Guia do usuário

AWS Systems Manager



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

AWS Systems Manager: Guia do usuário

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

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestigie a Amazon. Todas as outras marcas comerciais que não são propriedade da Amazon pertencem aos respectivos proprietários, os quais podem ou não ser afiliados, estar conectados ou ser patrocinados pela Amazon.

Table of Contents

O que é o AWS Systems Manager? 1
Como o Systems Manager beneficia minhas operações? 2
Quem deve usar o Systems Manager? 2
Quais são os principais recursos do Systems Manager? 3
Compatível com Regiões da AWS 4
Acessar o Systems Manager 4
Histórico de nomes de serviço do Systems Manager 6
Sistemas operacionais e tipos de máquinas compatíveis 6
Sistemas operacionais compatíveis com o Systems Manager 6
Tipos de máquinas compatíveis em ambientes híbridos e multinuvem 14
O que é o console unificado? 14
Configurar nós gerenciados para o AWS Systems Manager 17
Gerenciar instâncias do EC2 com o Systems Manager 17
Configurar permissões de instância obrigatórias para o Systems Manager 18
Melhorar a segurança das instâncias do EC2 usando endpoints da VPC para o Systems Manager 30
Gerenciar nós em ambientes híbridos e multinuvem com o Systems Manager 35
Criar o perfil de serviço do IAM obrigatório para o Systems Manager em ambientes híbridos e multinuvem 37
Criar uma ativação híbrida para registrar nós no Systems Manager 46
Instalar o SSM Agent em nós híbridos do Linux 53
Instalar o SSM Agent em nós híbridos do Windows Server 59
Gerenciar dispositivos de borda com o Systems Manager 64
Criar um perfil de serviço do IAM para seus dispositivos de borda 65
Configure seus dispositivos de borda para AWS IoT Greengrass 72
Atualizar o perfil de troca de token do AWS IoT Greengrass e instalar o SSM Agent em seus dispositivos de borda 72
Criar um administrador delegado do AWS Organizations para o Systems Manager 73
Usar um administrador delegado com o Change Manager 73
Usar um administrador delegado com o Explorer 74
Usar um administrador delegado com o OpsCenter 75
Usar um administrador delegado com o Quick Setup 75
Configuração geral 75
Inscrever-se para uma Conta da AWS 75

Criar um usuário com acesso administrativo ... 76
Configurar o AWS Systems Manager ... 78
Configurar o acesso ao console do Systems Manager ... 78
Política de integração do Systems Manager ... 78
Política do operador do console do AWS Systems Manager ... 84
Política somente leitura do operador do console do AWS Systems Manager ... 88
Configurar o console unificado do Systems Manager para uma organização ... 91
Pré-requisitos ... 91
Recursos do console unificado ... 92
Configurar o console unificado ... 94
Configurar o console unificado do Systems Manager para uma única conta e região ... 96
Recursos do console unificado ... 97
Configurar o console unificado ... 98
Desabilitar o console unificado do Systems Manager ... 100
Executar tarefas de gerenciamento de nós ... 102
Como revisar insights de nós ... 102
Adicionar ou remover widgets ... 105
Reorganizar widgets ... 106
Explorando nós ... 107
Explorar nós usando filtros de console ... 107
Explorar nós usando prompts de texto no Amazon Q ... 113
Visualizar detalhes individuais do nó e realizando ações em um nó ... 119
Baixar ou exportar um relatório de nó gerenciado ... 121
Gerenciar o conteúdo e a aparência de relatórios de nós ... 125
Acesso a nós just-in-time usando o Systems Manager ... 126
Configurar o acesso just-in-time com o Systems Manager ... 128
Iniciar uma sessão de acesso a nós just-in-time ... 178
Gerenciar solicitações de acesso just-in-time ... 179
Migrar para o acesso a nós just-in-time do Session Manager ... 181
Desabilitar o acesso just-in-time com o Systems Manager ... 186
Perguntas frequentes ... 186
Diagnosticar e remediar ... 188
Diagnosticar e corrigir implantações com falha ... 189
Diagnosticar e revisar configurações com desvios ... 190
Diagnosticar e corrigir instâncias não gerenciadas do Amazon EC2 ... 191
Tipos de impacto de remediação de ações do runbook ... 199

Visualizar detalhes do histórico de execução para correções 201
Ajustar as configurações do Systems Manager 202
Configurações da conta 202
Configurações organizacionais 203
Configurações de recursos 203
Configurações de Diagnosticar e corrigir 207
Como trabalhar com buckets do Amazon S3 e com políticas de bucket para o Systems Manager 209
Como usar ferramentas do AWS Systems Manager 219
Ferramentas de nós 219
Ferramentas de gerenciamento de alterações 221
Ferramentas de aplicações 223
Ferramentas de operações do 224
Executar uma tarefa de nó com o Systems Manager 225
Pré-requisitos 225
Inicie uma instância usando uma AMI com o SSM Agent pré-instalado 225
Conectar-se à sua instância gerenciada usando o Systems Manager 227
Limpe sua instância 227
Ferramentas de nós 227
Compliance 228
Distributor 248
Fleet Manager 305
Ativações híbridas 393
Inventory 395
Patch Manager 494
Run Command 829
Session Manager 898
State Manager 1051
Ferramentas de gerenciamento de alterações 1153
Automation 1153
Change Calendar 1636
Change Manager 1656
Documentos 1745
Maintenance Windows 1892
Quick Setup 2028
Ferramentas de aplicações 223

AWS AppConfig ... 2080
Application Manager ... 2081
Parameter Store ... 2128
Ferramentas de operações do ... 2288
Incident Manager ... 2289
Explorer ... 2289
OpsCenter ... 2331
Painéis do CloudWatch ... 2419
Trabalhar com o SSM Agent ... 2420
Conheça os detalhes técnicos sobre o SSM Agent ... 2421
Comportamento da credencial do SSM Agent versão 3.2.x.x ... 2421
Precedência de credenciais do SSM Agent ... 2421
Configurar o SSM Agent para usar com o Federal Information Processing Standard (FIPS, Padrão federal de processamento de informações) ... 2423
Sobre a conta local ssm-user ... 2425
SSM Agent e Instance Metadata Service (IMDS) ... 2426
Manter o SSM Agent atualizado ... 2426
Garantir que o diretório de instalação do SSM Agent não seja modificado, movido ou excluído ... 2427
Atualizações contínuas do SSM Agent nas Regiões da AWS ... 2427
Comunicações do SSM Agent com os buckets do S3 gerenciados pela AWS ... 2428
Encontrar AMIs com o SSM Agent pré-instalado ... 2436
Verifique o status do SSM Agent ... 2437
Trabalhar com o SSM Agent em instâncias do EC2 para Linux ... 2442
Verificar a assinatura do SSM Agent ... 2442
Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux ... 2457
Configurar o SSM Agent para usar um proxy em nós do Linux ... 2517
Trabalhar com o SSM Agent em instâncias do EC2 para macOS ... 2521
Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para macOS ... 2523
Trabalhar com o SSM Agent em instâncias do EC2 para Windows Server ... 2524
Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Windows Server ... 2526
Configurar o SSM Agent para usar um proxy para instâncias do Windows Server ... 2528
Verificar o status do SSM Agent e iniciar o agente ... 2532
Verificar o número de versão do SSM Agent ... 2535
Visualizar logs do SSM Agent ... 2539

Permitir o registro em log de depuração do SSM Agent ... 2540
Restringir o acesso aos comandos em nível raiz por meio do SSM Agent ... 2543
Automatizar atualizações do SSM Agent ... 2544
Atualizar automaticamente o SSM Agent ... 2546
Assinar as notificações do SSM Agent ... 2547
Solução de problemas do SSM Agent ... 2548
O SSM Agent está desatualizado ... 2548
Solucionar problemas usando arquivos de log do SSM Agent ... 2549
Arquivos de log do agente não alternam (Windows) ... 2549
Unable to connect to endpoint ... 2550
Verificar sua configuração de VPC ... 2551
Verificar seus atributos relacionados a DNS da VPC ... 2552
Verificar regras de entrada e saída em grupos de segurança de endpoints ... 2553
Usa a `ssm-cli` para solucionar problemas de disponibilidade do nó gerenciado ... 2554
Segurança ... 2555
Proteção de dados ... 2556
Criptografia de dados ... 2557
Privacidade do tráfego entre redes ... 2560
Gerenciamento de Identidade e Acesso ... 2560
Público ... 2561
Como autenticar com identidades ... 2561
Gerenciar o acesso usando políticas ... 2565
Como o AWS Systems Manager funciona com o IAM ... 2569
Exemplos de políticas baseadas em identidade ... 2580
Políticas gerenciadas pela AWS ... 2592
Solução de problemas ... 2656
Usar perfis vinculados a serviço ... 2658
Perfil de dados do Inventory e Explorer ... 2660
Ffunção de descoberta de contas do OpsCenter e Explorer ... 2662
OpsData e função de criação OpsItems ... 2666
Função de criação de insights operacionais ... 2670
Função de verificação de integridade da implantação do Quick Setup ... 2673
Exportar perfil de serviço do OpsData ... 2676
Perfil de serviço de acesso a nós just-in-time ... 2679
Perfil de serviço de notificações de solicitação de acesso a nós just-in-time ... 2683
Registrar em log e monitoramento ... 2686

Validação de conformidade 2689
Resiliência 2690
Segurança da infraestrutura 2691
Análise de configuração e vulnerabilidade 2691
Práticas recomendadas de segurança 2691
Melhores práticas de segurança preventivas do Systems Manager 2692
Melhores práticas de auditoria e monitoramento do Systems Manager 2696
Exemplos de código 2699
Conceitos básicos 2707
Hello Systems Manager 2710
Conheça os conceitos básicos 2714
Ações 2766
Registrar em log e monitoramento 3108
Ferramentas de monitoramento 3109
Enviar logs de nós para o CloudWatch Logs unificado (agente do CloudWatch) 3109
Migrar a coleta de logs de nós do Windows Server para o agente do CloudWatch 3111
Armazenar as configurações do agente do CloudWatch no Parameter Store 3122
Reverter para a coleta de logs com o SSM Agent 3123
Enviar logs do SSM Agent ao CloudWatch Logs 3127
Monitoramento dos seus eventos de solicitação de alteração 3130
Monitoramento das automações 3133
Métricas do Automation 3133
Monitorar métricas do Run Command com o Amazon CloudWatch 3134
Métricas e dimensões de Run Command do Systems Manager 3135
Registrar em log chamadas de API do AWS Systems Manager com o AWS CloudTrail 3136
Eventos de dados do Systems Manager no CloudTrail 3137
Eventos de gerenciamento do Systems Manager no CloudTrail 3139
Exemplos de eventos do Systems Manager 3139
Registro de saída de ações do Automation em log com o CloudWatch Logs 3145
Configurar o Amazon CloudWatch Logs para Run Command 3149
Especificar o CloudWatch Logs ao enviar comandos 3150
Visualizar a saída de comandos no CloudWatch Logs 3151
Monitorar com o Amazon EventBridge 3151
Configurar o EventBridge para eventos do Systems Manager 3153
Exemplos de eventos do Amazon EventBridge para Systems Manager 3157
Cenários de exemplo: destinos do Systems Manager em regras do Amazon EventBridge . 3173

Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS 3175
Configurar notificações do Amazon SNS para o AWS Systems Manager 3176
Exemplo de notificações do Amazon SNS para AWS Systems Manager 3188
Usar o Run Command para enviar um comando que retorna notificações de status 3189
Usar uma janela de manutenção para enviar um comando que retorna notificações de status 3193
Integrações de produtos e serviços 3199
Integração com Serviços da AWS 3199
Computação 3199
Internet das Coisas (IoT) 3202
Armazenamento 3203
Developer Tools 3204
Segurança, identidade e conformidade 3205
Criptografia e PKI 3208
Gerenciamento e governança 3209
Rede e entrega de conteúdo 3214
Analytics 3215
Integração de aplicativo 3217
AWS Management Console 3218
Executar scripts no Amazon S3 3219
Fazer referência a segredos do AWS Secrets Manager em parâmetros do Parameter Store 3223
Criptografia do AWS KMS para parâmetros SecureString do Parameter Store 3230
Use segredos do AWS Secrets Manager no Amazon Elastic Kubernetes Service 3244
Usar parâmetros do Parameter Store em funções do AWS Lambda 3264
Integração com outros produtos e serviços 3289
Executar scripts do GitHub 3293
Usar os perfis do Chef InSpec com o Systems Manager Compliance 3301
Integração com o ServiceNow 3307
Referência do AWS Systems Manager 3308
Como trabalhar com AWS SDKs 3309
Buckets do Amazon S3 para operações de aplicação de patches 3310
Buckets contendo documentos do SSM Command para operações de aplicação de patches (Linux e Windows Server) 3311

Buckets contendo documentos do SSM Command para operações de aplicação de patches (macOS) ... 3314
Buckets contendo snapshots da lista de referência de patches gerenciadas pela AWS ... 3317
Padrões e tipos de eventos do EventBridge para o Systems Manager ... 3320
Tipo de evento: automação ... 3321
Tipo de evento: Change Calendar ... 3322
Tipo de evento: Change Manager ... 3322
Tipo de evento: conformidade com a configuração ... 3323
Tipo de evento: inventário ... 3323
Tipo de evento: janela de manutenção ... 3324
Tipo de evento: OpsCenter ... 3327
Tipo de evento: Parameter Store ... 3327
Tipo de evento: Run Command ... 3328
Tipo de evento: State Manager ... 3329
Expressões cron e rate ... 3330
Informações gerais sobre as expressões cron e rate ... 3330
Expressões cron e rate para associações ... 3335
Expressões cron e rate para janelas de manutenção ... 3338
ec2messages, ssmmessages e outras operações da API ... 3340
Operações de API relacionadas a agentes (endpoints `ssmmessages` e `ec2messages`) ... 3341
Operações de API relacionadas à instância do namespace `ssm:*` ... 3343
ssm:* namespace outras operações de API ... 3345
Formatos de strings de data e hora para o Systems Manager ... 3345
Formatar strings de data e hora para o Systems Manager ... 3346
Criar strings personalizados de data e hora para o Systems Manager ... 3346
Casos de uso e melhores práticas ... 3349
Excluir recursos e artefatos do Systems Manager ... 3352
Selecionar entre State Manager e Maintenance Windows ... 3357
State Manager e Maintenance Windows: Casos de uso principais ... 3357
Informações relacionadas ... 3366
Histórico do documentos ... 3368
Atualizações antes de junho de 2018 ... 3602
Convenções do documento ... 3623

O que é o AWS Systems Manager?

O AWS Systems Manager ajuda você a visualizar, gerenciar e operar nós centralmente em escala em ambientes da AWS, on-premises e multicloud. Com o lançamento de uma experiência de console unificada, o Systems Manager consolida várias ferramentas para ajudar você a concluir tarefas de nós comuns nas Contas da AWS e Regiões da AWS.

Para usar o Systems Manager, os nós devem ser gerenciados, o que significa que o SSM Agent está instalado na máquina e o agente pode se comunicar com o serviço Systems Manager. Para ajudar você a descobrir por que os nós não estão se identificando como gerenciados, o Systems Manager oferece um runbook de diagnóstico e correção de problemas do agente com um clique que você pode configurar para ser executado automaticamente de acordo com uma programação definida por você. Esse recurso ajuda a identificar por que os nós não conseguem se conectar ao Systems Manager, incluindo configurações incorretas de rede. Esse recurso também fornece runbooks recomendados para corrigir problemas de rede e outros problemas que impedem que os nós sejam configurados como nós gerenciados.

A experiência do console unificado também inclui um painel que fornece uma visão geral de alto nível dos seus nós. Você pode detalhar informações mais específicas sobre nós, como quais nós estão executando software de sistema operacional (SO) desatualizado. Também é possível usar filtros para visualizações granulares com base nos metadados da instância, como sistemas operacionais e versões de sistema operacional, Regiões da AWS, Contas da AWS e versões do SSM Agent. Esses filtros ajudam você a recuperar informações relevantes em um nível específico de conta ou aplicação em toda a organização.

Tópicos

- Como o Systems Manager beneficia minhas operações?
- Quem deve usar o Systems Manager?
- Quais são os principais recursos do Systems Manager?
- Compatível com Regiões da AWS
- Acessar o Systems Manager
- Histórico de nomes de serviço do Systems Manager
- Sistemas operacionais e tipos de máquinas compatíveis
- O que é o console unificado?

Como o Systems Manager beneficia minhas operações?

Os benefícios do Systems Manager incluem:

- Melhorar a visibilidade em toda a sua infraestrutura

  O Systems Manager fornece uma visão centralizada dos nós nas contas e regiões da sua organização. Acesse rapidamente as informações da instância, como ID, nome, detalhes do sistema operacional e agentes instalados. Use o Amazon Q Developer para consultar metadados de instâncias usando linguagem natural, o que ajuda você a identificar problemas e agir com mais rapidez.
- Aumentar a eficiência operacional com automação

  Automatize tarefas operacionais comuns e reduza o tempo e os esforços necessários para manter seus sistemas. O Systems Manager fornece gerenciamento remoto seguro e protegido de seus nós em grande escala sem que seja necessário fazer login em seus servidores. Você não precisa mais usar bastion hosts, SSH ou PowerShell remoto. O Systems Manager também oferece uma forma simples de automatizar tarefas administrativas comuns em grupos de nós como edições de registro, gerenciamento de usuários e instalações de software e patches.
- Simplifique o gerenciamento de nós em escala em qualquer ambiente

  O Systems Manager ajuda você a gerenciar nós em ambientes da AWS, on-premises e multicloud. Agende diagnósticos automatizados para identificar problemas no SSM Agent e corrigi-los com runbooks com apenas um clique. Depois que seus nós forem configurados como nós gerenciados, você poderá executar tarefas operacionais críticas, como aplicar patches de segurança, iniciar sessões registradas e executar comandos remotamente.

Quem deve usar o Systems Manager?

O Systems Manager é usado por gerentes e operadores de TI, engenheiros de DevOps, gerentes de segurança e conformidade, diretores de TI e CIOs. De um modo geral, o Systems Manager é apropriado para:

- Organizações que desejam melhorar o gerenciamento e a segurança de seus nós em grande escala.
- Organizações que desejam aumentar a visibilidade e a agilidade operacional ao gerenciar sua infraestrutura.

- Organizações que desejam aumentar a eficiência operacional em grande escala.

Quais são os principais recursos do Systems Manager?

Os principais recursos do Systems Manager são compartilhados entre o console unificado e as ferramentas individuais fornecidas pelo Systems Manager para ajudar você a gerenciar nós em grande escala.

Console unificado

O console unificado fornece uma experiência centralizada para visualizar e gerenciar seus nós. Esse console utiliza várias ferramentas do Systems Manager e mais para fornecer:

- Visualizações centralizadas de seus nós
- Insights detalhados dos nós
- Diagnóstico e correção automatizados de problemas comuns de nós

Para obter mais informações sobre o console unificado, consulte O que é o console unificado?.

Ferramentas

As ferramentas consistem nos recursos individuais do Systems Manager e em seus recursos, como Run Command, Session Manager, Automation e Parameter Store. Com as ferramentas do Systems Manager, é possível:

- Aplicar patches em nós em grande escala
- Conectar-se com segurança aos nós sem abrir portas de entrada
- Executar comandos remotamente nos nós
- Armazenar com segurança dados referenciados por aplicações
- Automatizar tarefas comuns de administração de sistemas

Para obter mais informações sobre o Systems Manager, consulte Como usar ferramentas do AWS Systems Manager.

