
      exit 1
fi
if [[ $( sudo ls /var/tmp | sudo wc -l ) -gt 0 ]]; then
      echo "Deleting /var/tmp/*"
      sudo rm -rf /var/tmp/*
fi

# Shredding is not guaranteed to work well on rolling logs

if [[ -f "/var/lib/rsyslog/imjournal.state" ]]; then
      echo "Deleting /var/lib/rsyslog/imjournal.state"
      sudo rm -f /var/lib/rsyslog/imjournal.state
      sudo rm -f /var/lib/rsyslog/imjournal.state
fi

if [[ $( sudo ls /var/log/journal/ | sudo wc -l ) -gt 0 ]]; then
      echo "Deleting /var/log/journal/*"
      sudo find /var/log/journal/ -type f -exec rm -f {} \;
      sudo rm -rf /var/log/journal/*
fi

sudo touch /etc/machine-id
```

Sostituisci lo script di pulizia di Linux

Image Builder crea immagini che sono sicure per impostazione predefinita e seguono le nostre best practice di sicurezza. Tuttavia, alcuni casi d'uso più avanzati potrebbero richiedere di saltare una o più sezioni dello script di pulizia integrato. Se hai bisogno di saltare alcune operazioni di pulizia, ti consigliamo vivamente di testare l'AMI di output per garantire la sicurezza dell'immagine.

> **Important**
>
> Se si saltano le sezioni dello script di pulizia, nell'immagine finale possono essere incluse informazioni riservate, come i dettagli dell'account del proprietario o le chiavi SSH, e in ogni caso avviate da quell'immagine. Potresti inoltre riscontrare problemi con l'avvio in zone di disponibilità, regioni o account diversi.

La tabella seguente descrive le sezioni dello script di pulizia, i file che vengono eliminati in quella sezione e i nomi di file che è possibile utilizzare per contrassegnare una sezione che Image Builder deve ignorare. Per saltare una sezione specifica dello script di pulizia, è possibile utilizzare il modulo di azione del CreateFile componente o un comando nei dati utente (se sovrascritto) per creare un file vuoto con il nome specificato nella colonna Ignora il nome del file della sezione.

> **Note**
>
> I file creati per ignorare una sezione dello script di pulizia non devono includere un'estensione di file. Ad esempio, se si desidera saltare la `CLOUD_INIT_FILES` sezione dello script, ma si crea un file denominato`skip_cleanup_cloudinit_files.txt`, Image Builder non riconoscerà il file skip.

Input

Sezione di pulizia	File rimossi	Ignora il nome del file della sezione

`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`
		`skip_cleanup_instance_files`

Sezione di pulizia	File rimossi	Ignora il nome del file della sezione
	/var/lib/postfix/master.lock /var/spool/postfix/pid/master.pid /var/.updated /var/cache/yum/x86_64/2/.gpgkeyschecked.yum	
SSH_FILES	/etc/ssh/ssh_host_rsa_key /etc/ssh/ssh_host_rsa_key.pub /etc/ssh/ssh_host_ecdsa_key /etc/ssh/ssh_host_ecdsa_key.pub /etc/ssh/ssh_host_ed25519_key /etc/ssh/ssh_host_ed25519_key.pub /root/.ssh/authorized_keys /home/<all users>/.ssh/authorized_keys;	skip_cleanup_ssh_files

Sezione di pulizia	File rimossi	Ignora il nome del file della sezione
INSTANCE_LOG_FILES	/var/log/audit/audit.log /var/log/boot.log /var/log/dmesg /var/log/cron	skip_cleanup_instance_log_files
TOE_FILES	{{workingDirectory}}/TOE_*	skip_cleanup_toe_files
SSM_LOG_FILES	/var/log/amazon/ssm/*	skip_cleanup_ssm_log_files

Risolvere i problemi relativi a Image Builder

EC2 Image Builder si integra con il monitoraggio e la risoluzione dei problemi Servizi AWS per aiutarti a risolvere i problemi di creazione delle immagini. Image Builder tiene traccia e visualizza l'avanzamento di ogni fase del processo di creazione dell'immagine. Inoltre, Image Builder può esportare i log in una posizione Amazon S3 fornita da te.

Per una risoluzione avanzata dei problemi, puoi eseguire comandi e script predefiniti utilizzando Run Command.AWS Systems Manager

Indice

- Risolvi i problemi relativi alle build della pipeline
- Scenari per la risoluzione dei problemi

Risolvi i problemi relativi alle build della pipeline

Se la compilazione di una pipeline di Image Builder fallisce, Image Builder restituisce un messaggio di errore che descrive l'errore. Image Builder restituisce anche un messaggio `workflow execution ID` nel messaggio di errore, come quello nell'output di esempio seguente:

```
Workflow Execution ID: wf-12345abc-6789-0123-abc4-567890123abc failed with reason: …
```

Image Builder organizza e dirige le azioni di creazione delle immagini attraverso una serie di passaggi definiti per le fasi di runtime del processo standard di creazione dell'immagine. Le fasi di compilazione e test del processo hanno ciascuna un flusso di lavoro associato. Quando Image Builder esegue un flusso di lavoro per creare o testare una nuova immagine, genera una risorsa di metadati del flusso di lavoro che tiene traccia dei dettagli di runtime.

Le immagini dei contenitori hanno un flusso di lavoro aggiuntivo che viene eseguito durante la distribuzione.

Scopri i dettagli relativi agli errori delle istanze di runtime per il tuo flusso di lavoro

Per risolvere un errore di runtime del tuo flusso di lavoro, puoi chiamare le azioni GetWorkflowExecutione ListWorkflowStepExecutionsAPI con il tuo. `workflow execution ID`

Rivedi i registri di runtime del flusso di lavoro

- CloudWatch Registri Amazon

  Image Builder pubblica registri dettagliati di esecuzione del flusso di lavoro nel seguente gruppo e flusso di Image Builder Logs: CloudWatch

  LogGroup:

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

  LogStream (x.x.x/x):

  ```
  ImageVersion/ImageBuildVersion
  ```

  Con CloudWatch Logs, puoi cercare i dati di registro con modelli di filtro. Per ulteriori informazioni, consulta Ricerca nei dati di log utilizzando modelli di filtro nella Amazon CloudWatch Logs User Guide.

- AWS CloudTrail

  Tutte le attività di compilazione vengono registrate anche CloudTrail se sono attivate nel tuo account. Puoi filtrare CloudTrail gli eventi in base alla fonte`imagebuilder.amazonaws.com`. In alternativa, puoi cercare l'ID dell' EC2 istanza Amazon restituito nel log di esecuzione per visualizzare maggiori dettagli sull'esecuzione della pipeline.

- Servizio Amazon Simple Storage (S3)

  Se hai specificato un nome di bucket S3 e un prefisso chiave nella configurazione dell'infrastruttura, il percorso del log di runtime della fase del flusso di lavoro segue questo schema:

  ```
  S3://S3BucketName/KeyPrefix/ImageName/ImageVersion/ImageBuildVersion/
  WorkflowExecutionId/StepName
  ```

  I log che invii al tuo bucket S3 mostrano i passaggi e i messaggi di errore relativi all'attività sull' EC2 istanza durante il processo di creazione dell'immagine. I log includono gli output dei log provenienti dal gestore dei componenti, le definizioni dei componenti che sono stati eseguiti e l'output dettagliato (in JSON) di tutti i passaggi eseguiti sull'istanza. Se riscontri un problema, dovresti esaminare questi file, iniziando con`application.log`, per diagnosticare la causa del problema sull'istanza.

Per impostazione predefinita, Image Builder chiude l'istanza di EC2 build o test di Amazon in esecuzione in caso di errore della pipeline. Puoi modificare le impostazioni dell'istanza per la risorsa di configurazione dell'infrastruttura utilizzata dalla pipeline, per conservare l'istanza di build o test per la risoluzione dei problemi.

Per modificare le impostazioni dell'istanza nella console, è necessario deselezionare la casella di controllo Termina istanza in caso di errore situata nella sezione Impostazioni di risoluzione dei problemi della risorsa di configurazione dell'infrastruttura.

È inoltre possibile modificare le impostazioni dell'istanza con il update-infrastructure-configuration comando in. AWS CLI Imposta il `terminateInstanceOnFailure` valore `false` nel file JSON a cui il comando fa riferimento con il `--cli-input-json` parametro. Per informazioni dettagliate, consultare Aggiornare una configurazione dell'infrastruttura.

Scenari per la risoluzione dei problemi

Questa sezione elenca i seguenti scenari di risoluzione dei problemi dettagliati:

- Accesso negato: codice di stato 403
- Timeout di compilazione durante la verifica della disponibilità di Systems Manager Agent sull'istanza di compilazione
- Il disco secondario di Windows è offline all'avvio
- La compilazione fallisce con l'immagine di base protetta CIS
- AssertInventoryCollection fallisce (Systems Manager Automation)

Per visualizzare i dettagli di uno scenario, scegli il titolo dello scenario per espanderlo. Puoi espandere più titoli contemporaneamente.

Accesso negato: codice di stato 403

Descrizione

La compilazione della pipeline fallisce e restituisce il codice di stato "AccessDenied: Access Denied: 403".

Causa

Tra le cause possibili sono incluse:

- Il profilo dell'istanza non dispone delle autorizzazioni necessarie per l'accesso APIs o le risorse dei componenti.
- Al ruolo del profilo di istanza mancano le autorizzazioni necessarie per la registrazione su Amazon S3. Più comunemente, ciò si verifica quando il ruolo del profilo dell'istanza non dispone delle PutObjectautorizzazioni per i bucket S3.

Soluzione

A seconda della causa, questo problema può essere risolto nel modo seguente:

- Nel profilo dell'istanza mancano le policy gestite: aggiungi le politiche mancanti al ruolo del profilo dell'istanza. Quindi esegui nuovamente la pipeline.
- Nel profilo dell'istanza mancano le autorizzazioni di scrittura per il bucket S3: aggiungi una policy al ruolo del profilo dell'istanza che conceda PutObjectle autorizzazioni di scrittura nel bucket S3. Quindi esegui nuovamente la pipeline.

Timeout di compilazione durante la verifica della disponibilità di Systems Manager Agent sull'istanza di compilazione

Descrizione

La compilazione della pipeline fallisce con «status = 'TimedOut'» e «failure message = 'Step è scaduto mentre la fase sta verificando la disponibilità del Systems Manager Agent sulle istanze di destinazione'».

Causa

Tra le cause possibili sono incluse:

- L'istanza lanciata per eseguire le operazioni di compilazione ed eseguire i componenti non è stata in grado di accedere all'endpoint Systems Manager.
- Il profilo dell'istanza non dispone delle autorizzazioni richieste.

Soluzione

A seconda della possibile causa, questo problema può essere risolto nel modo seguente:

- Problema di accesso, sottorete privata: se state creando in una sottorete privata, assicuratevi di aver configurato gli PrivateLink endpoint per Systems Manager, Image Builder e, se volete registrare, Amazon S3/. CloudWatch Per ulteriori informazioni sulla configurazione degli endpoint, consulta PrivateLink Access services through. AWSAWS PrivateLink
- Autorizzazioni mancanti: aggiungi le seguenti politiche gestite al tuo ruolo collegato ai servizi IAM per Image Builder:
  - EC2InstanceProfileForImageBuilder
  - EC2InstanceProfileForImageBuilderECRContainerCostruisce
  - Amazon SSMManaged InstanceCore

  Per ulteriori informazioni sul ruolo collegato al servizio Image Builder, vedere. Usa i ruoli collegati ai servizi IAM per Image Builder

Il disco secondario di Windows è offline all'avvio

Descrizione

Quando il tipo di istanza utilizzato per creare un AMI Windows Image Builder non corrisponde al tipo di istanza utilizzato per l'avvio dall'AMI, può verificarsi un problema per cui i volumi non root sono offline all'avvio. Ciò si verifica principalmente quando l'istanza di compilazione utilizza un'architettura più recente rispetto all'istanza di avvio.

L'esempio seguente mostra cosa succede quando un'AMI Image Builder viene creata EC2 su un tipo di istanza Nitro e lanciata EC2 su un'istanza Xen:

Tipo di istanza di compilazione: m5.large (Nitro)

Tipo di istanza di avvio: t2.medium (Xen)