Compatível com Regiões da AWS

Para obter uma lista de Regiões da AWS que oferecem suporte às ferramentas do Systems Manager, consulte Endpoints de serviço do Systems Manager no Referência geral da Amazon Web Services.

O console unificado do Systems Manager, lançado em 21 de novembro de 2024, está disponível nas seguintes Regiões da AWS:

- Região Leste dos EUA (N. da Virgínia)
- Região Leste dos EUA (Ohio)
- Região Oeste dos EUA (N. da Califórnia)
- Região Oeste dos EUA (Oregon)
- Região Canadá (Central)
- Região América do Sul (São Paulo)
- Região Ásia-Pacífico (Mumbai)
- Região Ásia-Pacífico (Tóquio)
- Região Ásia-Pacífico (Seul)
- Região Ásia-Pacífico (Singapura)
- Região Ásia-Pacífico (Sydney)
- Região Europa (Frankfurt)
- Região Europa (Estocolmo)
- Região Europa (Irlanda)
- Região Europa (Londres)
- Região Europa (Paris)

Acessar o Systems Manager

Você pode trabalhar com o Systems Manager de qualquer uma das seguintes formas:

Console do Systems Manager:

O console do Systems Manager é uma interface baseada em navegador para acesso e uso do Systems Manager.

Console do AWS IoT Greengrass V2

Você pode visualizar e gerenciar dispositivos de borda configurados para oAWS IoT Greengrass no console do Greengrass.

Ferramentas da linha de comando da AWS

Usnado as ferramentas de linha de comando da AWS, você pode emitir comandos na linha de comando do seu sistema para executar o Systems Manager e outras tarefas da AWS. As ferramentas têm suporte no Linux, macOS e Windows. Usar AWS Command Line Interface (AWS CLI) pode ser mais rápido e mais conveniente do que usar o console. As ferramentas da linha de comando também são úteis se você quiser criar scripts que realizem tarefas da AWS.

A AWS fornece dois conjuntos de ferramentas de linha de comando: AWS Command Line Interface e AWS Tools for Windows PowerShell. Para obter informações sobre a instalação e o uso da AWS CLI, consulte o Guia do usuário da AWS Command Line Interface. Para obter informações sobre a instalação e o uso do Tools for Windows PowerShell, consulte o Manual do usuário do Ferramentas da AWS para PowerShell.

> **Note**
>
> Nas instâncias do Windows Server, o Windows PowerShell 3.0 ou posterior é necessário para executar determinados documentos do SSM (por exemplo, o documento legado `AWS-ApplyPatchBaseline`). Verifique se as instâncias do Windows Server estão executando o Windows Management Framework 3.0 ou posterior. O framework inclui o Windows PowerShell.

SDKs da AWS

A AWS fornece Kits de Desenvolvimento de Software (SDKs) que consistem em bibliotecas e códigos de exemplo para várias linguagens de programação e plataformas (por exemplo Java, Python, Ruby, .NET, iOS e Android e outras). Os SDKs fornecem uma maneira conveniente de conceder acesso programático ao Systems Manager. Para obter informações sobre os SDKs AWS, incluindo como fazer download e instalá-los, consulte Ferramentas da Amazon Web Services.

Histórico de nomes de serviço do Systems Manager

O AWS Systems Manager (Systems Manager) era conhecido anteriormente como "Amazon Simple Systems Manager (SSM)" e "Amazon EC2 Systems Manager (SSM)". O nome abreviado original do serviço, "SSM", ainda é refletido em vários recursos da AWS, incluindo alguns outros consoles de serviço. Alguns exemplos:

- Systems Manager Agent: SSM Agent
- Parâmetros do Systems Manager: parâmetros do SSM
- Endpoints do serviço do Systems Manager: `ssm.region.amazonaws.com`
- Tipos de recursos do AWS CloudFormation: `AWS::SSM::Document`
- Identificador de regra do AWS Config: `EC2_INSTANCE_MANAGED_BY_SSM`
- AWS Command Line Interface Comandos da (AWS CLI): `aws ssm describe-patch-baselines`
- AWS Identity and Access Management Nomes de políticas gerenciadas (IAM): `AmazonSSMReadOnlyAccess`
- ARNs de recursos do Systems Manager: `arn:aws:ssm:region:account-id:patchbaseline/pb-07d8884178EXAMPLE`

Sistemas operacionais e tipos de máquinas compatíveis

Antes de trabalhar com o Systems Manager, verifique se há suporte para o sistema operacional (SO), a versão do sistema operacional e o tipo de máquina como nós gerenciados.

Tópicos

- Sistemas operacionais compatíveis com o Systems Manager
- Tipos de máquinas compatíveis em ambientes híbridos e multinuvem

Sistemas operacionais compatíveis com o Systems Manager

As seções a seguir listam os SOs e as versões de SO compatíveis com o Systems Manager.

> **Note**
>
> Se você planeja gerenciar e configurar os dispositivos principais do AWS IoT Greengrass usando o Systems Manager, esses dispositivos devem atender aos requisitos para o AWS IoT Greengrass. Para obter mais informações, consulte Configurar dispositivos principais do AWS IoT Greengrass no Guia do desenvolvedor do AWS IoT Greengrass Version 2.
> Se você planeja gerenciar e configurar o AWS IoT e não os dispositivos de borda não AWS, esses dispositivos devem atender aos requisitos listados aqui e serem configurados como nós gerenciados on-premises para o Systems Manager. Para obter mais informações, consulte Gerenciar dispositivos de borda com o Systems Manager.

> **Important**
>
> Talvez o Patch Manager, uma ferramenta do Systems Manager, não seja compatível com todas as versões de SO listadas neste tópico. Para obter a lista completa de versões de SO compatíveis com o Patch Manager, consulte Pré-requisitos da Patch Manager.

Tipos de sistema operacional

- Linux
- macOS (somente instâncias do Amazon EC2)
- Raspberry Pi OS (anteriormente Raspbian)
- Windows Server

Selecione uma plataforma de sistema operacional para ver as versões principais e secundárias compatíveis.

Linux

AlmaLinux

Versões	x86	x86_64	ARM64
8.3 a 8.10		✓	✓
9.x		✓	✓

Amazon Linux 1

Versões	x86	x86_64	ARM64
2012.03-2018.03	✓	✓	

> **Note**
>
> A partir da versão 2015.03, o Amazon Linux 1 é lançado em versões x86_64.
> O Amazon Linux 1 chegou ao fim do ciclo de suporte padrão em 31 de dezembro de 2020 e atingiu o fim da vida útil em 31 de dezembro de 2023, conforme anunciado em Atualização sobre o fim da vida útil do Amazon Linux AMI no Blog de notícias da AWS. A AWS não fornece mais Amazon Machine Images (AMIs) para esse sistema operacional. No entanto, a AWS Systems Manager continua fornecendo suporte para instâncias existentes do Amazon Linux 1.

Amazon Linux 2

Versões	x86	x86_64	ARM64
2.0 e todas as versões posteriores		✓	✓

Amazon Linux 2023

Versões	x86	x86_64	ARM64
2023.0.20230315.0 e todas as versões posteriores		✓	✓

Bottlerocket

Versões	x86_64	ARM64
1.0.0 e todas as versões posteriores	✓	✓

CentOS

Versões	x86	x86_64	ARM64
6.x[1]	✓	✓	
7.1 e as versões posteriores a 7.x		✓	✓
8.x		✓	✓

[1] Para usar estas versões, é necessário usar uma versão 3.0.x do SSM Agent. Recomendamos usar a versão 3.0.x mais recente disponível do SSM Agent. Versões posteriores do SSM Agent (3.1 ou posterior) não são compatíveis.

CentOS Stream

Versões	x86	x86_64	ARM64
8		✓	✓
9		✓	✓

Debian Server

Versões	x86	x86_64	ARM64
Jessie (8)		✓	
Stretch (9)		✓	✓
Buster (10)		✓	✓

Versões	x86	x86_64	ARM64
Bullseye (11)		✓	✓
Bookworm (12)		✓	✓

Oracle Linux

Versões	x86	x86_64	ARM64
7.5 – 7.8		✓	
8.x		✓	
9.x		✓	

Red Hat Enterprise Linux (RHEL)

Versões	x86	x86_64	ARM64
6.x[1]	✓	✓	
7.0-7.5		✓	
7.6 – 8.x		✓	✓
9.x		✓	✓
10.x		✓	✓

[1] Para usar estas versões, é necessário usar uma versão 3.0.x do SSM Agent. Recomendamos usar a versão 3.0.x mais recente disponível do SSM Agent. Versões posteriores do SSM Agent (3.1 ou posterior) não são compatíveis.

Rocky Linux

Versões	x86	x86_64	ARM64
8.x		✓	✓

Versões	x86	x86_64	ARM64
9.x		✓	✓

SUSE Linux Enterprise Server (SLES)

Versões	x86	x86_64	ARM64
12 e as versões posteriores a 12.x		✓	
15x		✓	✓

Ubuntu Server

Versões	x86	x86_64	ARM64
12.04 LTS e 14.04 LTS	✓	✓	
16.04 LTS e 18.04 LTS		✓	✓
20.04 LTS e 20.10 STR		✓	✓
22.04 LTS		✓	✓
23.04, 23.10		✓	✓
24.04 LTS		✓	✓
24.10		✓	✓
25.04		✓	✓

macOS (somente instâncias do Amazon EC2)

Versão	x86	x86_64	Mac with Apple silicon
10.14.x (Mojave)		✓	
10.15.x (Catalina)		✓	
11.x (Big Sur)		✓	✓
12.x (Monterey)		✓	✓
13.x (Ventura)		✓	✓
14.x (Sonoma)		✓	✓
15.x (Sequoia)		✓	✓

> **Note**
>
> Não há suporte para macOS em todas as Regiões da AWS. Para obter mais informações sobre o suporte a instâncias do EC2 para macOS, consulte Instâncias Mac do Amazon EC2 no Guia do usuário do Amazon EC2.

Raspberry Pi OS (anteriormente Raspbian)

Versão	ARM32
8 (Jessie)	✓
9 (Stretch)	✓

Mais informações

- Gerenciar dispositivos Raspberry Pi usando o AWS Systems Manager

Windows Server

O SSM Agent requer o Windows PowerShell 3.0 ou posterior para executar determinados documentos do AWS Systems Manager (documentos do SSM) em instâncias do Windows Server (por exemplo, o documento do `AWS-ApplyPatchBaseline` legado). Verifique se as instâncias do Windows Server estão executando o Windows Management Framework 3.0 ou posterior. Esse framework inclui o Windows PowerShell. Para obter mais informações, consulte Windows Management Framework 3.0.

Versão	x86	x86_64	ARM64
2008[1]	✓	✓	
2008 R2[1]		✓	
2012 e 2012 R2[2]		✓	
2016		✓	
2019		✓	
2022		✓	
2025		✓	

[1] Suporte ao Windows Server 2008 e 2008 R2: a partir de 14 de janeiro de 2020, o Windows Server 2008 não é mais compatível para obter recursos ou atualizações de segurança da Microsoft. As Amazon Machine Images (AMIs) herdadas para Windows Server 2008 e 2008 R2 ainda incluem a versão 2 do SSM Agent pré-instalada, mas o Systems Manager não é oficialmente compatível com as versões 2008 e não atualiza mais o agente para essas versões do Windows Server. Além disso, o SSM Agent versão 3 pode não ser compatível com todas as operações no Windows Server 2008 e 2008 R2. A versão final do SSM Agent oficialmente compatível com as versões 2008 do Windows Server é a 2.3.1644.0.

[2] Suporte ao Windows Server 2012 e 2012 R2: o Windows Server 2012 e 2012 R2 atingiram o fim do suporte em 10 de outubro de 2023. Para usar o SSM Agent com essas versões, recomendamos usar as Extended Security Updates (ESUs) da Microsoft. Para obter mais informações, consulte Fim do suporte ao Windows Server 2012 e 2012 R2 no site da Microsoft.

Tipos de máquinas compatíveis em ambientes híbridos e multinuvem

O Systems Manager é compatível com vários tipos de máquinas como nós gerenciados. Um nó gerenciado é qualquer máquina configurada para trabalhar com o Systems Manager.

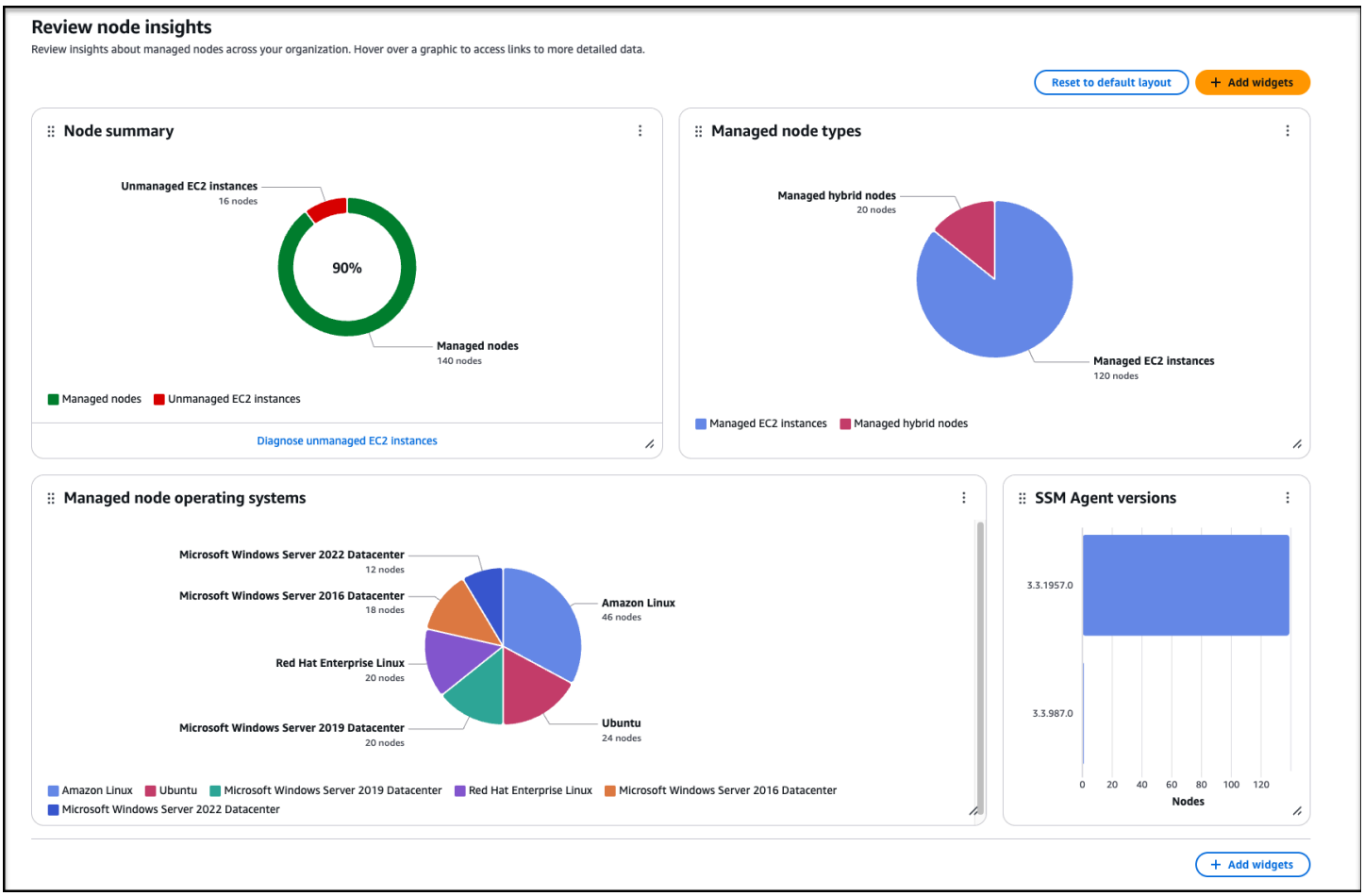
Este guia do usuário usa o termo híbrido e multinuvem para se referir a ambientes que contenham qualquer combinação dos seguintes tipos de máquinas:

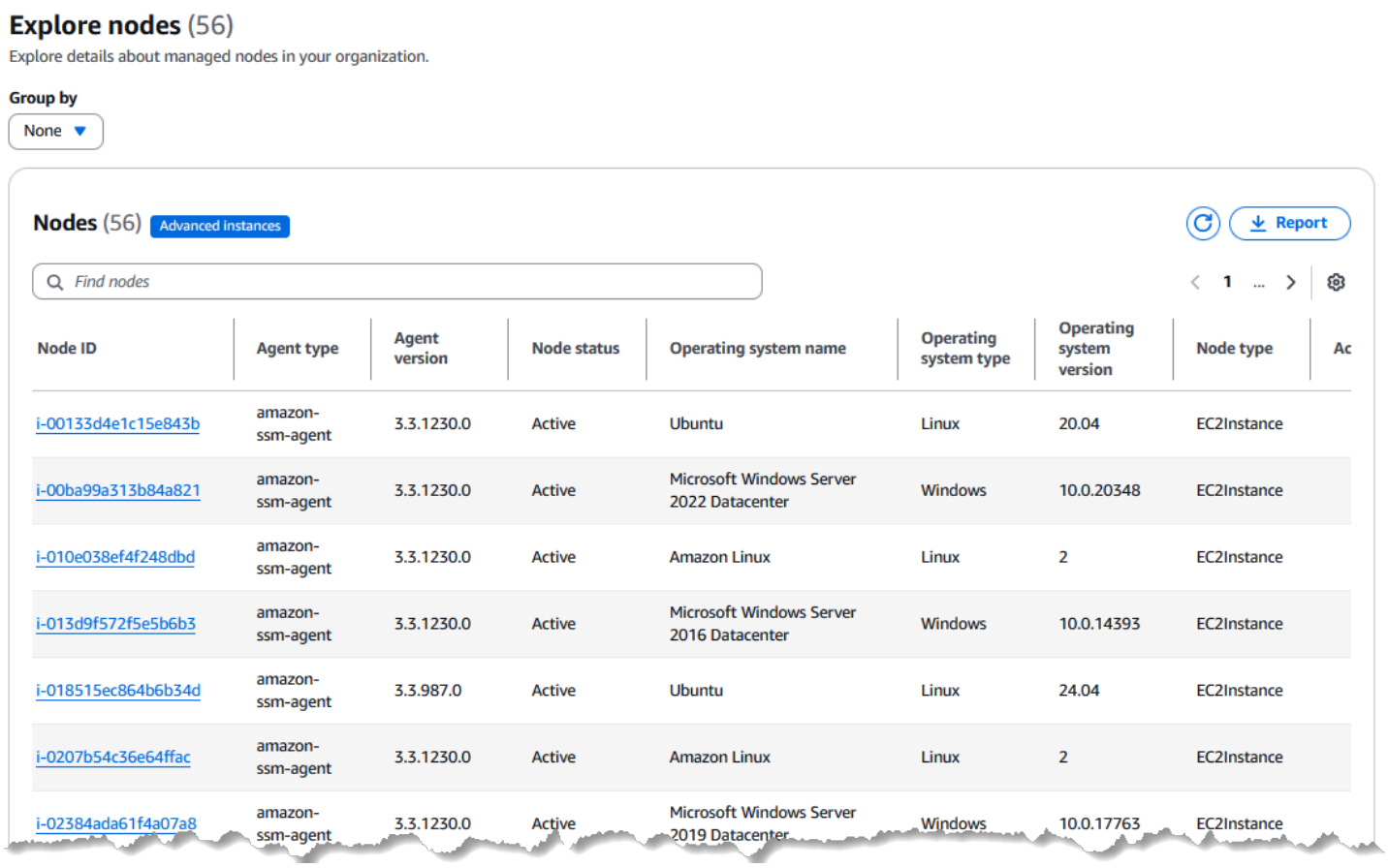
- Instâncias do Amazon Elastic Compute Cloud (Amazon EC2)
- Servidores em suas próprias instalações (servidores on-premises)
- Dispositivos principais do AWS IoT Greengrass
- AWS IoT e dispositivos de borda que não são da AWS
- Máquinas virtuais (VMs), inclusive VMs em outros ambientes de nuvem

Para obter informações sobre suporte da AWS para ambientes híbridos e multinuvem, consulte Soluções da AWS para nuvem híbrida e multinuvem.

O que é o console unificado?

O console unificado do Systems Manager é uma experiência consolidada que combina várias ferramentas para ajudar você a concluir tarefas de nós comuns em várias Contas da AWS e Regiões da AWS em uma organização do AWS Organizations, ou em uma única conta e região. Os nós podem ser instâncias do EC2, servidores híbridos ou servidores executados em um ambiente multinuvem. No console unificado, você recebe informações detalhadas sobre seus nós. É possível gerar relatórios para seus nós, diagnosticar e corrigir problemas comuns que impedem que os nós sejam reportados como gerenciados pelo Systems Manager, por exemplo, problemas de conectividade. Além dos resumos sobre seus nós, é possível visualizar detalhes específicos sobre um nó, como inventário de software e status de aplicação de patches.


Review node insights
Review insights about managed nodes across your organization. Hover over a graphic to access links to more detailed data.
Reset to default layout
Add widgets
Node summary
Unmanaged EC2 instances
16 nodes
90%
Managed nodes
140 nodes
Managed nodes
Unmanaged EC2 instances
Diagnose unmanaged EC2 instances
Managed node types
Managed hybrid nodes
20 nodes
Managed EC2 instances
120 nodes
Managed EC2 instances
Managed hybrid nodes
Managed node operating systems
Microsoft Windows Server 2022 Datacenter
12 nodes
Microsoft Windows Server 2016 Datacenter
18 nodes
Red Hat Enterprise Linux
20 nodes
Microsoft Windows Server 2019 Datacenter
20 nodes
Amazon Linux
46 nodes
Ubuntu
24 nodes
Amazon Linux
Ubuntu
Microsoft Windows Server 2019 Datacenter
Red Hat Enterprise Linux
Microsoft Windows Server 2016 Datacenter
Microsoft Windows Server 2022 Datacenter
SSM Agent versions
3.3.1957.0
3.3.987.0
0
20
40
60
80
100
120
Nodes
Add widgets



Se você tem nós em várias contas e regiões em uma organização ou nós em uma única conta e região, recomendamos usar o console unificado. O Systems Manager continua a adicionar ao console unificado a capacidade de concluir mais tarefas. Para saber mais sobre as tarefas do nó que você pode executar agora usando o console unificado, consulte Executar tarefas de gerenciamento de nós com o AWS Systems Manager.

Avance para os próximos tópicos para configurar seus nós para serem gerenciados pelo Systems Manager. Para saber mais sobre como configurar os nós para o Systems Manager, consulte Configurar nós gerenciados para o AWS Systems Manager. Após configurar seus nós, você poderá configurar o Systems Manager e o console unificado. Para saber mais sobre como configurar o Systems Manager, consulte Configurar o AWS Systems Manager.

Configurar nós gerenciados para o AWS Systems Manager

Conclua as tarefas nesta seção para configurar e configurar perfis, contas de usuário, permissões e recursos iniciais para usar ferramentas do AWS Systems Manager. As tarefas descritas nesta seção são normalmente executadas pela Conta da AWS e pelos administradores de sistemas. Quando essas etapas forem concluídas, os usuários na organização poderão usar o Systems Manager para configurar, gerenciar e acessar os nós gerenciados. Um nó gerenciado é qualquer máquina configurada para uso com o Systems Manager em um ambiente híbrido e multinuvem.

> **Note**
>
> Se você planeja usar as instâncias do Amazon EC2 e seus próprios recursos de computação em um ambiente híbrido e multinuvem, siga primeiro as etapas em Gerenciar instâncias do EC2 com o Systems Manager. Esse tópico apresenta as etapas na melhor ordem para concluir a configuração do Systems Manager para instâncias do EC2 e máquinas que não são do EC2.

Se você já usa outros Serviços da AWS, já concluiu algumas dessas etapas. No entanto, outras etapas são específicas ao Systems Manager. Portanto, recomendamos analisar toda esta seção para garantir que você está pronto para usar todas as ferramentas do Systems Manager.

Tópicos

- Gerenciar instâncias do EC2 com o Systems Manager
- Gerenciar nós em ambientes híbridos e multinuvem com o Systems Manager
- Gerenciar dispositivos de borda com o Systems Manager
- Criar um administrador delegado do AWS Organizations para o Systems Manager
- Configuração geral para AWS Systems Manager

Gerenciar instâncias do EC2 com o Systems Manager

Conclua as tarefas desta seção para definir e configurar perfis, permissões e recursos iniciais para o AWS Systems Manager. As tarefas descritas nesta seção são normalmente executadas pela Conta da AWS e pelos administradores de sistemas. Quando essas etapas forem concluídas, os usuários

na organização poderão usar o Systems Manager para configurar, gerenciar e acessar as instâncias do Amazon Elastic Compute Cloud (Amazon EC2).

> **Note**
>
> Se você planeja usar o Systems Manager para gerenciar e configurar máquinas on-premises, siga as etapas de configuração em Gerenciar nós em ambientes híbridos e multinuvem com o Systems Manager. Para usar as instâncias do Amazon EC2 e máquinas que não são do EC2 em um ambiente híbrido e multinuvem, siga estas etapas primeiro. Esta seção apresenta etapas na ordem recomendada para configurar as funções, usuários, permissões e recursos iniciais para uso nas suas operações do Systems Manager.

Se você já usa outros Serviços da AWS, já concluiu algumas dessas etapas. No entanto, outras etapas são específicas ao Systems Manager. Portanto, recomendamos analisar toda esta seção para garantir que você está pronto para usar todas as ferramentas do Systems Manager.

Conteúdo

- Configurar permissões de instância obrigatórias para o Systems Manager
- Melhorar a segurança das instâncias do EC2 usando endpoints da VPC para o Systems Manager

Configurar permissões de instância obrigatórias para o Systems Manager

Por padrão, o AWS Systems Manager não tem permissão para executar ações em suas instâncias. É possível fornecer permissões de instância no nível da conta usando um perfil do AWS Identity and Access Management (IAM) ou no nível da instância usando um perfil de instância. Se o seu caso de uso permitir, recomendamos conceder acesso no nível da conta usando a configuração de gerenciamento de host padrão.

> **Note**
>
> Você pode pular essa etapa e permitir que o Systems Manager aplique as permissões necessárias às suas instâncias para você ao configurar o console unificado. Para obter mais informações, consulte Configurar o AWS Systems Manager.

Configuração recomendada para permissões de instâncias do EC2

A configuração de gerenciamento de host padrão permite que o Systems Manager gerencie as instâncias do Amazon EC2 automaticamente. Após ativar essa configuração, todas as instâncias que usam a versão 2 do serviço de metadados da instância (IMDSv2) na Região da AWS e na Conta da AWS com a versão 3.2.582.0 ou posterior do SSM Agent instalada automaticamente, tornam-se instâncias gerenciadas. A configuração de gerenciamento de host padrão não oferece suporte à versão 1 do serviço de metadados da instância. Para obter informações sobre como fazer a transição para IMDSv2, consulte Transição para usar o Serviço de metadados da instância versão 2 no Guia do usuário do Amazon EC2. Para obter informações sobre como verificar a versão do SSM Agent instalada em sua instância, consulte Verificar o número de versão do SSM Agent. Para obter informações sobre como atualizar o SSM Agent, consulte Atualizar automaticamente o SSM Agent. Os benefícios das instâncias gerenciadas incluem o seguinte:

- Conectar-se às suas instâncias com segurança usando o Session Manager.
- Realizar verificações de patches automatizadas usando o Patch Manager.
- Visualizar informações detalhadas sobre suas instâncias usando o Inventário do Systems Manager.
- Acompanhar e gerenciar instâncias usando o Fleet Manager.
- Manter o SSM Agent atualizado automaticamente.

Fleet Manager, Inventory, Patch Manager, e Session Manager são ferramentas do AWS Systems Manager.

A configuração de gerenciamento de host padrão permite o gerenciamento de instância sem o uso de perfis de instância e garante que o Systems Manager tenha permissões para gerenciar todas as instâncias na região e na conta. Se as permissões fornecidas não forem suficientes para seu caso de uso, também é possível adicionar políticas ao perfil do IAM padrão criado pela configuração de gerenciamento de host padrão. Como alternativa, se você não precisar de permissões para todas as funcionalidades fornecidas pelo perfil do IAM padrão, poderá criar seu próprio perfil e as políticas personalizadas. Quaisquer alterações realizadas no perfil do IAM que você escolher para a configuração de gerenciamento de host padrão se aplicarão a todas as instâncias do Amazon EC2 gerenciadas na região e na conta. Para obter mais informações sobre a política usada pela configuração de gerenciamento de host padrão, consulte Política gerenciada pela AWS: AmazonSSMManagedEC2InstanceDefaultPolicy. Para obter mais informações sobre a configuração

de gerenciamento do host padrão, consulte Gerenciar instâncias do EC2 automaticamente com a Configuração de gerenciamento de hosts padrão.

> **Important**
>
> As instâncias registradas usando a Configuração Padrão de Gerenciamento de Host armazenam informações de registro localmente nos diretórios `/lib/amazon/ssm` ou `C:\ProgramData\Amazon`. A remoção desses diretórios ou de seus arquivos impedirá que a instância adquira as credenciais necessárias para se conectar ao Systems Manager usando a Configuração Padrão de Gerenciamento de Host. Nesses casos, você deve usar um perfil de instância para fornecer as permissões necessárias à sua instância, ou então recriar a instância.

> **Note**
>
> Esse procedimento deve ser executado somente por administradores. Implemente acesso com privilégio mínimo ao permitir que indivíduos configurem ou modifiquem a configuração de gerenciamento de host padrão. É necessário ativar a configuração de gerenciamento de host padrão em cada Região da AWS que deseja gerenciar automaticamente as instâncias do Amazon EC2.

Como ativar a definição de configuração de gerenciamento de host padrão

É possível ativar a configuração de gerenciamento de host padrão no console do Fleet Manager. Para concluir este procedimento com êxito usando o AWS Management Console ou sua ferramenta de linha de comando preferida, você deve ter permissões para as operações de API GetServiceSetting, ResetServiceSetting e UpdateServiceSetting. Além disso, você deve ter permissões para a permissão `iam:PassRole` para o perfil do IAM `AWSSystemsManagerDefaultEC2InstanceManagementRole`. Veja abaixo um exemplo de política . Substitua cada *`espaço reservado para recurso de exemplo`* por suas próprias informações.

JSON

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

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetServiceSetting",
                "ssm:ResetServiceSetting",
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:servicesetting/ssm/
managed-instance/default-ec2-instance-management-role"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/service-role/
AWSSystemsManagerDefaultEC2InstanceManagementRole",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ssm.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

Antes de começar, se você tiver perfis de instância anexados às suas instâncias do Amazon EC2, remova todas as permissões que permitem a operação `ssm:UpdateInstanceInformation`. O SSM Agent tenta usar as permissões de perfil de instância antes de usar as permissões de configuração de gerenciamento de host padrão. Se você permitir a operação `ssm:UpdateInstanceInformation` em seus perfis de instância, a instância não usará as permissões de configuração de gerenciamento de host padrão.

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Escolha Definir Configuração de gerenciamento de host padrão no menu suspenso Gerenciamento de contas.

4. Ative Habilitar a configuração de gerenciamento de host padrão.
5. Escolha o perfil do IAM usado para habilitar as ferramentas do Systems Manager para suas instâncias. Recomendamos usar o perfil padrão fornecido pela configuração de gerenciamento de host padrão. Ele contém o conjunto mínimo de permissões necessárias para gerenciar as instâncias do Amazon EC2 usando o Systems Manager. Se você preferir usar um perfil personalizado, a política de confiança do perfil deve permitir que o Systems Manager seja uma entidade confiável.
6. Escolha Configurar para concluir a configuração.

Após ativar a configuração de gerenciamento de host padrão, pode demorar até 30 minutos para que as instâncias usem as credenciais do perfil escolhido. É necessário ativar a configuração de gerenciamento de host padrão em cada região que deseja gerenciar automaticamente as instâncias do Amazon EC2.

Configuração alternativa para permissões de instâncias do EC2

É possível conceder acesso no nível da instância individual usando um perfil de instância do AWS Identity and Access Management (IAM). Um perfil de instância é um contêiner que transmite as informações da função do IAM para uma instância do Amazon Elastic Compute Cloud (Amazon EC2) na inicialização. Você pode criar um perfil de instância para o Systems Manager anexando uma ou mais políticas do IAM que definem as permissões necessárias para uma nova função ou uma função que você já tnha criado.

> **Note**
>
> Você pode usar o Quick Setup, uma ferramenta do AWS Systems Manager, para configurar rapidamente um perfil de instância em todas as instâncias em seu Conta da AWS. O Quick Setup também cria um perfil de serviço do IAM (ou perfil assume), o que permite que o Systems Manager execute com segurança comandos em seu nome em suas instâncias. Com o uso do Quick Setup, pode-se ignorar esta etapa (Etapa 3) e a Etapa 4. Para obter mais informações, consulte AWS Systems Manager Quick Setup.

Observe os seguintes detalhes sobre a criação de um perfil da instância do IAM:

- Se estiver configurando máquinas que não são do EC2 em um ambiente híbrido e multinuvem para o Systems Manager, não é necessário criar um perfil de instância para elas. Em vez disso,

configure seus servidores e VMs para usar uma função de serviço do IAM. Para obter mais informações, consulte Criar o perfil de serviço do IAM obrigatório para o Systems Manager em ambientes híbridos e multinuvem.

- Se você alterar o perfil da instância do IAM, poderá levar algum tempo até as credenciais da instância serem atualizadas. O SSM Agent não processará solicitações até que isso aconteça. Para acelerar o processo de atualização, reinicie o SSM Agent ou a instância.

Se você estiver criando uma nova função para o perfil da instância ou adicionando as permissões necessárias a uma função existente, use um dos procedimentos a seguir.

Para criar um perfil de instância para instâncias gerenciadas do Systems Manager (console)

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Roles e Create role.
3. Em Trusted entity type (Tipo da entidade confiável), escolha AWS service (Serviço da AWS).
4. Diretamente sob Use case (Caso de uso), escolha EC2 e Next (Avançar).
5. Na página Add permissions (Adicionar permissões), faça o seguinte:

   - Use o campo Search (Pesquisar) para localizar AmazonSSMManagedInstanceCore. Marque a caixa de seleção próximo a seu nome, conforme mostrado na ilustração a seguir.



   O console reterá sua seleção mesmo que você pesquise outras políticas.

   - Se você tiver criado uma política de bucket do S3 personalizada no procedimento anterior, (Opcional) Criação de uma política personalizada para acesso ao bucket do S3, marque a caixa de seleção ao lado do nome dela.
   - Se você planejar ingressar instâncias em um Active Directory gerenciado pelo AWS Directory Service, pesquise por AmazonSSMDirectoryServiceAccess e marque a caixa de seleção ao lado do nome.

- Se você planejar usar o EventBridge ou o CloudWatch Logs para gerenciar ou monitorar sua instância, pesquise por CloudWatchAgentServerPolicy e marque a caixa de seleção ao lado do nome.

6. Escolha Próximo.
7. Em Role name (Nome da função), insira um nome para seu novo perfil da instância, como **`SSMInstanceProfile`**.

> **Note**
>
> Anote o nome da função. Você escolherá essa função ao criar novas instâncias que deseja gerenciar usando o Systems Manager.

8. (Opcional) Em Description (Descrição), atualize a descrição deste perfil de instância.
9. (Opcional) Em Tags (Etiquetas), adicione um ou mais pares de valores etiqueta-chave para organizar, monitorar ou controlar o acesso para esse perfil e, em seguida, escolha Create role (Criar perfil). O sistema faz com que você retorne para a página Roles.

Para adicionar permissões de perfil de instância para o Systems Manager a uma função existente (console)

1. Abra o console do IAM, em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Roles (Funções) e, em seguida, escolha a função existente que você deseja associar a um perfil de instância para operações do Systems Manager.
3. Na guia Permissions (Permissões), escolha Add permissions, Attach policies (Adicionar permissões, anexar políticas).
4. Na página Attach Policy (Anexar política), faça o seguinte:

   - Use o campo Search (Pesquisar) para localizar AmazonSSMManagedInstanceCore. Marque a caixa de seleção ao lado do nome.
   - Se você tiver criado uma política de bucket do S3 personalizada, procure-a e marque a caixa de seleção ao lado do nome dela. Para obter informações sobre políticas de bucket do S3 personalizadas para um perfil de instância, consulte (Opcional) Criação de uma política personalizada para acesso ao bucket do S3.

- Se você planejar ingressar instâncias em um Active Directory gerenciado pelo AWS Directory Service, pesquise por AmazonSSMDirectoryServiceAccess e marque a caixa de seleção ao lado do nome.
- Se você planejar usar o EventBridge ou o CloudWatch Logs para gerenciar ou monitorar sua instância, pesquise por CloudWatchAgentServerPolicy e marque a caixa de seleção ao lado do nome.

5. Escolha Anexar políticas.

Para obter mais informações sobre como atualizar uma função para incluir uma entidade confiável ou restringir ainda mais o acesso, consulte Modificar uma função no Guia do usuário do IAM.

(Opcional) Criação de uma política personalizada para acesso ao bucket do S3

A criação de uma política personalizada para acesso ao Amazon S3 é necessária se você usa um endpoint da VPC ou um bucket do S3 próprio nas suas operações do Systems Manager. É possível anexar essa política ao perfil do IAM padrão criado pela configuração de gerenciamento de host padrão ou a um perfil de instância criado no procedimento anterior.

Para obter informações sobre os buckets do S3 gerenciados pela AWS aos quais você fornece acesso na política a seguir, consulte Comunicações do SSM Agent com os buckets do S3 gerenciados pela AWS.