```
PS C:\Users\Administrator> get-disk
Number  Friendly Name  Serial Number         Health Status  Operational Status  Total
 Size  Partition Style
------  -------------  -------------         -------------  ------------------
 ----------  ---------------
0       AWS PVDISK     vol0abc12d34e567f8a9  Healthy        Online                  30
 GB  MBR
1       AWS PVDISK     vol1bcd23e45f678a9b0  Healthy        Offline                  8
 GB  MBR
```

Causa

A causa delle impostazioni predefinite di Windows, i dischi appena scoperti non vengono automaticamente portati online e formattati. Quando il tipo di istanza viene modificato EC2, Windows lo considera come se venissero scoperti nuovi dischi. Ciò è dovuto alla modifica del driver sottostante.

Soluzione

Ti consigliamo di utilizzare lo stesso sistema di tipi di istanze quando crei l'AMI Windows da cui intendi eseguire l'avvio. Non includere tipi di istanze basati su sistemi diversi nella configurazione dell'infrastruttura. Se uno dei tipi di istanza specificati utilizza il sistema Nitro, tutti dovrebbero utilizzare il sistema Nitro.

Per ulteriori informazioni sulle istanze basate sul sistema Nitro, consulta Instances built on the Nitro System nella Amazon User Guide. EC2

La compilazione fallisce con l'immagine di base protetta CIS

Descrizione

Stai utilizzando un'immagine di base protetta CIS e la compilazione ha esito negativo.

Causa

Quando la `/tmp` directory è classificata come`noexec`, può causare il fallimento di Image Builder.

Soluzione

Scegliete una posizione diversa per la cartella di lavoro nel `workingDirectory` campo della ricetta dell'immagine. Per ulteriori informazioni, consulta la descrizione del tipo di ImageRecipedati.

AssertInventoryCollection fallisce (Systems Manager Automation)

Descrizione

Systems Manager Automation mostra un errore nella fase di `AssertInventoryCollection` automazione.

Causa

Tu o la tua organizzazione potreste aver creato un'associazione Systems Manager State Manager che raccoglie informazioni di inventario per le EC2 istanze. Se la raccolta avanzata di metadati delle immagini è abilitata per la pipeline di Image Builder (questa è l'impostazione predefinita), Image

Builder tenta di creare una nuova associazione di inventario per l'istanza di build. Tuttavia, Systems Manager non consente più associazioni di inventario per le istanze gestite e impedisce una nuova associazione se ne esiste già una. Ciò causa il fallimento dell'operazione e la creazione della pipeline non riuscita.

Soluzione

Per risolvere questo problema, disattivate la raccolta avanzata di metadati delle immagini utilizzando uno dei seguenti metodi:

- Aggiorna la pipeline di immagini nella console per deselezionare la casella di controllo Abilita raccolta avanzata di metadati. Salva le modifiche ed esegui una build della pipeline.

  Per ulteriori informazioni sull'aggiornamento della pipeline di immagini AMI utilizzando la console EC2 Image Builder, vedere. Aggiorna le pipeline di immagini AMI dalla console Per ulteriori informazioni sull'aggiornamento della pipeline di immagini del contenitore utilizzando la console EC2 Image Builder, vedere. Aggiorna una pipeline di immagini del contenitore dalla console
- È inoltre possibile aggiornare la pipeline di immagini con il update-image-pipeline comando contenuto in. AWS CLI Per fare ciò, includi la `EnhancedImageMetadataEnabled` proprietà nel tuo file JSON, impostata su. `false` L'esempio seguente mostra la proprietà impostata su. `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"
}
```

Per evitare che ciò accada per le nuove pipeline, deselezionate la casella di controllo Abilita raccolta avanzata di metadati quando create una nuova pipeline utilizzando la console EC2 Image Builder oppure impostate il valore della `EnhancedImageMetadataEnabled` proprietà nel file JSON su `false` quando create la pipeline utilizzando il. AWS CLI

Cronologia della documentazione delle modifiche alla guida per l'utente di Image Builder

La tabella seguente descrive le modifiche importanti alla documentazione per data. Per ricevere notifiche sugli aggiornamenti di questa documentazione, è possibile sottoscrivere un feed RSS.