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Políticas e, em seguida, Criar política.
3. Escolha a guia JSON e substitua o texto padrão conforme a seguir.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": [
                "arn:aws:s3:::aws-ssm-us-east-2/*",
                "arn:aws:s3:::aws-windows-downloads-us-east-2/*",
                "arn:aws:s3:::amazon-ssm-us-east-2/*",
```

```
                "arn:aws:s3:::amazon-ssm-packages-us-east-2/*",
                "arn:aws:s3:::us-east-2-birdwatcher-prod/*",
                "arn:aws:s3:::aws-ssm-distributor-file-us-east-2/*",
                "arn:aws:s3:::aws-ssm-document-attachments-us-east-2/*",
                "arn:aws:s3:::patch-baseline-snapshot-us-east-2/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:GetEncryptionConfiguration"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/*",
                "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
        }
    ]
}
```

[1] O primeiro elemento `Statement` será necessário somente se você estiver usando um endpoints da VPC.

[2] O segundo elemento, `Statement`, será necessário somente se você estiver usando um bucket do S3 que você criou para usar em suas operações do Systems Manager.

[3] A permissão `PutObjectAcl` da lista de controle de acesso será necessária somente se você planeja oferecer suporte ao acesso entre contas para buckets do S3 em outras contas.

[4] O elemento `GetEncryptionConfiguration` será necessário se o bucket do S3 estiver configurado para usar criptografia.

[5] Se o bucket do S3 estiver configurado para usar criptografia, a raiz do bucket do S3 (por exemplo, `arn:aws:s3:::amzn-s3-demo-bucket`) deverá estar listada na seção Resource (Recurso). Seu usuário, grupo ou perfil deve ser configurado com acesso ao bucket raiz.

4. Se você estiver usando um endpoint da VPC em suas operações, faça o seguinte:

No primeiro elemento `Statement`, substitua cada espaço reservado *`região`* pelo identificador da Região da AWS na qual essa política será usada. Por exemplo, use `us-east-2` para a região Leste dos EUA (Ohio). Para ver uma lista dos valores de *`região`* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

> **Important**
>
> Recomendamos que você evite usar caracteres curinga (*) no lugar das regiões específicas nessa política. Por exemplo, use `arn:aws:s3:::aws-ssm-us-east-2/*` e não use `arn:aws:s3:::aws-ssm-*/*`. O uso de curingas pode fornecer acesso a buckets do S3 aos quais você não pretende conceder acesso. Se você quiser usar o perfil de instância para mais de uma região, recomendamos repetir o primeiro elemento `Statement` para cada região.

- ou -

Se não estiver usando um endpoint da VPC em suas operações, você poderá excluir o primeiro elemento `Statement`.

5. Se você estiver usando um bucket do S3 de sua propriedade em suas operações do Systems Manager, faça o seguinte:

   No segundo elemento `Statement`, substitua *`amzn-s3-demo-bucket`* pelo nome de um bucket do S3 em sua conta. Você usará esse bucket para suas operações do Systems Manager. Ele fornecerá permissão para objetos no bucket, usando `"arn:aws:s3:::my-bucket-name/*"` como o recurso. Para obter mais informações sobre como fornecer permissões para buckets ou objetos em buckets, consulte o tópico Ações do Amazon S3 no Guia do usuário do Amazon Simple Storage Service e a postagem do blog da AWS IAM Policies and Bucket Policies and ACLs! (Políticas de buckets e do IAM e ACLs!). Nossa! (Controlar o acesso aos recursos do S3).

   > **Note**
   >
   > Se você usar mais de um bucket, forneça o ARN de cada um. Veja o exemplo a seguir sobre permissões em buckets.
   >
   > ```
   > "Resource": [
   > ```

```
"arn:aws:s3:::amzn-s3-demo-bucket1/*",
"arn:aws:s3:::amzn-s3-demo-bucket2/*"
            ]
```

- ou -

Se não estiver usando um bucket do S3 de sua propriedade em suas operações do Systems Manager, você poderá excluir o segundo elemento `Statement`.

6. Escolha Próximo: tags.
7. (Opcional) Adicione tags escolhendo Add tag (Adicionar tag) e inserindo as tags preferenciais para a política.
8. Escolha Próximo: revisar.
9. Em Name (Nome), insira um nome para identificar essa política, como **`SSMInstanceProfileS3Policy`**.
10. Escolha Criar política.

Considerações de política adicionais para instâncias gerenciadas

Esta seção descreve algumas das políticas que você pode adicionar ao perfil do IAM padrão criado pela configuração de gerenciamento de host padrão ou seus perfis de instância para o AWS Systems Manager. Para fornecer permissões para a comunicação entre instâncias e a API do Systems Manager, recomendamos criar políticas personalizadas que reflitam as necessidades do sistema e os requisitos de segurança. Dependendo do seu plano de operações, você pode precisar de permissões representadas em uma ou mais das outras políticas.

Política: **`AmazonSSMDirectoryServiceAccess`**

Obrigatória somente se você planejar incluir instâncias do Amazon EC2 para Windows Server em um diretório do Microsoft AD.

Essa política gerenciada da AWS permite que o SSM Agent acesse o AWS Directory Service em seu nome para solicitações para ingressar no domínio pela instância gerenciada. Para obter mais informações, consulte Integrar-se facilmente a uma instância do Windows EC2 no Guia de administração do AWS Directory Service.

Política: **`CloudWatchAgentServerPolicy`**

Exigido apenas se você planeja instalar e executar o agente CloudWatch em suas instâncias para ler dados métricos e de logs em uma instância e gravá-los no Amazon CloudWatch. Eles ajudam você a monitorar, analisar e responder rapidamente a problemas ou alterações em seus recursos da AWS.

Seu perfil do IAM padrão criado pela configuração de gerenciamento de host padrão ou o perfil de instância precisa dessa política somente se você pretende usar recursos como o Amazon EventBridge ou o Amazon CloudWatch Logs. (Você também pode criar uma política mais restritiva que, por exemplo, limita o acesso à gravação a um determinado fluxo de logs do CloudWatch Logs.)

> **Note**
>
> O uso dos recursos do EventBridge e do CloudWatch Logs é opcional. Contudo, recomendamos configurá-los no início do seu processo de configuração do Systems Manager caso tenha decidido usá-los. Para obter mais informações, consulte o Guia do usuário do Amazon EventBridge e o Guia do usuário do Amazon CloudWatch Logs.

Para criar políticas do IAM com permissões para ferramentas adicionais do Systems Manager, consulte os recursos a seguir:

- Restringir o acesso a parâmetros do Parameter Store usando políticas do IAM
- Configurar a automação
- Etapa 2: verificar ou adicionar permissões de instância para o Session Manager

Anexe o perfil de instância do Systems Manager a uma instância (console)

O procedimento a seguir descreve como anexar um perfil de instância do IAM a uma instância do Amazon EC2 usando o console do Amazon EC2.

1. Faça login no AWS Management Console e abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Instâncias, escolha Instâncias.
3. Navegue até a lista e escolha a instância do EC2 na lista.

4. No menu Actions (Açõeescolha s), Security (Segurança), Modify IAM role (Modificar função do IAM).
5. Em IAM role (Função do IAM), selecione o perfil da instância que você criou usando o procedimento em Configuração alternativa para permissões de instâncias do EC2.
6. Escolha Update IAM role (Atualizar perfil do IAM).

Para obter mais informações sobre como anexar funções do IAM a instâncias, escolha uma das seguintes opções, dependendo do tipo de sistema operacional selecionado:

- Anexar um perfil do IAM a uma instância no Guia do usuário do Amazon EC2
- Anexar um perfil do IAM a uma instância no Guia do usuário do Amazon EC2

Avance para Melhorar a segurança das instâncias do EC2 usando endpoints da VPC para o Systems Manager.

Melhorar a segurança das instâncias do EC2 usando endpoints da VPC para o Systems Manager

Você pode melhorar o procedimento de segurança de seus nós gerenciados (inclusive máquinas que não são do EC2 em um ambiente híbrido e multinuvem) configurando o AWS Systems Manager para usar um endpoint da VPC de interface na Amazon Virtual Private Cloud (Amazon VPC). Por meio de um endpoint da VPC de interface (endpoint de interface), você pode se conecter a serviços com a tecnologia AWS PrivateLink. AWS PrivateLink é uma tecnologia que permite acessar de forma privada APIs da Amazon Elastic Compute Cloud (Amazon EC2) e do Systems Manager usando endereços IP privados.

O AWS PrivateLink limita todo o tráfego de rede entre as instâncias gerenciadas, o Systems Manager, o Amazon EC2 e a rede da Amazon. Isso significa que suas instâncias gerenciadas não precisam ter acesso à Internet. Se você usa o AWS PrivateLink, não precisa de um gateway da Internet, de um dispositivo NAT ou de um gateway privado virtual.

Não é necessário configurar o AWS PrivateLink, mas é recomendável. Para obter mais informações sobre o AWS PrivateLink e os endpoints da VPC, consulte AWS PrivateLink e endpoints da VPC.

> **Note**
>
> A alternativa ao uso de um endpoint da VPC é permitir o acesso à Internet de saída em suas instâncias gerenciadas. Nesse caso, as instâncias gerenciadas também devem permitir tráfego de saída HTTPS (porta 443) para os seguintes endpoints:
>
> - `ssm.region.amazonaws.com`
> - `ssmmessages.region.amazonaws.com`
> - `ec2messages.region.amazonaws.com`
>
> O SSM Agent inicia todas as conexões com o serviço Systems Manager na nuvem. Por essa razão, você não precisa configurar o firewall para permitir o tráfego de entrada nas instâncias para o Systems Manager.
> Para obter mais informações sobre chamadas para esses endpoints, consulte Referência: ec2messages, ssmmessages e outras operações da API.

Sobre a Amazon VPC

O Amazon Virtual Private Cloud (Amazon VPC) permite definir uma rede virtual em sua própria área isolada logicamente na Nuvem AWS, conhecida como uma nuvem privada virtual (VPC). Você pode iniciar os recursos da AWS, como as instâncias, na VPC. Sua VPC assemelha-se a uma rede tradicional que você poderia operar no seu próprio data center, com os benefícios de usar a infraestrutura escalável da AWS. É possível configurar seu VPC, selecionar o intervalo de endereços IP dele, criar sub-redes e definir tabelas de rotas, gateways de rede e configurações de segurança. Você pode se conectar às suas instâncias na VPC. É possível conectar a VPC a seu próprio data center corporativo, tornando a Nuvem AWS uma extensão do seu data center. Para proteger os recursos em cada sub-rede, você pode usar várias camadas de segurança, incluindo grupos de segurança e listas de controle de acesso de rede. Para obter mais informações, consulte o Manual do usuário da Amazon VPC.

Tópicos

- Restrições e limitações do VPC endpoint
- Criar endpoints da VPC para o Systems Manager
- Criar uma política de VPC endpoint de interface

Restrições e limitações do VPC endpoint

Antes de configurar endpoints da VPC para o Systems Manager, fique atento às restrições e limitações a seguir.

Conexões de emparelhamento da VPC

Os endpoints da interface da VPC podem ser acessados por meio de conexões de emparelhamento de VPC dentro da região e entre regiões. Para obter mais informações sobre solicitações de conexão de emparelhamento de VPC para endpoints de interface da VPC, consulte Conexões de emparelhamento da VPC (Cotas) no Guia do usuário da Amazon Virtual Private Cloud.

Não é possível estender conexões de endpoint de gateway de VPC para fora de uma VPC. Os recursos do outro lado de uma conexão de emparelhamento de VPC em sua VPC não podem usar o endpoint de gateway para se comunicar com recursos no serviço de endpoint de gateway. Para obter mais informações sobre solicitações de conexão de emparelhamento de VPC para endpoints do gateway da VPC, consulte Endpoints da VPC (Cotas) no Guia do usuário da Amazon Virtual Private Cloud.

Conexões de entrada

O grupo de segurança anexado ao VPC endpoint deve permitir conexões de entrada na porta 443 na sub-rede privada da instância gerenciada. Se conexões de entrada não são permitidas, a instância gerenciada não pode se conectar ao SSM e endpoints do EC2.

Resolução do DNS

Para usar um servidor DNS personalizado, você deve adicionar um encaminhador condicional para qualquer consulta ao domínio `amazonaws.com` para o servidor Amazon DNS de sua VPC.

Buckets do S3

Sua política de endpoint da VPC deve permitir acesso ao menos aos buckets do Amazon S3 listados em Comunicações do SSM Agent com os buckets do S3 gerenciados pela AWS.

> **Note**
>
> Se você usar um firewall on-premises e planeja usar o Patch Manager, esse firewall também deverá permitir o acesso ao endpoint da lista de referência de patches apropriado.

Amazon CloudWatch Logs

Se você não permitir que suas instâncias acessem a Internet, crie um endpoint da VPC para que o CloudWatch Logs use recursos que enviem logs para o CloudWatch Logs. Para obter mais informações sobre como criar um endpoint para o CloudWatch Logs, consulte Criar um endpoint da VPC para o CloudWatch Logs no Manual do usuário do Amazon CloudWatch Logs.

DNS em ambiente híbrido e multinuvem

Para obter informações sobre como configurar o DNS para trabalhar com endpoints do AWS PrivateLink em ambientes híbridos e multinuvem, consulte Private DNS for interface endpoints no Guia do usuário da Amazon VPC. Se quiser usar seu próprio DNS, poderá usar o resolvedor do Route 53. Para obter mais informações, consulte Resolver consultas de DNS entre VPCs e sua rede no Guia do desenvolvedor do Amazon Route 53.

Criar endpoints da VPC para o Systems Manager

Use as informações a seguir para criar endpoints de interface da VPC para o AWS Systems Manager. Este tópico contém links para procedimentos noManual do usuário da Amazon VPC.

> **Note**
>
> A *`região`* representa o identificador da região para uma região da Região da AWS compatível com o AWS Systems Manager, como `us-east-2` para a região Leste dos EUA (Ohio). Para ver uma lista dos valores de *`região`* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

Siga as etapas em Create an interface endpoint (Criar um endpoint de interface) para criar os seguintes endpoints de interface:

- **`com.amazonaws.`*`region`*`.ssm`**: o endpoint para o serviço Systems Manager.
- **`com.amazonaws.`*`region`*`.ec2messages`**: o Systems Manager usa esse endpoint para fazer chamadas do SSM Agent para o serviço do Systems Manager. A partir da versão 3.3.40.0 do SSM Agent, o Systems Manager começou a usar o endpoint `ssmmessages:*` (Amazon Message Gateway Service) sempre que disponível, em vez do endpoint `ec2messages:*` (Amazon Message Delivery Service).
- **`com.amazonaws.`*`region`*`.ec2`**: se você estiver usando o Systems Manager para criar snapshots habilitados para VSS, será necessário ter um endpoint para o serviço EC2. Sem o endpoint do EC2 definido, a chamada para enumerar os volumes anexados do Amazon EBS falha, o que faz com que o comando do Systems Manager falhe.

- **com.amazonaws.*region*.s3**: o Systems Manager usa esse endpoint para atualizarSSM Agent. O Systems Manager também usará esse endpoint se, opcionalmente, você optar por recuperar scripts ou outros arquivos armazenados em buckets ou fazer upload dos logs de saída para buckets. Se o grupo de segurança associado a suas instâncias restringir o tráfego de saída, será preciso adicionar uma regra para permitir o tráfego para a lista de prefixos do Amazon S3. Para obter mais informações, consulte Modificar seu grupo de segurança no Guia do AWS PrivateLink.
- **com.amazonaws.*region*.ssmmessages** — Este endpoint será necessário para que o SSM Agent se comunique com o serviço Systems Manager, para Run Command, e se você estiver se conectando às suas instâncias por meio de um canal de dados seguro usando o Session Manager. Para obter mais informações, consulte AWS Systems Manager Session Manager e Referência: ec2messages, ssmmessages e outras operações da API.
- (Opcional) **com.amazonaws.*region*.kms**: crie esse endpoint se você quiser usar a criptografia AWS Key Management Service (AWS KMS) para os parâmetros do Session Manager ou do Parameter Store.
- (Opcional) **com.amazonaws.*region*.logs**: crie este endpoint se você deseja usar o Amazon CloudWatch Logs (CloudWatch Logs) para logs do Session Manager, Run Command ou SSM Agent.

Para obter informações sobre os buckets do S3 gerenciados pela AWS que o SSM Agent deve ser capaz de acessar, consulte Comunicações do SSM Agent com os buckets do S3 gerenciados pela AWS. Se você estiver usando um endpoint da nuvem privada virtual (VPC) nas operações do Systems Manager, deverá fornecer permissão explícita em um perfil de instância do EC2 para o Systems Manager ou em um perfil de serviço para nós gerenciados que não são do EC2 em um ambiente híbrido e multinuvem.

Criar uma política de VPC endpoint de interface

Você pode criar políticas de endpoints de interface da VPC para o AWS Systems Manager nas quais é possível especificar:

- A entidade principal que pode executar ações
- As ações que podem ser executadas
- Os recursos que podem ter ações executadas neles

Para obter mais informações, consulte Controlar o acesso a serviços com endpoints da VPCs no Guia do usuário da Amazon VPC.

Gerenciar nós em ambientes híbridos e multinuvem com o Systems Manager

Você pode usar o AWS Systems Manager para gerenciar instâncias do Amazon Elastic Compute Cloud (EC2) e vários tipos de máquinas que não são do EC2. Esta seção descreve as tarefas de configuração que os administradores de sistema e de conta executam para gerenciar máquinas que não são do EC2 usando o Systems Manager em um ambiente híbrido e multinuvem. Depois que essas etapas forem concluídas, os usuários que tiveram permissões concedidas pelo administrador da Conta da AWS poderão usar o Systems Manager para configurar e gerenciar as máquinas da organização que não são do EC2.

Qualquer máquina configurada para uso com o Systems Manager é chamada de nó gerenciado.

> **Note**
>
> - É possível registrar dispositivos de borda como nós gerenciados usando as mesmas etapas de ativação híbrida usadas para outras máquinas que não são do EC2. Esses tipos de dispositivos de borda incluem dispositivos do AWS IoT e dispositivos que não são dispositivos do AWS IoT. Use o processo descrito nesta seção para configurar esses tipos de dispositivos de borda.
>
> O Systems Manager também oferece suporte a dispositivos de borda que usam o software AWS IoT Greengrass Core. O processo de configuração e os requisitos dos dispositivos AWS IoT Greengrass principais são diferentes daqueles para dispositivos de AWS IoT de borda que não sejam dispositivos de borda da AWS. Para obter informações sobre o registro de dispositivos AWS IoT Greengrass para uso com o Systems Manager, consulte Gerenciar dispositivos de borda com o Systems Manager.
> - Máquinas do macOS que não sejam do EC2 não são compatíveis com ambientes híbridos e multinuvem do Systems Manager.

Se você planeja usar o Systems Manager para gerenciar instâncias do Amazon Elastic Compute Cloud (Amazon EC2) ou usar instâncias do Amazon EC2 e máquinas que não são do EC2 em um ambiente híbrido e multinuvem, siga primeiro as etapas descritas em Gerenciar instâncias do EC2 com o Systems Manager.

Após configurar o ambiente híbrido e multinuvem para o Systems Manager, é possível fazer o seguinte:

- Crie uma forma consistente e segura de gerenciar remotamente suas workloads híbridas e multinuvem com base em um local usando as mesmas ferramentas ou scripts.
- Centralize o controle de acesso para as ações que podem ser realizadas nas máquinas, usando o AWS Identity and Access Management (IAM).
- Centralize a auditoria das operações realizadas em suas máquinas visualizando a atividade da API registrada no AWS CloudTrail.

  Para obter informações sobre como usar o CloudTrail para monitorar ações do Systems Manager, consulte Registrar em log chamadas de API do AWS Systems Manager com o AWS CloudTrail.
- Centralize o monitoramento configurando o Amazon EventBridge e o Amazon Simple Notification Service (Amazon SNS) para enviarem notificações sobre o êxito da execução do serviço.

  Para obter informações sobre como usar o EventBridge para monitorar eventos do Systems Manager, consulte Monitorar eventos do Systems Manager com o Amazon EventBridge.

Sobre nós gerenciados

Depois de concluir a configuração de máquinas que não são do EC2 para o Systems Manager, conforme descrito nesta seção, as máquinas ativadas para ambiente híbrido serão listadas no AWS Management Console e descritas como nós gerenciados. No console, os IDs dos nós gerenciados ativados para ambiente híbrido são diferenciados das instâncias do Amazon EC2 com o prefixo “mi-”. Os IDs das instâncias do Amazon EC2 usam o prefixo “i-”.

Um nó gerenciado é qualquer máquina configurada para o Systems Manager. Anteriormente, todos os nós gerenciados eram chamados de instâncias gerenciadas. O termo instância agora se refere somente às instâncias do EC2. O comando deregister-managed-instance foi nomeado antes da mudança de terminologia.

Para ter mais informações, consulte Trabalhar com nós gerenciados.

Sobre níveis de instância

O Systems Manager oferece um nível de instâncias padrão e um nível de instâncias avançadas para nós gerenciados que não são do EC2 em seu ambiente híbrido e multinuvem. O nível de instâncias padrão permite registrar no máximo mil máquinas ativadas para ambiente híbrido por Conta da AWS

e por Região da AWS. Se precisar registrar mais de mil máquinas que não são do EC2 em uma única conta e região, use o nível de instâncias avançadas. Instâncias avançadas também permitem que você se conecte às suas máquinas que não são do EC2 usando o Session Manager do AWS Systems Manager. O Session Manager fornece acesso via shell interativo aos nós gerenciados.

Para ter mais informações, consulte Configurar níveis de instâncias.

Tópicos

- Criar o perfil de serviço do IAM obrigatório para o Systems Manager em ambientes híbridos e multinuvem
- Criar uma ativação híbrida para registrar nós no Systems Manager
- Instalar o SSM Agent em nós híbridos do Linux
- Instalar o SSM Agent em nós híbridos do Windows Server

Criar o perfil de serviço do IAM obrigatório para o Systems Manager em ambientes híbridos e multinuvem

Máquinas que não são do Amazon Elastic Compute Cloud (EC2) em um ambiente híbrido e multinuvem exigem um perfil de serviço do AWS Identity and Access Management (IAM) para se comunicarem com o serviço do AWS Systems Manager. A atribuição de função do AWS Security Token Service (AWS STS) `AssumeRole` confiança para o serviço Systems Manager. Só é necessário criar um perfil de serviço para um ambiente híbrido e multinuvem uma vez para cada Conta da AWS. No entanto, você poderá optar por criar vários perfis de serviço para diferentes ativações híbridas se as máquinas em seu ambiente híbrido e multinuvem exigirem permissões diferentes.

Os procedimentos a seguir descrevem como criar a função de serviço necessária usando o console do Systems Manager ou sua ferramenta de linha de comando preferida.

Usar o AWS Management Console para criar um perfil de serviço do IAM para ativações híbridas do Systems Manager

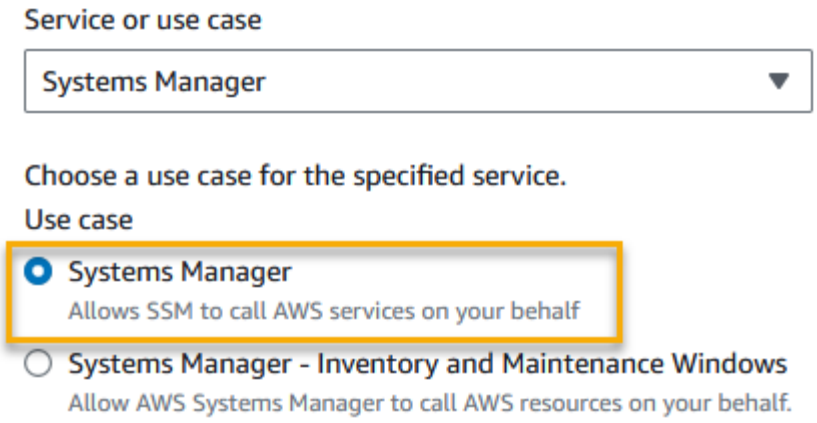
Use o procedimento a seguir para criar um perfil de serviço para a ativação híbrida. Este procedimento usa a política `AmazonSSMManagedInstanceCore` para a funcionalidade principal do Systems Manager. Dependendo do seu caso de uso, talvez seja necessário adicionar políticas à sua função de serviço em máquinas on-premises para poder acessar outras ferramentas do Systems

Manager ou Serviços da AWS. Por exemplo, sem acesso aos buckets necessários do Amazon Simple Storage Service (Amazon S3), gerenciados pela AWS, as operações de aplicação de patches do Patch Manager falham.

Como criar uma função de serviço do (console)

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Roles e Create role.
3. Em Select trusted entity (Selecionar entidade confiável), faça as seguintes escolhas:
   1. Em Tipo de Entidade Confiável, escolha AWS service (Serviço da AWS).
   2. Em Casos de uso para outros Serviços da AWS, escolha Systems Manager.
   3. Escolha Systems Manager.

   A imagem a seguir destaca a localização da opção Systems Manager.



4. Escolha Próximo.
5. Na página Add permissions (Adicionar permissões), faça o seguinte:
   - Use o campo Search (Pesquisar) para localizar AmazonSSMManagedInstanceCore. Marque a caixa de seleção próximo a seu nome, conforme mostrado na ilustração a seguir.



> **Note**
>
> O console reterá sua seleção mesmo que você pesquise outras políticas.

- Se você tiver criado uma política de bucket do S3 personalizada no procedimento anterior, (Opcional) Criação de uma política personalizada para acesso ao bucket do S3, procure-a e marque a caixa de seleção ao lado de seu nome.
- Se você planejar ingressar máquinas que não são do EC2 em um Active Directory gerenciado pelo AWS Directory Service, pesquise por AmazonSSMDirectoryServiceAccess e marque a caixa de seleção ao lado do nome.
- Se você pretende usar o EventBridge ou o CloudWatch Logs para gerenciar ou monitorar seu nó gerenciado, pesquise por CloudWatchAgentServerPolicy e marque a caixa de seleção ao lado do nome.

6. Escolha Próximo.
7. Em Nome do perfil, insira um nome para o novo perfil de servidor do IAM, como **`SSMServerRole`**.

> **Note**
>
> Anote o nome da função. Você escolherá esse perfil ao registrar novas máquinas que deseja gerenciar usando o Systems Manager.

8. (Opcional) Em Descrição, atualize a descrição deste perfil de servidor do IAM.
9. (Opcional) em Tags, adicione um ou mais pares de valores tag-chave para organizar, monitorar ou controlar acesso para essa função.
10. Selecione Create role (Criar função). O sistema faz com que você retorne para a página Roles.

Usar o AWS CLI para criar um perfil de serviço do IAM para ativações híbridas do Systems Manager

Use o procedimento a seguir para criar um perfil de serviço para a ativação híbrida. Este procedimento usa a política `AmazonSSMManagedInstanceCore` para a funcionalidade principal do Systems Manager. Dependendo do caso de uso, talvez seja necessário adicionar outras políticas ao perfil de serviço em máquinas não EC2 em um ambiente híbrido e multinuvem para poder acessar outras ferramentas ou Serviços da AWS.

Requisito de política do bucket do S3

Se qualquer um dos seguintes casos for verdadeiro, crie uma política de permissões personalizada do IAM e para os buckets do Amazon Simple Storage Service (Amazon S3) antes de concluir este procedimento:

- Caso 1: você está usando um endpoint da VPC para conectar de forma privada sua VPC aos Serviços da AWS compatíveis e aos serviços do endpoint da VPC com a tecnologia AWS PrivateLink.
- Caso 2: você planeja usar um bucket do Amazon S3 criado como parte das operações do Systems Manager, por exemplo, para armazenar a saída para comandos do Run Command ou as sessões do Session Manager em um bucket do S3. Antes de continuar, siga as etapas em Create a custom S3 bucket policy for an instance profile (Criar uma política personalizada de bucket do S3 para um perfil de instância). As informações sobre políticas de bucket do S3 neste tópico também se aplicam à sua função de serviço.

AWS CLI

Para criar um perfil de serviço do IAM para um ambiente híbrido e multinuvem (AWS CLI)

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Em sua máquina local, crie um arquivo de texto com um nome como `SSMService-Trust.json` com a política de confiança a seguir. Certifique-se de salvar o arquivo com a extensão de arquivo `.json`. Especifique a Conta da AWS e a Região da AWS no ARN onde você criou a ativação híbrida. Substitua os *`valores dos espaços reservados`* para ID da conta e região por suas próprias informações.

   JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"",
         "Effect":"Allow",
         "Principal":{
```

```
            "Service":"ssm.amazonaws.com"
         },
         "Action":"sts:AssumeRole",
         "Condition":{
            "StringEquals":{
               "aws:SourceAccount":"123456789012"
            },
            "ArnEquals":{
               "aws:SourceArn":"arn:aws:ssm:us-east-2:123456789012:*"
            }
         }
      }
   ]
}
```

3. Abra a AWS CLI e, no diretório em que você criou o arquivo JSON, execute o create-role para criar a função de serviço. Este exemplo cria uma função chamada `SSMServiceRole`. É possível escolher outro nome, se você preferir.

   Linux & macOS

   ```
   aws iam create-role \
       --role-name SSMServiceRole \
       --assume-role-policy-document file://SSMService-Trust.json
   ```

   Windows

   ```
   aws iam create-role ^
       --role-name SSMServiceRole ^
       --assume-role-policy-document file://SSMService-Trust.json
   ```

4. Execute o comando attach-role-policy da maneira a seguir para permitir que a função de serviço recém-criada crie um token de sessão. O token de sessão concede ao seu nó gerenciado permissão para executar comandos usando o Systems Manager.

   > **Note**
   >
   > As políticas que você adicionar para um perfil de serviço para nós gerenciados em um ambiente híbrido e multinuvem serão as mesmas políticas usadas para criar um perfil de instância para instâncias do Amazon Elastic Compute Cloud (Amazon EC2). Para obter mais informações sobre as políticas da AWS usadas nos comandos a

seguir, consulte Configurar permissões de instância obrigatórias para o Systems Manager.

(Obrigatório) Execute o comando a seguir para permitir que um nó gerenciado use a funcionalidade básica do serviço do AWS Systems Manager.

Linux & macOS

```
aws iam attach-role-policy \
    --role-name SSMServiceRole \
    --policy-arn arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
```

Windows

```
aws iam attach-role-policy ^
    --role-name SSMServiceRole ^
    --policy-arn arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
```

Se você tiver criado uma política de bucket do S3 personalizada para sua função de serviço, execute o comando a seguir para permitir que o AWS Systems Manager Agent (SSM Agent) acesse os buckets que você especificou na política. Substitua *account-id* e *amzn-s3-demo-bucket* pelo ID da Conta da AWS e o nome do bucket.

Linux & macOS

```
aws iam attach-role-policy \
    --role-name SSMServiceRole \
    --policy-arn arn:aws:iam::account-id:policy/amzn-s3-demo-bucket
```

Windows

```
aws iam attach-role-policy ^
    --role-name SSMServiceRole ^
    --policy-arn arn:aws:iam::account-id:policy/amzn-s3-demo-bucket
```

(Opcional) Execute o comando a seguir para permitir que o SSM Agent acesse o AWS Directory Service em seu nome para solicitações para ingressar no domínio pelo nó gerenciado. Seu perfil de serviço precisará dessa política somente se você integrar seus nós a um diretório do Microsoft AD.

Linux & macOS

```
aws iam attach-role-policy \
    --role-name SSMServiceRole \
    --policy-arn arn:aws:iam::aws:policy/AmazonSSMDirectoryServiceAccess
```

Windows

```
aws iam attach-role-policy ^
    --role-name SSMServiceRole ^
    --policy-arn arn:aws:iam::aws:policy/AmazonSSMDirectoryServiceAccess
```

(Opcional) Execute o comando a seguir para permitir que o agente do CloudWatch seja executado nos seus nós gerenciados. Este comando permite ler informações em um nó e gravá-las no CloudWatch. Seu perfil de serviço precisará dessa política somente se você pretende usar serviços como o Amazon EventBridge ou Amazon CloudWatch Logs.

```
aws iam attach-role-policy \
    --role-name SSMServiceRole \
    --policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
```

Tools for PowerShell

Para criar um perfil de serviço do IAM para um ambiente híbrido e multinuvem (AWS Tools for Windows PowerShell)

1. Instale e configure o Ferramentas da AWS para PowerShell (Ferramentas para Windows PowerShell), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar o Ferramentas da AWS para PowerShell.

2. Em sua máquina local, crie um arquivo de texto com um nome como `SSMService-Trust.json` com a política de confiança a seguir. Certifique-se de salvar o arquivo com a extensão de arquivo `.json`. Especifique a Conta da AWS e a Região da AWS no ARN onde você criou a ativação híbrida.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "ArnEquals": {
                    "aws:SourceArn": "arn:aws:ssm:us-
east-1:123456789012:*"
                }
            }
        }
    ]
}
```

3. Abra o PowerShell no modo administrativo e no diretório em que você criou o arquivo JSON, execute oNew-IAMRoleda seguinte maneira para criar uma função de serviço. Este exemplo cria uma função chamada `SSMServiceRole`. É possível escolher outro nome, se você preferir.

```
New-IAMRole `
    -RoleName SSMServiceRole `
    -AssumeRolePolicyDocument (Get-Content -raw SSMService-Trust.json)
```

4. Use Register-IAMRolePolicy conforme a seguir para permitir a função de serviço que você criou para criar um token de sessão. O token de sessão concede ao seu nó gerenciado permissão para executar comandos usando o Systems Manager.

> **Note**
>
> As políticas que você adicionar para um perfil de serviço para nós gerenciados em um ambiente híbrido e multinuvem serão as mesmas políticas usadas para criar um perfil de instância para instâncias do EC2. Para obter mais informações sobre as políticas da AWS usadas nos comandos a seguir, consulte Configurar permissões de instância obrigatórias para o Systems Manager.

(Obrigatório) Execute o comando a seguir para permitir que um nó gerenciado use a funcionalidade básica do serviço do AWS Systems Manager.

```
Register-IAMRolePolicy `
    -RoleName SSMServiceRole `
    -PolicyArn arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
```

Se você tiver criado uma política de bucket do S3 personalizada para sua função de serviço, execute o comando a seguir para permitir que o SSM Agent acesse os buckets que você especificou na política. Substitua *account-id* e *my-bucket-policy-name* pelo ID da Conta da AWS e o nome do bucket.

```
Register-IAMRolePolicy `
    -RoleName SSMServiceRole `
    -PolicyArn arn:aws:iam::account-id:policy/my-bucket-policy-name
```

(Opcional) Execute o comando a seguir para permitir que o SSM Agent acesse o AWS Directory Service em seu nome para solicitações para ingressar no domínio pelo nó gerenciado. Seu perfil de servidor precisará dessa política somente se você integrar seus nós a um diretório do Microsoft AD.

```
Register-IAMRolePolicy `
    -RoleName SSMServiceRole `
    -PolicyArn arn:aws:iam::aws:policy/AmazonSSMDirectoryServiceAccess
```

(Opcional) Execute o comando a seguir para permitir que o agente do CloudWatch seja executado nos seus nós gerenciados. Este comando permite ler informações em um nó e gravá-las no CloudWatch. Seu perfil de serviço precisará dessa política somente se você pretende usar serviços como o Amazon EventBridge ou Amazon CloudWatch Logs.

```
Register-IAMRolePolicy `
    -RoleName SSMServiceRole `
    -PolicyArn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
```

Avance para Criar uma ativação híbrida para registrar nós no Systems Manager.

Criar uma ativação híbrida para registrar nós no Systems Manager

Para configurar máquinas que não sejam instâncias do Amazon Elastic Compute Cloud (EC2) como nós gerenciados para um ambiente híbrido e multinuvem, crie e aplique uma ativação híbrida. Após concluir a ativação com êxito, você receberá imediatamente um código de ativação e um ID de ativação na parte de cima da página do console. Especifique essa combinação de código e ID ao instalar o AWS Systems Manager SSM Agent em máquinas que não são do EC2 em seu ambiente híbrido e multinuvem. O código e o ID fornecem acesso seguro ao serviço do Systems Manager a partir dos seus nós gerenciados.

> **Important**
>
> O Systems Manager retorna imediatamente o ID e o código de ativação para o console ou a janela de comando, dependendo de como você criou a ativação. Copie essas informações e armazene-as em um local seguro. Se você sair do console ou fechar a janela de comando, poderá perder essas informações. Se perdê-las, você deverá criar outra ativação.

Sobre expirações de ativação

Uma expiração de ativação é uma janela de tempo em que você pode registrar máquinas on-premises no Systems Manager. Uma ativação expirada não tem impacto sobre seus servidores ou VMs registradas anteriormente no Systems Manager. Se uma ativação expirar, você não poderá registrar mais servidores ou VMs no Systems Manager usando essa ativação específica. Você simplesmente precisa criar uma nova.

Cada VM e servidor on-premises registrado anteriormente permanecerá registrado como um nó gerenciado do Systems Manager até que você cancele o registro deles explicitamente. Você pode cancelar o registro de um nó gerenciado não EC2 das seguintes formas:

- Use a guia Nós gerenciados no Fleet Manager no console do Systems Manager
- Use o comando deregister-managed-instance da AWS CLI
- Use a ação da API DeregisterManagedInstance.

Para obter mais informações, consulte os tópicos a seguir

- Cancele o registro e registre um nó gerenciado novamente (Linux)
- Cancelar o registro e registrar um nó gerenciado novamente (Windows Server)

Sobre nós gerenciados

Um nó gerenciado é qualquer máquina configurada para o AWS Systems Manager. O AWS Systems Manager oferece suporte a instâncias, dispositivos de borda, servidores on-premises ou VMs do Amazon Elastic Compute Cloud (Amazon EC2), incluindo VMs em outros ambientes de nuvem. Anteriormente, todos os nós gerenciados eram chamados de instâncias gerenciadas. O termo instância agora se refere somente às instâncias do EC2. O comando deregister-managed-instance foi nomeado antes da mudança de terminologia.

Sobre tags de ativação

Se você criar uma ativação usando a AWS Command Line Interface (AWS CLI) ou o AWS Tools for Windows PowerShell, poderá especificar tags. Tags são metadados opcionais que você atribui a um recurso. As tags permitem categorizar um recurso de diferentes formas, como por finalidade, proprietário ou ambiente. Veja a seguir um comando de exemplo da AWS CLI a ser executado na região Leste dos EUA (Ohio) em uma máquina Linux local que inclui tags opcionais.