- Versione API: 2025-04-30

Modifica	Descrizione	Data
Aggiornamenti STIG Q1 2025	Versioni aggiornate di Windows STIG e versioni dei componenti STIGS e SCAP applicate per la versione del primo trimestre 2025.	5 maggio 2025
Versione di funzionalità: Support SSM Parameters	Image Builder ora supporta l'uso di AWS Systems Manager(SSM) Parameter Store Parameters nelle ricette e durante la distribuzione delle immagini.	30 aprile 2025
Aggiornamenti STIG Q4	Versioni aggiornate di Windows STIG e applicazione di STIGS per la versione del quarto trimestre 2024.	4 febbraio 2025
Aggiornamento della politica IAM: policy sul ruolo del servizio	È stata aggiornata la politica relativa ai ruoli collegati ai servizi per supportare la creazione di immagini da file ISO del sistema operativo client Microsoft ufficiali importati. Per ulteriori informazioni, consulta	30 dicembre 2024

	la AWSServiceRoleForImageBuilderpolitica.	
Aggiornamento della policy IAM: policy del profilo dell'istanza	È stata aggiornata la politica del ruolo del profilo dell'istanza per supportare la creazione di immagini dai file di immagine del disco. Per ulteriori informazioni, consulta la EC2InstanceProfileForImageBuilderpolicy.	30 dicembre 2024
Versione di funzionalità: da ISO a AMI	Image Builder ora può creare un'immagine da file disco ISO ufficiali per i sistemi operativi client Microsoft Windows 11 e versioni successive.	30 dicembre 2024
Aggiornamenti STIG Q4	Versioni Linux STIG aggiornate e STIGS applicato per la versione del quarto trimestre 2024. Sono state aggiunte informazioni su due nuovi parametri di input per i componenti Linux.	10 dicembre 2024
Aggiornamento della policy IAM: policy del profilo dell'istanza	È stata aggiornata la politica del profilo dell'istanza per concedere l'accesso ai Marketplace AWS componenti Get. Per ulteriori informazioni, consulta la EC2InstanceProfileForImageBuilderpolicy.	2 dicembre 2024
Versione di funzionalità: componenti Marketplace AWS	È stato aggiunto il supporto per i componenti Marketplace AWS software.	2 dicembre 2024

Versione di funzionalità: supporto macOS	È stato aggiunto il supporto per le immagini macOS.	22 ottobre 2024
Supporto documentale per operatori logici in AWS Task Orchestrator and Executor	Utilizzate gli operatori logici per aggiungere o modificare espressioni condizionali nel documento componente.	16 agosto 2024
Supporto documentale per costrutti condizionali in AWS Task Orchestrator and Executor	Utilizzate costrutti condizionali come l'istruzione «if» per indirizzare il flusso di controllo delle azioni dei componenti nel documento del componente.	16 agosto 2024
Supporto documentale per gli operatori di confronto in AWS Task Orchestrator and Executor	Utilizzate gli operatori di confronto per confrontare i valori nel documento del componente.	16 agosto 2024
Aggiunto il modulo di azione Assert	È stata aggiunta la documentazione per il modulo di `ExecuteDocument` azione in`General execution` .	14 agosto 2024
Supporto del sistema operativo	È stato aggiunto il supporto per RHEL 9 e Ubuntu 24.04 LTS.	2 agosto 2024
Aggiornamento del documento: riorganizza il contenuto	Documentazione riorganizzata per migliorare la presentazione e la navigazione.	21 giugno 2024

Aggiornamenti STIG Q2	Versioni Linux STIG aggiornate e STIGS applicato per la versione del secondo trimestre 2024. È stato aggiunto il supporto per RHEL 9, CentOS Stream 9 e Ubuntu 22.04. Non sono state apportate modifiche alle versioni di Windows.	10 maggio 2024
Aggiornamenti STIG Q1	Versioni Linux STIG aggiornate e STIGS applicato per la versione del primo trimestre 2024. Non sono state apportate modifiche alle versioni di Windows.	23 febbraio 2024
Aggiornamenti STIG Q1	Versioni Linux STIG aggiornate e STIGS applicato per la versione del primo trimestre 2024. Non sono state apportate modifiche alle versioni di Windows.	23 febbraio 2024
Versione della funzionalità: gestione del flusso di lavoro delle immagini	Con i flussi di lavoro basati sulle immagini, hai più flessibilità, visibilità e controllo sul processo di creazione delle immagini. È possibile personalizzare le fasi di compilazione e test per i flussi di lavoro oppure utilizzare il flusso di lavoro predefinito di Image Builder.	12 dicembre 2023

Aggiornamenti STIG Q4	Versioni Linux STIG aggiornate e STIGS applicato per la versione del quarto trimestre 2023. Non sono state apportate modifiche alle versioni di Windows. Inoltre, Linux e Windows SCAP sono stati aggiornati con nuovi componenti, software e numeri di benchmark.	7 dicembre 2023
Versione della funzionalità: gestione del ciclo di vita delle immagini	Con le politiche e le regole di gestione del ciclo di vita delle immagini, è possibile definire la strategia di gestione delle risorse per garantire che le immagini obsolete e le risorse associate vengano sottoposte a un processo di etichettatura e rimozione.	17 novembre 2023
Aggiornamento della politica IAM: ruolo di servizio	È stata aggiornata la politica dei ruoli collegati ai servizi, ad esempio il supporto al posizionamento. Per ulteriori informazioni, consulta la AWSServiceRoleForImageBuilderpolitica.	19 ottobre 2023

Aggiornamenti STIG Q3	Versioni STIG aggiornate e STIGS applicato per la versione del terzo trimestre 2023. Inoltre, è stata aggiornat a la messaggistica per chiarire che i pacchetti di terze parti non vengono installati automaticamente, con pochissime eccezioni. Tutti i dati ignorati vengono registrati STIGs .	5 ottobre 2023
Nuove versioni STIG	Versioni STIG aggiornate e STIGS applicato per il rilascio del secondo trimestre 2023.	3 maggio 2023
Nuove versioni STIG	Versioni STIG aggiornate e STIGS applicato per il rilascio del primo trimestre 2023. È stato aggiunto il supporto per 023. AL2	14 aprile 2023
Aggiorna le regioni supportate per AWSTOE	È stato aggiunto il AWSTOE supporto per quanto segue Regioni AWS: Asia Pacifico (Hyderabad), Asia Pacifico (Giacarta), Europa (Zurigo), Europa (Spagna) e Medio Oriente (Emirati Arabi Uniti).	13 aprile 2023

AWSTOE aggiornamenti per il download delle applicazioni	È stata aggiornata la firma per il download AWSTOE dell'installazione su Windows. Inoltre, TLS aggiornato: nota che i download di applicazioni dai bucket S3 ora richiedono la versione TLS 1.2 o successiva.	31 marzo 2023
Versione della funzionalità: flussi di lavoro di compilazione migliorati	Sono stati aggiunti dettagli di runtime per le build di immagini nella nuova scheda del flusso di lavoro nei dettagli della versione di creazione dell'immagine. Informazioni migliorate per la risoluzione dei problemi delle build.	30 marzo 2023

Versione della funzionalità: rilevamento e segnalazione CVE	Per gli account che hanno attivato le scansioni di Amazon Inspector, Image Builder può acquisire i risultati di vulnerabilità ed esposizioni comuni (CVE) di Amazon Inspector durante la fase di test del processo di creazione di nuove immagini, incluse le immagini dei container archiviate in Amazon ECR. Image Builder crea un'istant anea dei risultati per supportar e l'analisi dei dettagli. Image Builder riporta anche il numero di risultati che possono essere filtrati per account, pipeline o immagine, con la possibilità di approfondire i dettagli.	30 marzo 2023
Cronologia delle versioni aggiunta	È stata aggiunta la cronologi a delle versioni alle sezioni Windows e Linux.	17 febbraio 2023
Nuove versioni STIG	Versioni STIG aggiornate e STIGS applicato per la versione del quarto trimestre 2022.	1 febbraio 2023