```
aws ssm create-activation \
  --default-instance-name MyWebServers \
  --description "Activation for Finance department webservers" \
  --iam-role service-role/AmazonEC2RunCommandRoleForManagedInstances \
  --registration-limit 10 \
  --region us-east-2 \
  --tags "Key=Department,Value=Finance"
```

Se você especificar tags ao criar uma ativação, essas tags serão atribuídas automaticamente aos seus nós gerenciados quando você ativá-los.

Você não pode adicionar ou excluir tags de uma ativação existente. Se você não quiser atribuir tags automaticamente aos servidores e VMs on-premises usando uma ativação, poderá adicionar tags a eles posteriormente. Mais especificamente, você poderá marcar os servidores on-premises e VMs depois que eles se conectarem ao Systems Manager pela primeira vez. Depois de se conectarem, eles receberão um ID de nó gerenciado e serão listados no console do Systems Manager com um ID prefixado com "mi-".

> **Note**
>
> Não é possível atribuir tags a uma ativação se você criá-la usando o console do Systems Manager. Você deve criá-la usando a AWS CLI ou Tools for Windows PowerShell.

Se não quiser mais gerenciar um servidor on-premises ou uma máquina virtual (VM) usando o Systems Manager, você poderá cancelar o registro. Para mais informações, consulte Cancelar o registro de nós gerenciados em um ambiente híbrido e multinuvem.

Tópicos

- Usar o AWS Management Console para criar uma ativação para registrar nós gerenciados com o Systems Manager
- Usar a linha de comando para criar uma ativação para registrar nós gerenciados com o Systems Manager

Usar o AWS Management Console para criar uma ativação para registrar nós gerenciados com o Systems Manager

Para criar uma ativação de nó gerenciado

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, selecione Ativações híbridas do.
3. Escolha Create activation.

   - ou -

Se estiver acessando Hibrid Activations (Ativações híbridas) pela primeira vez na Região da AWS atual, escolha Create an Activation (Criar uma ativação).

4. (Opcional) Em Activation description (Descrição da ativação), insira uma descrição para essa ativação. Recomendamos inserir uma descrição caso pretenda ativar um grande número de servidores e VMs.
5. Em Instance limit (Limite da instância), especifique o número total de nós que deseja registrar com a AWS como parte dessa ativação. O valor padrão é uma instância.
6. Em IAM role (Perfil do IAM), escolha uma opção de perfil de serviço que permita que seus servidores e VMs comuniquem-se com o AWS Systems Manager na nuvem:

   - Opção 1: escolha Use the default role created by the system (Usar o perfil padrão criado pelo sistema) para usar um perfil e uma política gerenciada fornecida pela AWS.
   - Opção 2: escolha Select an existing custom IAM role that has the required permissions (Selecionar um perfil do IAM personalizado existente que tenha as permissões necessárias) para usar o perfil personalizado opcional que você criou anteriormente. Essa função deve ter uma política de relação de confiança que especifique o`"Service": "ssm.amazonaws.com"`. Se sua função do IAM não especificar esse princípio em uma política de relacionamento de confiança, você receberá o seguinte erro:

     ```
     An error occurred (ValidationException) when calling the CreateActivation
                                        operation: Not existing role:
      arn:aws:iam::<accountid>:role/SSMRole
     ```

     Para obter mais informações sobre a criação do perfil, consulte Criar o perfil de serviço do IAM obrigatório para o Systems Manager em ambientes híbridos e multinuvem.
7. Em Activation expiry date (Data de expiração da ativação), especifique uma data de expiração para a ativação. A data de validade deve ser no futuro, e não mais de 30 dias no futuro. O valor padrão é 24 horas.

   > **Note**
   >
   > Se você quiser registrar nós gerenciados adicionais após a data de expiração, deverá criar uma nova ativação. A data de expiração não tem impacto sobre nós registrados e em execução.

8. (Opcional) No campo Default instance name (Nome da instância padrão), especifique um valor de nome de identificação a ser exibido para todos os nós gerenciados associados a essa ativação.
9. Escolha Create activation. O Systems Manager retorna imediatamente o código de ativação e o ID para o console.

Usar a linha de comando para criar uma ativação para registrar nós gerenciados com o Systems Manager

O procedimento a seguir descreve como usar a AWS Command Line Interface (AWS CLI) (no Linux ou no Windows Server) ou o Ferramentas da AWS para PowerShell para criar uma ativação de nó gerenciado.

Para criar uma ativação

1. Instale e configure a AWS CLI ou o Ferramentas da AWS para PowerShell, caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI e Instalar o Ferramentas da AWS para PowerShell.
2. Execute o seguinte comando para criar uma ativação.

> **Note**
>
> - No comando a seguir, substitua *`region`* por suas próprias informações. Para ver uma lista dos valores de *`região`* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.
> - A função que você especifica para o*`IAM role`*O parâmetro do precisa ter uma política de relação de confiança que especifique o`"Service": "ssm.amazonaws.com"`. Se suas receitas AWS Identity and Access Management (IAM) não especificar esse princípio em uma política de relacionamento de confiança, você recebe o seguinte erro:
>
> ```
> An error occurred (ValidationException) when calling the CreateActivation
>                                         operation: Not existing role:
>  arn:aws:iam::<accountid>:role/SSMRole
> ```

Para obter mais informações sobre a criação do perfil, consulte Criar o perfil de serviço do IAM obrigatório para o Systems Manager em ambientes híbridos e multinuvem.

- Para `--expiration-date`, forneça uma data no formato de carimbo de data/hora, como `"2021-07-07T00:00:00"`, para quando o código de ativação expirar. Você pode especificar uma data com até 30 dias de antecedência. Se você não fornecer uma data de expiração, o código de ativação expira em 24 horas.

Linux & macOS

```
aws ssm create-activation \
    --default-instance-name name \
    --iam-role iam-service-role-name \
    --registration-limit number-of-managed-instances \
    --region region \
    --expiration-date "timestamp" \\
    --tags "Key=key-name-1,Value=key-value-1" "Key=key-name-2,Value=key-value-2"
```

Windows

```
aws ssm create-activation ^
    --default-instance-name name ^
    --iam-role iam-service-role-name ^
    --registration-limit number-of-managed-instances ^
    --region region ^
    --expiration-date "timestamp" ^
    --tags "Key=key-name-1,Value=key-value-1" "Key=key-name-2,Value=key-value-2"
```

PowerShell

```
New-SSMActivation -DefaultInstanceName name `
    -IamRole iam-service-role-name `
    -RegistrationLimit number-of-managed-instances `
    -Region region `
    -ExpirationDate "timestamp" `
    -Tag @{"Key"="key-name-1";"Value"="key-value-1"},@{"Key"="key-
name-2";"Value"="key-value-2"}
```

Aqui está um exemplo.

Linux & macOS

```
aws ssm create-activation \
    --default-instance-name MyWebServers \
    --iam-role service-role/AmazonEC2RunCommandRoleForManagedInstances \
    --registration-limit 10 \
    --region us-east-2 \
    --expiration-date "2021-07-07T00:00:00" \
    --tags "Key=Environment,Value=Production" "Key=Department,Value=Finance"
```

Windows

```
aws ssm create-activation ^
    --default-instance-name MyWebServers ^
    --iam-role service-role/AmazonEC2RunCommandRoleForManagedInstances ^
    --registration-limit 10 ^
    --region us-east-2 ^
    --expiration-date "2021-07-07T00:00:00" ^
    --tags "Key=Environment,Value=Production" "Key=Department,Value=Finance"
```

PowerShell

```
New-SSMActivation -DefaultInstanceName MyWebServers `
    -IamRole service-role/AmazonEC2RunCommandRoleForManagedInstances `
    -RegistrationLimit 10 `
    –Region us-east-2 `
    -ExpirationDate "2021-07-07T00:00:00" `
    -Tag
 @{"Key"="Environment";"Value"="Production"},@{"Key"="Department";"Value"="Finance"}
```

Se a ativação for criada com êxito, o sistema retornará imediatamente um ID e código de ativação.

Instalar o SSM Agent em nós híbridos do Linux

Este tópico descreve como instalar o SSM Agent do AWS Systems Manager em máquinas Linux que não são do (EC2) Amazon Elastic Compute Cloud em um ambiente híbrido e multinuvem. Para obter informações sobre como instalar o SSM Agent em instâncias do EC2 para , consulte Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux.

Antes de começar, localize o Código de Ativação e o ID de Ativação que foram gerados durante o processo de ativação híbrida, conforme descrito em Criar uma ativação híbrida para registrar nós no Systems Manager. Você especifica o código e o ID no procedimento a seguir.

Para instalar o SSM Agent em máquinas que não sejam do EC2 em um ambiente híbrido e multinuvem

1. Faça logon em um servidor ou VM no seu ambiente híbrido ou multinuvem.
2. Se você usar um proxy HTTP ou HTTPS, defina o `http_proxy` ou as variáveis de ambiente `https_proxy` na sessão do shell atual. Se você não estiver usando um proxy, ignore esta etapa.

   Para um servidor proxy HTTP, insira os seguintes comandos na linha de comando:

   ```
   export http_proxy=http://hostname:port
   export https_proxy=http://hostname:port
   ```

   Para um servidor proxy HTTPS, insira os seguintes comandos na linha de comando:

   ```
   export http_proxy=http://hostname:port
   export https_proxy=https://hostname:port
   ```
3. Copie e cole um dos seguintes blocos de comandos no SSH. Substitua os valores de espaços reservados pelo Código de Ativação e o ID de Ativação gerados durante o processo de ativação híbrido e com o identificador da Região da AWS da qual deseja baixar o SSM Agent, e em seguida pressione `Enter`.

   Important

   Observe os seguintes detalhes importantes:

- O uso de `ssm-setup-cli` para instalações não EC2 maximiza a segurança da instalação e configuração do Systems Manager.
- O `sudo` não é necessário se você for um usuário root.
- Faça o download `ssm-setup-cli` da mesma Região da AWS em que sua ativação híbrida foi criada.
- `ssm-setup-cli` aceita uma opção `manifest-url` que determina a origem da qual o agente é baixado. Não especifique um valor para a opção, a menos que isso seja exigido pela sua organização.
- Ao registrar instâncias, somente use o link de download fornecido para `ssm-setup-cli`. `ssm-setup-cli` não deve ser armazenado separadamente para uso futuro.
- É possível usar o script fornecido aqui para validar a assinatura de `ssm-setup-cli`.

A *`região`* representa o identificador da região para uma região da Região da AWS compatível com o AWS Systems Manager, como `us-east-2` para a região Leste dos EUA (Ohio). Para ver uma lista dos valores de *`região`* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

Além disso, `ssm-setup-cli` inclui as opções a seguir:

- `version`: os valores válidos são `latest` e `stable`.
- `downgrade`: permite que o SSM Agent seja atualizado para uma versão anterior. Especifique `true` para instalar uma versão anterior do agente.
- `skip-signature-validation`: ignora a validação da assinatura durante o download e a instalação do agente.

RHEL 6.x e CentOS 6.x

```
mkdir /tmp/ssm
curl https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/3.0.1479.0/linux_amd64/
amazon-ssm-agent.rpm -o /tmp/ssm/amazon-ssm-agent.rpm
sudo yum install -y /tmp/ssm/amazon-ssm-agent.rpm
sudo stop amazon-ssm-agent
sudo -E amazon-ssm-agent -register -code "activation-code" -id "activation-id" -region
 "region"
sudo start amazon-ssm-agent
```

Amazon Linux 1

```
mkdir /tmp/ssm
curl https://amazon-ssm-region.s3.region.amazonaws.com/latest/linux_amd64/ssm-setup-cli
 -o /tmp/ssm/ssm-setup-cli
sudo chmod +x /tmp/ssm/ssm-setup-cli
sudo /tmp/ssm/ssm-setup-cli -register -activation-code "activation-code" -id
 "activation-id" -region "region"
```

Amazon Linux 2, RHEL 7.x, Oracle Linux, CentOS 7.x e SLES

```
mkdir /tmp/ssm
curl https://amazon-ssm-region.s3.region.amazonaws.com/latest/linux_amd64/ssm-setup-cli
 -o /tmp/ssm/ssm-setup-cli
sudo chmod +x /tmp/ssm/ssm-setup-cli
sudo /tmp/ssm/ssm-setup-cli -register -activation-code "activation-code" -activation-id
 "activation-id" -region "region"
```

RHEL 8.x e CentOS 8.x

```
mkdir /tmp/ssm
curl https://amazon-ssm-region.s3.region.amazonaws.com/latest/linux_amd64/ssm-setup-cli
 -o /tmp/ssm/ssm-setup-cli
sudo chmod +x /tmp/ssm/ssm-setup-cli
sudo /tmp/ssm/ssm-setup-cli -register -activation-code "activation-code" -activation-id
 "activation-id" -region "region"
```

Debian Server

```
mkdir /tmp/ssm
curl https://amazon-ssm-region.s3.region.amazonaws.com/latest/debian_amd64/ssm-setup-
cli -o /tmp/ssm/ssm-setup-cli
sudo chmod +x /tmp/ssm/ssm-setup-cli
sudo /tmp/ssm/ssm-setup-cli -register -activation-code "activation-code" -activation-id
 "activation-id" -region "region"
```

Raspberry Pi OS (anteriormente Raspbian)

```
mkdir /tmp/ssm
curl https://amazon-ssm-region.s3.region.amazonaws.com/latest/debian_arm/ssm-setup-cli
 -o /tmp/ssm/ssm-setup-cli
```

```
sudo chmod +x /tmp/ssm/ssm-setup-cli
sudo /tmp/ssm/ssm-setup-cli -register -activation-code "activation-code" -activation-id
 "activation-id" -region "region"
```

Ubuntu Server

- Usar pacotes .deb

```
mkdir /tmp/ssm
curl https://amazon-ssm-region.s3.region.amazonaws.com/latest/debian_amd64/ssm-setup-
cli -o /tmp/ssm/ssm-setup-cli
sudo chmod +x /tmp/ssm/ssm-setup-cli
sudo /tmp/ssm/ssm-setup-cli -register -activation-code "activation-code" -activation-
id "activation-id" -region "region"
```

- Usar pacotes de Snap

  Você não precisa especificar um URL para download porque o comando `snap` faz download automático do agente da loja de aplicativos Snap em https://snapcraft.io.

  No Ubuntu Server 20.10 STR e 20.04, 18.04 e 16.04 LTS, os arquivos do instalador do SSM Agent, incluindo arquivos binários do agente e arquivos de configuração, são armazenados no seguinte diretório: `/snap/amazon-ssm-agent/current/`. Se você fizer alterações em qualquer arquivo de configuração nesse diretório, copie esses arquivos do diretório `/snap` para o `/etc/amazon/ssm/`. Os arquivos de log e biblioteca não foram alterados (`/var/lib/amazon/ssm`, `/var/log/amazon/ssm`).

```
sudo snap install amazon-ssm-agent --classic
sudo systemctl stop snap.amazon-ssm-agent.amazon-ssm-agent.service
sudo /snap/amazon-ssm-agent/current/amazon-ssm-agent -register -code "activation-
code" -id "activation-id" -region "region"
sudo systemctl start snap.amazon-ssm-agent.amazon-ssm-agent.service
```

> **Important**
>
> Ocandidatona loja Snap contém a versão mais recente doSSM Agent; não o canal estável. Se você quiser acompanhar as informações de versão do SSM Agent no canal candidato, execute o comando a seguir em seus nós gerenciados do Ubuntu Server 18.04 e 16.04 LTS de 64 bits.

```
sudo snap switch --channel=candidate amazon-ssm-agent
```

O comando baixa e instala o SSM Agent na máquina ativada para ambiente híbrido em seu ambiente híbrido e multinuvem. O comando interrompe o SSM Agent e registra a máquina no serviço do Systems Manager. A máquina agora é um nó gerenciado. As instâncias do Amazon EC2 configuradas para o Systems Manager também são nós gerenciados. Porém, no console do Systems Manager, seus nós ativados para ambientes híbridos são diferenciados das instâncias do Amazon EC2 com o prefixo “mi-”.

Avance para Instalar o SSM Agent em nós híbridos do Windows Server.

Configurando a rotação automática da chave privada

Para fortalecer seu procedimento de segurança, você pode configurar o AWS Systems Manager Agent (SSM Agent) para alternar automaticamente a chave privada de seu ambiente híbrido e multinuvem. Você pode acessar esse recurso usando o SSM Agent versão 3.0.1031.0 ou posterior. Ative esse recurso usando procedimento a seguir.

Para configurar o SSM Agent para alternar a chave privada em um ambiente híbrido e multinuvem

1. Navegue até `/etc/amazon/ssm/` em uma máquina Linux ou `C:\Program Files\Amazon\SSM` em uma máquina Windows.
2. Copie o conteúdo do `amazon-ssm-agent.json.template` em um arquivo chamado `amazon-ssm-agent.json`. Salve o `amazon-ssm-agent.json` no mesmo diretório em que `amazon-ssm-agent.json.template` está localizado.
3. Localizar `Profile`, `KeyAutoRotateDays`. Insira o número de dias que você deseja entre as rotações automáticas de chave privada.
4. Reinicie o SSM Agent.

Toda vez que você alterar a configuração, reinicie o SSM Agent.

Você pode personalizar outros recursos do SSM Agent usando o mesmo procedimento. Para obter uma lista atualizada das propriedades de configuração disponíveis e seus valores padrão, consulte Config Property Definitions (Definições de Propriedades do Config).

Cancele o registro e registre um nó gerenciado novamente (Linux)

É possível cancelar o registro de um nó gerenciado ativado para ambiente híbrido chamando a operação da API DeregisterManagedInstance na AWS CLI ou em ferramentas para Windows PowerShell. Veja um exemplo de comando da CLI:

```
aws ssm deregister-managed-instance --instance-id "mi-1234567890"
```

Para remover as informações de registro restantes do agente, remova a chave `IdentityConsumptionOrder` no arquivo `amazon-ssm-agent.json`. Em seguida, dependendo do tipo de instalação, execute um dos comandos a seguir.

Nos nós Ubuntu Server em que o SSM Agent foi instalado usando pacotes Snap:

```
sudo /snap/amazon-ssm-agent/current/amazon-ssm-agent -register -clear
```

Em todas as outras instalações Linux:

```
amazon-ssm-agent -register -clear
```

Para registrar novamente um nó gerenciado em uma máquina Linux que não é do EC2

É possível registrar novamente uma máquina depois de cancelar o registro dela. No entanto, você deve usar um Código de Ativação e um ID de Ativação diferentes dos usados anteriormente para registrar a máquina.

1. Conectar-se à máquina.
2. Execute o seguinte comando: Substitua os valores dos espaços reservados pelo código de ativação e ID de ativação gerados ao criar uma ativação de nó gerenciado e pelo identificador da região da qual você quer baixar o SSM Agent.

   ```
   echo "yes" | sudo /tmp/ssm/ssm-setup-cli -register -activation-code "activation-code" -activation-id "activation-id" -region "region
   ```

Solução de problemas de instalação do SSM Agent em máquinas Linux que não são do EC2

Use as informações a seguir para ajudar você a solucionar problemas de instalação do SSM Agent em máquinas Linux ativadas para ambiente híbrido em um ambiente híbrido e multinuvem.

Você recebe o erro DeliveryTimedOut

Problema: ao configurar uma máquina em um Conta da AWS como um nó gerenciado para uma Conta da AWS separada, você receberá `DeliveryTimedOut` depois de executar os comandos para instalar o SSM Agent na máquina de destino.

Solução:`DeliveryTimedOut`é o código de resposta esperado para este cenário. O comando para instalar o SSM Agent no nó de destino altera o ID do nó do nó de origem. Como o ID do nó foi alterado, o nó de origem não é capaz de responder ao nó de destino que o comando falhou, foi concluído ou expirou durante a execução.

Não foi possível carregar associações de nós

Problema: Depois de executar os comandos de instalação, você verá o seguinte erro na seçãoSSM AgentLogs de erros:

```
Unable to load instance associations, unable to retrieve
associations unable to retrieve associations error occurred in
RequestManagedInstanceRoleToken: MachineFingerprintDoesNotMatch:
Fingerprint doesn't match
```

Esse erro é exibido quando o ID da máquina não persiste após uma reinicialização.

Solução: para corrigir esse problema, execute o comando a seguir. Esse comando força o ID da máquina a persistir após uma reinicialização.