Versione di funzionalità: Marketplace AWS integrazione e rafforzamento del CIS	Aggiunta Marketplace AWS integrazione per trovare e utilizzare facilmente un'immagine sottoscritta come base per una nuova immagine personalizzata, tra cui immagini CIS Hardened Images e un nuovo componente CIS Hardening del Center for Internet Security.	13 gennaio 2023
Componenti di protezione CIS	Aggiunti componenti di protezione CIS di proprietà e gestiti dal CIS.	13 gennaio 2023
Nuove versioni STIG	Introdotto il supporto per Ubuntu, versioni STIG aggiornate e STIGS applicato per la versione del secondo trimestre 2022.	20 luglio 2022
Aggiornamento del documento : pagina del documento del componente Navigation for Create YAML	Ha spostato il contenuto del documento del componente Create YAML nella relativa pagina e ha aggiornato altre pagine per farvi riferimento.	7 luglio 2022
Nuove versioni STIG	Versioni STIG aggiornate e STIGS applicato per la versione del primo trimestre 2022.	25 aprile 2022
Modulo d'azione aggiunto ExecuteDocument	È stata aggiunta la documentazione per il modulo `ExecuteDocument` d'azione in`General execution` .	28 marzo 2022

Versione di funzionalità: Supporto per un avvio più rapido dell'AMI Windows	Sono state aggiunte impostazioni di configurazione della distribuzione per supportare un avvio più rapido per Windows. AMIs	21 febbraio 2022
Versione di manutenzione: aggiornamento dell'impronta digitale AWSTOE binaria	Impronta binaria aggiornata per il certificato del firmatario. AWSTOE	18 febbraio 2022
Versione della funzionalità: configura l'input per AWSTOE	È stato aggiunto il supporto per l'utilizzo di un file di configurazione JSON come input per il AWSTOE run comando.	3 febbraio 2022
Nuove versioni STIG	Versioni STIG aggiornate e STIGS applicato per il rilascio del quarto trimestre 2021. È stata inoltre aggiunta una sezione per i nuovi componenti SCAP Compliance Checker (SCC).	22 dicembre 2021
Versione di funzionalità: integrazione VM (Import/Export VMIE)	È stato aggiunto il supporto per l'importazione di VM tramite tutti i canali (console, . API/CLI, etc.), and for VM export via API/CLI L'esportazione di macchine virtuali non è attualmente disponibile dalla console Image Builder.	20 dicembre 2021

Versione di funzionalità: condivisione AMI per AWS Organizations e OUs	Configurazione di distribuzione aggiornata per aggiungere il supporto per la condivisione dell'output AMIs con AWS Organizations e OUs.	24 novembre 2021
Aggiornamento del documento: aggiorna le fasi e le fasi dei componenti	Contenuto esteso per gli stadi dei componenti in Image Builder e il modo in cui questi interagiscono con le fasi AWSTOE dei componenti.	22 settembre 2021
Aggiornamento del documento: aggiungi contenuti di CloudTrail integrazione	È stato aggiunto un riepilogo del monitoraggio e un contenuto di CloudTrail integrazione.	17 settembre 2021
Nuove versioni STIG	Versioni STIG aggiornate e STIGS applicato per il rilascio del terzo trimestre 2021.	10 settembre 2021
Versione di funzionalità: EventBridge integrazione con Amazon	È stato aggiunto EventBridge il supporto che consente di connettere Image Builder con eventi correlati Servizi AWS e di avviare eventi in base alle regole definite in. EventBridge	18 agosto 2021
Aggiornamento del documento: riordina le pagine AWSTOE	AWSTOE Pagine riorganizzate per maggiore chiarezza.	11 agosto 2021

Versione della funzionalità: componenti parametrizzati e configurazione aggiuntiva dell'istanza	È stato aggiunto il supporto per specificare i parametri per personalizzare i componenti per le ricette. Configurazione estesa delle EC2 istanze utilizzate per creare e testare le immagini, inclusa la possibili tà di specificare i comandi da eseguire all'avvio e un maggiore controllo sull'inst allazione e la rimozione dell'agente Systems Manager.	7 luglio 2021
Nuove versioni STIG	Versioni STIG aggiornate e STIGS applicato per il rilascio del secondo trimestre 2021.	30 giugno 2021
Miglioramento: miglioramenti nell'etichettatura	Messaggistica migliorat a relativa al tagging delle risorse.	25 giugno 2021
Versione della funzionalità: Avvia l'integrazione dei modelli	È stato aggiunto il supporto per l'utilizzo dei modelli di EC2 avvio di Amazon per la distribuzione AMI nelle impostazioni di distribuzione.	7 Aprile 2021
Versione della funzionalità: miglioramenti alla costruzione dei container	È stato aggiunto il supporto per la configurazione delle mappature dei dispositivi a blocchi e la specifica dell'util izzo come immagine di AMIs base per le build dei contenito ri.	7 Aprile 2021
Nuove versioni STIG	Versioni STIG aggiornate e STIGS applicati.	5 marzo 2021

Aggiorna le espressioni cron	L'elaborazione cron di Image Builder viene aggiornata per aumentare la granularità delle espressioni cron al minuto e utilizza un motore di pianificazione cron standard. Gli esempi vengono aggiornati con il nuovo formato.	8 febbraio 2021
Versione di funzionalità: supporto per contenitori	È stato aggiunto il supporto per la creazione di immagini di container Docker utilizzando Image Builder, con registrazione e archiviazione delle immagini risultanti su Amazon Elastic Container Registry (Amazon ECR) Elastic Container Registry (Amazon ECR). I contenuti sono stati riorganizzati per riflettere nuove funzionalità e adattarsi alle crescite future.	17 dicembre 2020
Documentazione cron ristrutturata	Questa pagina ora evidenzia ulteriori informazioni su come cron funziona con le build della pipeline di Image Builder e include dettagli sull'ora UTC. I caratteri jolly non consentiti per campi specifici sono stati rimossi. Gli esempi ora includono esempi di espressioni sia per la console che per la CLI.	13 Novembre 2020

Versione 2.0 della console: modifica aggiornata della pipeline	Modifiche ai contenuti durante le fasi iniziali e i tutorial sulla creazione della pipeline, oltre alla pagina di gestione delle pipeline di immagini, per incorporare nuove funzionalità e flussi della console.	13 Novembre 2020
Nuove versioni STIG	Versioni STIG aggiornate e STIGS applicati. Nota: il formato dell'elenco è stato modificato per mostrare STIGs quello applicato di default.	15 ottobre 2020
Support per il looping dei costrutti in AWSTOE	Create costrutti ciclici per definire una sequenza ripetuta di istruzioni nell'applicazione. AWSTOE	29 luglio 2020
Support per lo sviluppo locale di AWSTOE componenti	Sviluppa e testa i componenti dell'immagine localmente con l' AWSTOE applicazione.	28 luglio 2020
Criptato AMIs	EC2 Image Builder aggiunge il supporto per la distribuzione di AMI crittografate.	1 luglio 2020
AutoScaling deprecazione	Obsoleto dell'uso di per avviare le istanze. AutoScaling	15 giugno 2020

Support per la connettività tramite AWS PrivateLink	È possibile stabilire una connessione privata tra il VPC e Image EC2 Builder creando un endpoint VPC di interfaccia. Gli endpoint di interfaccia sono alimentati da AWS PrivateLink, una tecnologia che consente di accedere in modo privato a Image APIs Builder senza un gateway Internet, un dispositi vo NAT, una connessione VPN o AWS una connessio ne Direct Connect. Le istanze nel tuo VPC non necessita no di indirizzi IP pubblici per comunicare con Image Builder. APIs Il traffico tra il tuo VPC e Image Builder non esce dalla rete Amazon.	10 giugno 2020
Nuove versioni STIG	Versioni STIG aggiornate e STIGS applicati.	23 gennaio 2020
Risoluzione dei problemi	Aggiunti scenari generali di risoluzione dei problemi.	22 gennaio 2020
Componenti STIG	È possibile creare immagini conformi a STIG con i componenti STIG. AWSTOE	22 gennaio 2020

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.