```
umount /etc/machine-id
systemd-machine-id-setup
```

Instalar o SSM Agent em nós híbridos do Windows Server

Este tópico descreve como instalar o SSM Agent do AWS Systems Manager em máquinas do Windows Server em um ambiente híbrido e multinuvem. Para obter informações sobre como instalar o SSM Agent em instâncias do EC2 para Windows Server, consulte Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Windows Server.

Antes de começar, localize o Código de Ativação e o ID de Ativação que foram gerados durante o processo de ativação híbrida, conforme descrito em Criar uma ativação híbrida para registrar nós no Systems Manager. Você especifica o código e o ID no procedimento a seguir.

Para instalar o SSM Agent em máquinas Windows Server que não sejam do EC2 em um ambiente híbrido e multinuvem

1. Faça logon em um servidor ou VM no seu ambiente híbrido ou multinuvem.
2. Se você usar um proxy HTTP ou HTTPS, defina o `http_proxy` ou as variáveis de ambiente `https_proxy` na sessão do shell atual. Se você não estiver usando um proxy, ignore esta etapa.

   Para um servidor proxy HTTP, defina esta variável:

   ```
   http_proxy=http://hostname:port
   https_proxy=http://hostname:port
   ```

   Para um servidor proxy HTTPS, defina esta variável:

   ```
   http_proxy=http://hostname:port
   https_proxy=https://hostname:port
   ```
3. Abra o Windows PowerShell no modo elevado (administrativo).
4. Copie e cole um o bloco de comandos a seguir no Windows PowerShell. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações. Por exemplo, o código de ativação e ID de ativação gerados ao criar uma ativação híbrida e pelo identificador da Região da AWS em que você deseja baixar o SSM Agent.

   > **Important**
   >
   > Observe os seguintes detalhes importantes:
   >
   > - O uso de `ssm-setup-cli` para instalações não EC2 maximiza a segurança da instalação e configuração do Systems Manager.
   > - `ssm-setup-cli` aceita uma opção `manifest-url` que determina a origem da qual o agente é baixado. Não especifique um valor para a opção, a menos que isso seja exigido pela sua organização.
   > - É possível usar o script fornecido aqui para validar a assinatura de `ssm-setup-cli`.
   > - Ao registrar instâncias, somente use o link de download fornecido para `ssm-setup-cli`. `ssm-setup-cli` não deve ser armazenado separadamente para uso futuro.

A *região* representa o identificador da região para uma região da Região da AWS compatível com o AWS Systems Manager, como `us-east-2` para a região Leste dos EUA (Ohio). Para ver uma lista dos valores de *região* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

Além disso, `ssm-setup-cli` inclui as opções a seguir:

- `version`: os valores válidos são `latest` e `stable`.
- `downgrade`: reverte o agente para uma versão anterior.
- `skip-signature-validation`: ignora a validação da assinatura durante o download e a instalação do agente.

64-bit

```
[System.Net.ServicePointManager]::SecurityProtocol = 'TLS12'
$code = "activation-code"
$id = "activation-id"
$region = "us-east-1"
$dir = $env:TEMP + "\ssm"
New-Item -ItemType directory -Path $dir -Force
cd $dir
(New-Object System.Net.WebClient).DownloadFile("https://amazon-ssm-$region.s3.
$region.amazonaws.com/latest/windows_amd64/ssm-setup-cli.exe", $dir + "\ssm-
setup-cli.exe")
./ssm-setup-cli.exe -register -activation-code="$code" -activation-id="$id" -
region="$region"
Get-Content ($env:ProgramData + "\Amazon\SSM\InstanceData\registration")
Get-Service -Name "AmazonSSMAgent"
```

32-bit

```
"[System.Net.ServicePointManager]::SecurityProtocol = 'TLS12'"
$code = "activation-code"
$id = "activation-id"
$region = "us-east-1"
$dir = $env:TEMP + "\ssm"
New-Item -ItemType directory -Path $dir -Force
cd $dir
```

```
(New-Object System.Net.WebClient).DownloadFile("https://amazon-ssm-$region.s3.
$region.amazonaws.com/latest/windows_386/ssm-setup-cli.exe", $dir + "\ssm-setup-
cli.exe")
./ssm-setup-cli.exe -register -activation-code="$code" -activation-id="$id" -
region="$region"
Get-Content ($env:ProgramData + "\Amazon\SSM\InstanceData\registration")
Get-Service -Name "AmazonSSMAgent"
```

5. Pressione `Enter`.

> **Note**
>
> Se o comando falhar, verifique se você está executando a versão mais recente do Ferramentas da AWS para PowerShell.

O comando faz o seguinte:

- Baixa e instala o SSM Agent na máquina.
- Registra a máquina no serviço Systems Manager.
- Retorna uma resposta à solicitação semelhante à seguinte:

```
    Directory: C:\Users\ADMINI~1\AppData\Local\Temp\2


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----       07/07/2018   8:07 PM                ssm
{"ManagedInstanceID":"mi-008d36be46EXAMPLE","Region":"us-east-2"}

Status      : Running
Name        : AmazonSSMAgent
DisplayName : Amazon SSM Agent
```

A máquina agora é um nó gerenciado. Esses nós gerenciados agora são identificados com o prefixo "mi-". Você pode visualizar os nós gerenciados na página Nós gerenciados no Fleet Manager, usando o comando da AWS CLI describe-instance-information ou usando o comando da API DescribeInstanceInformation.

Configurando a rotação automática da chave privada

Para fortalecer seu procedimento de segurança, você pode configurar o AWS Systems Manager Agent (SSM Agent) para alternar automaticamente a chave privada de um ambiente híbrido e multinuvem. Você pode acessar esse recurso usando o SSM Agent versão 3.0.1031.0 ou posterior. Ative esse recurso usando procedimento a seguir.

Para configurar o SSM Agent para alternar a chave privada em um ambiente híbrido e multinuvem

1. Navegue até `/etc/amazon/ssm/` em uma máquina Linux ou `C:\Program Files\Amazon\SSM` em uma máquina Windows Server.
2. Copie o conteúdo do `amazon-ssm-agent.json.template` em um arquivo chamado `amazon-ssm-agent.json`. Salve o `amazon-ssm-agent.json` no mesmo diretório em que `amazon-ssm-agent.json.template` está localizado.
3. Localizar `Profile`, `KeyAutoRotateDays`. Insira o número de dias que você deseja entre as rotações automáticas de chave privada.
4. Reinicie o SSM Agent.

Toda vez que você alterar a configuração, reinicie o SSM Agent.

Você pode personalizar outros recursos do SSM Agent usando o mesmo procedimento. Para obter uma lista atualizada das propriedades de configuração disponíveis e seus valores padrão, consulte Config Property Definitions (Definições de Propriedades do Config).

Cancelar o registro e registrar um nó gerenciado novamente (Windows Server)

É possível cancelar o registro de um nó gerenciado chamando a operação da API DeregisterManagedInstance na AWS CLI ou no Tools for Windows PowerShell. Veja um exemplo de comando da CLI:

```
aws ssm deregister-managed-instance --instance-id "mi-1234567890"
```

Para remover as informações de registro restantes do agente, remova a chave `IdentityConsumptionOrder` no arquivo `amazon-ssm-agent.json`. Em seguida, execute o seguinte comando:

```
amazon-ssm-agent -register -clear
```

Para registrar novamente um nó gerenciado em uma máquina híbrida Windows Server

É possível registrar novamente uma máquina depois de cancelar o registro dela. No entanto, você deve usar um Código de Ativação e um ID de Ativação diferentes dos usados anteriormente para registrar a máquina.

1. Conectar-se à máquina.
2. Execute o seguinte comando: Substitua os valores dos espaços reservados pelo código de ativação e ID de ativação gerados ao criar uma ativação híbrida e pelo identificador da região da qual você quer baixar o SSM Agent.

```
'yes' | & Start-Process ./ssm-setup-cli.exe -ArgumentList @("-register", "-
activation-code=$code", "-activation-id=$id", "-region=$region") -Wait
Get-Content ($env:ProgramData + "\Amazon\SSM\InstanceData\registration")
Get-Service -Name "AmazonSSMAgent"
```

Gerenciar dispositivos de borda com o Systems Manager

Esta seção descreve as tarefas de configuração que os administradores de sistema e da conta executam para habilitar a configuração e o gerenciamento dos dispositivos principais do AWS IoT Greengrass. Depois de concluir essas tarefas, os usuários que receberam permissões pelo administrador da Conta da AWS poderão usar o AWS Systems Manager para configurar e gerenciar os dispositivos principais do AWS IoT Greengrass na organização.

> **Note**
>
> - O SSM Agent para AWS IoT Greengrass não é compatível com macOS e Windows 10. Não é possível usar ferramentas do Systems Manager para gerenciar e configurar dispositivos de borda que usam esses sistemas operacionais.
> - O Systems Manager também oferece suporte a dispositivos de borda que não estejam configurados como dispositivos principais do AWS IoT Greengrass. Para usar o Systems Manager para gerenciar os dispositivos principais do AWS IoT e dispositivos de borda que não são da AWS, você deve configurá-los usando uma ativação para ambientes híbridos. Para obter mais informações, consulte Gerenciar nós em ambientes híbridos e multinuvem com o Systems Manager.

- Para usar o Session Manager e patches em aplicações da Microsoft com seus dispositivos de borda, você deve habilitar o nível de instâncias avançadas. Para obter mais informações, consulte Ativar o nível de instâncias avançadas.

Antes de começar

Verifique se os dispositivos de borda atendem aos requisitos a seguir.

- Seus dispositivos de borda devem atender aos requisitos para serem configurados como dispositivos principais do AWS IoT Greengrass. Para obter mais informações, consulte Configurar dispositivos principais do AWS IoT Greengrass no Guia do desenvolvedor do AWS IoT Greengrass Version 2.
- Seus dispositivos de borda devem ser compatíveis com o Agente do AWS Systems Manager (SSM Agent). Para obter mais informações, consulte Sistemas operacionais compatíveis com o Systems Manager.
- Seus dispositivos de borda devem conseguir se comunicar com o serviço do Systems Manager na nuvem. O Systems Manager não oferece suporte a dispositivos de borda desconectados.

Sobre a configuração de dispositivos de borda

Configurar dispositivos AWS IoT Greengrass para o Systems Manager envolve os processos a seguir.

> **Note**
>
> Para obter informações sobre a desinstalação do SSM Agent de um dispositivo de borda, consulte Desinstalar o agente do AWS Systems Manager no Guia do desenvolvedor do AWS IoT Greengrass Version 2.

Criar um perfil de serviço do IAM para seus dispositivos de borda

Os dispositivos principais do AWS IoT Greengrass requerem uma função de serviço do AWS Identity and Access Management (IAM) para se comunicar com o AWS Systems Manager. A atribuição de função do AWS Security Token Service (AWS STS) AssumeRole confiança para o serviço Systems Manager. Você só precisa criar a função de serviço uma vez para cada Conta da AWS. Você especificará essa função para o parâmetro `RegistrationRole` quando configurar e implantar o

componente SSM Agent dos dispositivos AWS IoT Greengrass. Caso já tenha criado esse perfil ao configurar nós que não são do EC2 para um ambiente híbrido e multinuvem, você poderá ignorar esta etapa.

> **Note**
>
> Usuários na sua empresa ou organização que usarão o Systems Manager em seus dispositivos de borda deverão receber permissão no IAM para chamar a API do Systems Manager.

Requisito de política do bucket do S3

Se qualquer um dos seguintes casos for verdadeiro, crie uma política de permissões personalizada do IAM e para os buckets do Amazon Simple Storage Service (Amazon S3) antes de concluir este procedimento:

- Caso 1: você está usando um endpoint da VPC para conectar de forma privada sua VPC aos Serviços da AWS compatíveis e aos serviços do endpoint da VPC com a tecnologia AWS PrivateLink.
- Caso 2: você planeja usar um bucket do S3 criado como parte das operações do Systems Manager, por exemplo, para armazenar a saída para comandos do Run Command ou as sessões do Session Manager em um bucket do S3. Antes de continuar, siga as etapas em Create a custom S3 bucket policy for an instance profile (Criar uma política personalizada de bucket do S3 para um perfil de instância). As informações sobre políticas de bucket do S3 neste tópico também se aplicam à sua função de serviço.

  > **Note**
  >
  > Se seus dispositivos estiverem protegidos por um firewall e você planeja usar o Patch Manager, o firewall deve permitir o acesso ao endpoint da lista de referência de patches `arn:aws:s3:::patch-baseline-snapshot-`*`região`*`/*`.
  > A *`região`* representa o identificador da região para uma região da Região da AWS compatível com o AWS Systems Manager, como `us-east-2` para a região Leste dos EUA (Ohio). Para ver uma lista dos valores de *`região`* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

AWS CLI

Para criar uma função de serviço do IAM para um ambiente AWS IoT Greengrass (AWS CLI)

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Em sua máquina local, crie um arquivo de texto com um nome como `SSMService-Trust.json` com a política de confiança a seguir. Certifique-se de salvar o arquivo com a extensão de arquivo `.json`.

   > **Note**
   >
   > Anote o nome. Você o especificará ao implantar o SSM Agent para seus dispositivos principais do AWS IoT Greengrass.

   JSON

   ```
   {
       "Version": "2012-10-17",
       "Statement": {
           "Effect": "Allow",
           "Principal": {
               "Service": "ssm.amazonaws.com"
           },
           "Action": "sts:AssumeRole"
       }
   }
   ```

3. Abra a AWS CLI e, no diretório em que você criou o arquivo JSON, execute o create-role para criar a função de serviço. Substitua cada *`espaço reservado para recurso de exemplo`* por suas próprias informações.

   Linux & macOS

   ```
   aws iam create-role \
       --role-name SSMServiceRole \
       --assume-role-policy-document file://SSMService-Trust.json
   ```

Windows

```
aws iam create-role ^
    --role-name SSMServiceRole ^
    --assume-role-policy-document file://SSMService-Trust.json
```

4. Execute o comando attach-role-policy da maneira a seguir para permitir que a função de serviço recém-criada crie um token de sessão. O token de sessão concede aos dispositivos de borda permissão para executar comandos usando o Systems Manager.

> **Note**
>
> As políticas que você adicionar a um perfil de serviço para dispositivos de borda são as mesmas políticas usadas para criar um perfil da instância para instâncias do Amazon Elastic Compute Cloud (Amazon EC2). Para obter mais informações sobre as políticas do IAM usadas nos comandos a seguir, consulte Configurar permissões de instância necessárias para o Systems Manager.

(Obrigatório) Execute o comando a seguir para permitir que um dispositivo de borda use a funcionalidade básica do serviço do AWS Systems Manager.

Linux & macOS

```
aws iam attach-role-policy \
    --role-name SSMServiceRole \
    --policy-arn arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
```

Windows

```
aws iam attach-role-policy ^
    --role-name SSMServiceRole ^
    --policy-arn arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
```

Se você tiver criado uma política de bucket do S3 personalizada para sua função de serviço, execute o comando a seguir para permitir que o AWS Systems Manager Agent (SSM Agent) acesse os buckets que você especificou na política. Substitua *account_ID* e *my_bucket_policy_name* pelo ID da Conta da AWS e o nome do bucket.

Linux & macOS

```
aws iam attach-role-policy \
    --role-name SSMServiceRole \
    --policy-arn arn:aws:iam::account_ID:policy/my_bucket_policy_name
```

Windows

```
aws iam attach-role-policy ^
    --role-name SSMServiceRole ^
    --policy-arn arn:aws:iam::account_id:policy/my_bucket_policy_name
```

(Opcional) Execute o comando a seguir para permitir que o SSM Agent acesse o AWS Directory Service em seu nome para que as solicitações ingressem no domínio dos dispositivos de borda. A função de serviço precisará dessa política somente se você integrar os dispositivos de borda a um diretório do Microsoft AD.

Linux & macOS

```
aws iam attach-role-policy \
    --role-name SSMServiceRole \
    --policy-arn arn:aws:iam::aws:policy/AmazonSSMDirectoryServiceAccess
```

Windows

```
aws iam attach-role-policy ^
    --role-name SSMServiceRole ^
    --policy-arn arn:aws:iam::aws:policy/AmazonSSMDirectoryServiceAccess
```

(Opcional) Execute o comando a seguir para permitir que o agente do CloudWatch seja executado nos dispositivos de borda. Esse comando permite ler informações em um dispositivo e gravá-las no CloudWatch. Sua função de serviço precisará dessa política somente se você pretende usar serviços como o Amazon EventBridge ou Amazon CloudWatch Logs.

```
aws iam attach-role-policy \
    --role-name SSMServiceRole \
    --policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
```

Tools for PowerShell

Para criar uma função de serviço do IAM para um ambiente AWS IoT Greengrass (AWS Tools for Windows PowerShell)

1. Instale e configure o Ferramentas da AWS para PowerShell (Ferramentas para Windows PowerShell), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar o Ferramentas da AWS para PowerShell.

2. Em sua máquina local, crie um arquivo de texto com um nome como `SSMService-Trust.json` com a política de confiança a seguir. Certifique-se de salvar o arquivo com a extensão de arquivo `.json`.

   > **Note**
   >
   > Anote o nome. Você o especificará ao implantar o SSM Agent para seus dispositivos principais do AWS IoT Greengrass.

   JSON

   ```
   {
       "Version": "2012-10-17",
       "Statement": {
           "Effect": "Allow",
           "Principal": {
               "Service": "ssm.amazonaws.com"
           },
           "Action": "sts:AssumeRole"
       }
   }
   ```

3. Abra o PowerShell no modo administrativo e no diretório em que você criou o arquivo JSON, execute o New-IAMRole da seguinte maneira para criar uma função de serviço.

   ```
   New-IAMRole `
       -RoleName SSMServiceRole `
       -AssumeRolePolicyDocument (Get-Content -raw SSMService-Trust.json)
   ```

4. Use Register-IAMRolePolicy conforme a seguir para permitir a função de serviço que você criou para criar um token de sessão. O token de sessão concede aos dispositivos de borda permissão para executar comandos usando o Systems Manager.

> **Note**
>
> As políticas que você adicionar em uma função de serviço para dispositivos de borda em um ambiente AWS IoT Greengrass são as mesmas políticas usadas para criar um perfil para instâncias do EC2. Para obter mais informações sobre as políticas da AWS usadas nos comandos a seguir, consulte Configurar permissões de instância necessárias para o Systems Manager.

(Obrigatório) Execute o comando a seguir para permitir que um dispositivo de borda use a funcionalidade básica do serviço do AWS Systems Manager.

```
Register-IAMRolePolicy `
    -RoleName SSMServiceRole `
    -PolicyArn arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
```

Se você tiver criado uma política de bucket do S3 personalizada para sua função de serviço, execute o comando a seguir para permitir que o SSM Agent acesse os buckets que você especificou na política. Substitua *account_ID* e *my_bucket_policy_name* pelo ID da Conta da AWS e o nome do bucket.

```
Register-IAMRolePolicy `
    -RoleName SSMServiceRole `
    -PolicyArn arn:aws:iam::account_ID:policy/my_bucket_policy_name
```

(Opcional) Execute o comando a seguir para permitir que o SSM Agent acesse o AWS Directory Service em seu nome para que as solicitações ingressem no domínio dos dispositivos de borda. A função de serviço precisará dessa política somente se você integrar os dispositivos de borda a um diretório do Microsoft AD.

```
Register-IAMRolePolicy `
    -RoleName SSMServiceRole `
    -PolicyArn arn:aws:iam::aws:policy/AmazonSSMDirectoryServiceAccess
```

(Opcional) Execute o comando a seguir para permitir que o agente do CloudWatch seja executado nos dispositivos de borda. Esse comando permite ler informações em um dispositivo e gravá-las no CloudWatch. Sua função de serviço precisará dessa política somente se você pretende usar serviços como o Amazon EventBridge ou Amazon CloudWatch Logs.

```
Register-IAMRolePolicy `
    -RoleName SSMServiceRole `
    -PolicyArn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
```

Configure seus dispositivos de borda para AWS IoT Greengrass

Configure seus dispositivos de borda como dispositivos principais do AWS IoT Greengrass. O processo de configuração envolve a verificação de sistemas operacionais e requisitos do sistema compatíveis, bem como a instalação e configuração do software principal do AWS IoT Greengrass em seus dispositivos. Para obter mais informações, consulte Configurar dispositivos principais do AWS IoT Greengrass no Guia do desenvolvedor do AWS IoT Greengrass Version 2.

Atualizar o perfil de troca de token do AWS IoT Greengrass e instalar o SSM Agent em seus dispositivos de borda

A etapa final para configurar e configurar seus dispositivos principais AWS IoT Greengrass para o Systems Manager exige que você atualize a função de serviço do dispositivo AWS IoT Greengrass AWS Identity and Access Management (IAM), também chamado de função de troca de token e implante o AWS Systems Manager Agent (Agente) (SSM Agent) para os seus dispositivos AWS IoT Greengrass. Para obter informações sobre esses processos, consulte Install the AWS Systems Manager Agent no Guia do desenvolvedor do AWS IoT Greengrass Version 2.

Depois de implantar o SSM Agent em seus dispositivos, o AWS IoT Greengrass registra automaticamente os dispositivos com o Systems Manager. Nenhum registro adicional é necessário. Você pode começar a usar as ferramentas do Systems Manager para acessar, gerenciar e configurar seus dispositivos do AWS IoT Greengrass.

> **Note**
>
> Seus dispositivos de borda devem conseguir se comunicar com o serviço do Systems Manager na nuvem. O Systems Manager não oferece suporte a dispositivos de borda desconectados.

Criar um administrador delegado do AWS Organizations para o Systems Manager

Ao configurar uma organização no AWS Organizations, você atribui uma conta de gerenciamento para realizar todas as tarefas administrativas para todos os Serviços da AWS. O usuário da conta de gerenciamento pode atribuir uma conta de administrador delegado somente para que o Systems Manager execute tarefas administrativas para o Change Manager, o Explorer e o OpsCenter. O AWS Organizations é um serviço de gerenciamento de contas que pode ser usado para criar uma organização e atribuir Contas da AWS para gerenciar essas contas de maneira centralizada. Para obter informações sobre AWS Organizations, consulte AWS Organizations no Guia do usuário do AWS Organizations.

O Change Manager, o Explorer e o OpsCenter, todos ferramentas do AWS Systems Manager, trabalham com o AWS Organizations para realizar tarefas em todas as contas de membro da organização. É possível atribuir somente um administrador delegado a todas as ferramentas do Systems Manager. A conta de administrador delegado deve ser um membro da organização à qual ela está atribuída.

Tópicos

- Usar um administrador delegado com o Change Manager
- Usar um administrador delegado com o Explorer
- Usar um administrador delegado com o OpsCenter
- Usar um administrador delegado com o Quick Setup

Usar um administrador delegado com o Change Manager

O Change Manager é um framework empresarial de gerenciamento de alterações para solicitar, aprovar, implementar e emitir relatórios sobre alterações operacionais em sua configuração e infraestrutura de aplicações.

Se você usa o Change Manager em uma organização, atribua uma conta de administrador delegado para gerenciar modelos, aprovações e relatórios de alteração para todas as contas de membro. Com a Quick Setup, você pode configurar o Change Manager para usar com uma organização e selecionar a conta de administrador delegado. Se você usar o Change Manager apenas com uma única Conta da AWS, a conta de administrador delegado não será necessária.

Por padrão, o Change Manager exibe todas as tarefas relacionadas a alterações na conta do administrador delegado. Para obter instruções sobre como configurar um administrador delegado ao configurar o Change Manager para uma organização, consulte Configure o Change Manager para uma organização (conta de gerenciamento).

> **Important**
>
> Se você usar o Change Manager em uma organização, recomendamos sempre fazer as alterações na conta de administrador delegado. Embora seja possível fazer alterações de outras contas na organização, essas alterações não serão relatadas ou visíveis na conta do administrador delegado.

Usar um administrador delegado com o Explorer

O Explorer é um painel de operações personalizável que exibe uma visualização agregada dos dados de operações (OpsData) para as Contas da AWS em todas as Regiões da AWS.

Você pode configurar uma conta de administrador delegado para o Systems Manager agregar dados do Explorer de várias regiões e contas usando a sincronização de dados de recursos com o AWS Organizations. O administrador delegado pode pesquisar, filtrar e agregar dados do Explorer usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) ou o AWS Tools for Windows PowerShell.

Ao usar uma conta de administrador delegado para o Explorer, você limita o número de administradores que podem criar ou excluir sincronizações de dados de recurso de várias contas e regiões a apenas uma Conta da AWS individual.

Você pode sincronizar dados de operações em todas as Contas da AWS da organização usando o Explorer. Para obter informações sobre como atribuir um administrador delegado pelo Explorer, consulte Configuração de um administrador delegado para Explorer.

Usar um administrador delegado com o OpsCenter

O OpsCenter fornece um local central em que engenheiros de operações e profissionais de TI podem gerenciar itens de trabalho operacionais (OpsItems) relacionados a recursos da AWS. Se você quiser usar o OpsCenter para gerenciar o OpsItems de maneira centralizada entre contas, é necessário configurar a organização no AWS Organizations.

Com a Quick Setup para o OpsCenter, você pode atribuir uma conta de administrador delegado e configurar o OpsCenter para gerenciar OpsItems de maneira centralizada. Para obter mais informações, consulte (Opcional) Configurar o OpsCenter para gerenciar OpsItems entre contas usando a Quick Setup.

Usar um administrador delegado com o Quick Setup

O Quick Setup é uma ferramenta do Systems Manager que ajuda configurar rapidamente serviços e recursos da AWS usados frequentemente de acordo com as práticas recomendadas. É possível configurar uma conta de administrador delegado da Quick Setup para implantar e gerenciar configurações em contas e regiões usando o AWS Organizations. Um administrador delegado da Quick Setup pode criar, atualizar, visualizar e excluir recursos do gerenciador de configurações em sua organização. O Systems Manager registra um administrador delegado para a Quick Setup como parte do processo de configuração da experiência do console integrado. Para obter mais informações, consulte Configurar o console unificado do Systems Manager para uma organização.

Configuração geral para AWS Systems Manager

Se você ainda não fez isso, conecte–se em uma Conta da AWS e crie um usuário administrativo.

Inscrever-se para uma Conta da AWS

Se você ainda não tem uma Conta da AWS, siga as etapas abaixo para criar uma.

Como cadastrar uma Conta da AWS

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga as instruções online.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

Quando você se inscreve para uma Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

A AWS envia um e-mail de confirmação depois que o processo de inscrição é concluído. A qualquer momento, é possível exibir as atividades da conta atual e gerenciar sua conta acessando https://aws.amazon.com/ e selecionando Minha conta.

Criar um usuário com acesso administrativo

Depois de se cadastrar em uma Conta da AWS, proteja seu Usuário raiz da conta da AWS, habilite o AWS IAM Identity Center e crie um usuário administrativo para não usar o usuário-raiz em tarefas cotidianas.

Proteger o Usuário raiz da conta da AWS

1. Faça login no AWS Management Console como o proprietário da conta ao escolher a opção Usuário-raiz e inserir o endereço de e-mail da Conta da AWS. Na próxima página, insira a senha.

   Para obter ajuda ao fazer login usando o usuário-raiz, consulte Fazer login como usuário-raiz no Guia do usuário do Início de Sessão da AWS.
2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

   Para obter instruções, consulte Habilitar um dispositivo MFA virtual para sua Conta da AWS de usuário-raiz (console) no Guia do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

   Para obter instruções, consulte Habilitar o AWS IAM Identity Center no Guia do usuário do AWS IAM Identity Center.
2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

Para obter um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como a fonte de identidade, consulte Configurar o acesso dos usuários com o Diretório do Centro de Identidade do IAM padrão no Guia do usuário do AWS IAM Identity Center.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

  Para obter ajuda para fazer login usando um usuário do Centro de Identidade do IAM, consulte Fazer login no portal de acesso da AWS, no Guia do usuário do Início de Sessão da AWS.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

   Para obter instruções, consulte Criar um conjunto de permissões no Guia do usuário do AWS IAM Identity Center.
2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

   Para obter instruções, consulte Adicionar grupos no Guia do usuário do AWS IAM Identity Center.

Configurar o AWS Systems Manager

Os tópicos a seguir descrevem como configurar o console do AWS Systems Manager unificado para organizações do AWS Organizations e Contas da AWS individuais.

Tópicos

- Configurar o acesso ao console do Systems Manager
- Configurar o console unificado do Systems Manager para uma organização
- Configurar o console unificado do Systems Manager para uma única conta e região
- Desabilitar o console unificado do Systems Manager

Configurar o acesso ao console do Systems Manager

Para usar o AWS Systems Manager no AWS Management Console, é necessário ter as permissões corretas configuradas.

Para obter mais informações sobre como criar políticas do AWS Identity and Access Management e anexá-las a identidades do IAM, consulte Criar políticas do IAM no Guia do usuário do IAM

Política de integração do Systems Manager

Você pode criar uma política do IAM como a mostrada no exemplo a seguir e anexar a política às suas identidades do IAM. Esta política concede acesso total à integração e à configuração do Systems Manager.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `ssm-quicksetup`: permite que as entidades principais acessem todas as ações do Configuração rápida do AWS Systems Manager.
- `ssm`: permite que as entidades principais acessem o Systems Manager Automation e o Explorador de recursos.
- `organizations`: permite que as entidades principais leiam a estrutura de uma organização no AWS Organizations e gerenciem os administradores delegados ao se integrarem ao Systems Manager como uma organização.
- `cloudformation`: permite que as entidades principais gerenciem suas pilhas da Quick Setup.

- `iam`: permite que as entidades principais gerenciem os perfis e políticas do IAM necessários para a integração do Systems Manager.

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
     {
       "Sid": "QuickSetupActions",
       "Effect": "Allow",
       "Action": [
         "ssm-quicksetup:*"
       ],
       "Resource": "*"
     },

     {
       "Sid": "SsmReadOnly",
       "Effect": "Allow",
       "Action": [
         "ssm:DescribeAutomationExecutions",
         "ssm:GetAutomationExecution",
         "ssm:ListAssociations",
         "ssm:DescribeAssociation",
         "ssm:ListDocuments",
         "ssm:ListResourceDataSync",
         "ssm:DescribePatchBaselines",
         "ssm:GetPatchBaseline",
         "ssm:DescribeMaintenanceWindows",
         "ssm:DescribeMaintenanceWindowTasks"
       ],
       "Resource": "*"
     },
     {
       "Sid": "SsmDocument",
       "Effect": "Allow",
       "Action": [
         "ssm:GetDocument",
         "ssm:DescribeDocument"
       ],
       "Resource": [
```

```
        "arn:aws:ssm:*:*:document/AWSQuickSetupType-*",
        "arn:aws:ssm:*:*:document/AWS-EnableExplorer"
      ]
    },
    {
      "Sid": "SsmEnableExplorer",
      "Effect": "Allow",
      "Action": "ssm:StartAutomationExecution",
      "Resource": "arn:aws:ssm:*:*:automation-definition/AWS-EnableExplorer:*"
    },
    {
      "Sid": "SsmExplorerRds",
      "Effect": "Allow",
      "Action": [
        "ssm:GetOpsSummary",
        "ssm:CreateResourceDataSync",
        "ssm:UpdateResourceDataSync"
      ],
      "Resource": "arn:aws:ssm:*:*:resource-data-sync/AWS-QuickSetup-*"
    },
    {
      "Sid": "OrgsReadOnly",
      "Effect": "Allow",
      "Action": [
        "organizations:DescribeAccount",
        "organizations:DescribeOrganization",
        "organizations:ListDelegatedAdministrators",
        "organizations:ListRoots",
        "organizations:ListParents",
        "organizations:ListOrganizationalUnitsForParent",
        "organizations:DescribeOrganizationalUnit",
        "organizations:ListAWSServiceAccessForOrganization"
      ],
      "Resource": "*"
    },
    {
      "Sid": "OrgsAdministration",
      "Effect": "Allow",
      "Action": [
        "organizations:EnableAWSServiceAccess",
        "organizations:RegisterDelegatedAdministrator",
        "organizations:DeregisterDelegatedAdministrator"
      ],
      "Resource": "*",
```

```
      "Condition": {
        "StringEquals": {
          "organizations:ServicePrincipal": [
            "ssm.amazonaws.com",
            "ssm-quicksetup.amazonaws.com",
            "member.org.stacksets.cloudformation.amazonaws.com",
            "resource-explorer-2.amazonaws.com"
          ]
        }
      }
    },
    {
      "Sid": "CfnReadOnly",
      "Effect": "Allow",
      "Action": [
        "cloudformation:ListStacks",
        "cloudformation:DescribeStacks",
        "cloudformation:ListStackSets",
        "cloudformation:DescribeOrganizationsAccess"
      ],
      "Resource": "*"
    },
    {
      "Sid": "OrgCfnAccess",
      "Effect": "Allow",
      "Action": [
        "cloudformation:ActivateOrganizationsAccess"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CfnStackActions",
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateStack",
        "cloudformation:DeleteStack",
        "cloudformation:DescribeStackResources",
        "cloudformation:DescribeStackEvents",
        "cloudformation:GetTemplate",
        "cloudformation:RollbackStack",
        "cloudformation:TagResource",
        "cloudformation:UntagResource",
        "cloudformation:UpdateStack"
      ],
```

```
        "Resource": [
          "arn:aws:cloudformation:*:*:stack/StackSet-AWS-QuickSetup-*",
          "arn:aws:cloudformation:*:*:stack/AWS-QuickSetup-*",
          "arn:aws:cloudformation:*:*:type/resource/*"
        ]
      },
      {
        "Sid": "CfnStackSetActions",
        "Effect": "Allow",
        "Action": [
          "cloudformation:CreateStackInstances",
          "cloudformation:CreateStackSet",
          "cloudformation:DeleteStackInstances",
          "cloudformation:DeleteStackSet",
          "cloudformation:DescribeStackInstance",
          "cloudformation:DetectStackSetDrift",
          "cloudformation:ListStackInstanceResourceDrifts",
          "cloudformation:DescribeStackSet",
          "cloudformation:DescribeStackSetOperation",
          "cloudformation:ListStackInstances",
          "cloudformation:ListStackSetOperations",
          "cloudformation:ListStackSetOperationResults",
          "cloudformation:TagResource",
          "cloudformation:UntagResource",
          "cloudformation:UpdateStackSet"
        ],
        "Resource": [
          "arn:aws:cloudformation:*:*:stackset/AWS-QuickSetup-*",
          "arn:aws:cloudformation:*:*:type/resource/*",
          "arn:aws:cloudformation:*:*:stackset-target/AWS-QuickSetup-*:*"
        ]
      },
      {
        "Sid": "ValidationReadonlyActions",
        "Effect": "Allow",
        "Action": [
          "iam:ListRoles",
          "iam:GetRole"
        ],
        "Resource": "*"
      },
      {
        "Sid": "IamRolesMgmt",
        "Effect": "Allow",
```

```
        "Action": [
          "iam:CreateRole",
          "iam:DeleteRole",
          "iam:GetRole",
          "iam:AttachRolePolicy",
          "iam:DetachRolePolicy",
          "iam:GetRolePolicy",
          "iam:ListRolePolicies"
        ],
        "Resource": [
          "arn:aws:iam::*:role/AWS-QuickSetup-*",
          "arn:aws:iam::*:role/service-role/AWS-QuickSetup-*"
        ]
      },
      {
        "Sid": "IamPassRole",
        "Effect": "Allow",
        "Action": [
          "iam:PassRole"
        ],
        "Resource": [
          "arn:aws:iam::*:role/AWS-QuickSetup-*",
          "arn:aws:iam::*:role/service-role/AWS-QuickSetup-*"
        ],
        "Condition": {
          "StringEquals": {
            "iam:PassedToService": [
              "ssm.amazonaws.com",
              "ssm-quicksetup.amazonaws.com",
              "cloudformation.amazonaws.com"
            ]
          }
        }
      },
      {
        "Sid": "IamRolesPoliciesMgmt",
        "Effect": "Allow",
        "Action": [
          "iam:AttachRolePolicy",
          "iam:DetachRolePolicy"
        ],
        "Resource": [
          "arn:aws:iam::*:role/AWS-QuickSetup-*",
          "arn:aws:iam::*:role/service-role/AWS-QuickSetup-*"
```

```
      ],
      "Condition": {
        "ArnEquals": {
          "iam:PolicyARN": [
            "arn:aws:iam::aws:policy/
AWSSystemsManagerEnableExplorerExecutionPolicy",
            "arn:aws:iam::aws:policy/AWSQuickSetupSSMDeploymentRolePolicy"
          ]
        }
      }
    },
    {
      "Sid": "CfnStackSetsSLR",
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/aws-service-role/
stacksets.cloudformation.amazonaws.com/
AWSServiceRoleForCloudFormationStackSetsOrgAdmin",
        "arn:aws:iam::*:role/aws-service-role/ssm.amazonaws.com/
AWSServiceRoleForAmazonSSM",
        "arn:aws:iam::*:role/aws-service-role/
accountdiscovery.ssm.amazonaws.com/AWSServiceRoleForAmazonSSM_AccountDiscovery",
        "arn:aws:iam::*:role/aws-service-role/ssm-quicksetup.amazonaws.com/
AWSServiceRoleForSSMQuickSetup",
        "arn:aws:iam::*:role/aws-service-role/resource-explorer-2.amazonaws.com/
AWSServiceRoleForResourceExplorer"
      ]
    }
  ]
}
```

Política do operador do console do AWS Systems Manager

Você pode criar uma política do IAM como a mostrada no exemplo a seguir e anexar a política às suas identidades do IAM. Essa política concede acesso total para operar o Systems Manager e permite que o Systems Manager execute documentos do Automation para diagnóstico e correção.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `ssm`: permite que as entidades principais acessem todas as APIs do Systems Manager.
- `ssm-quicksetup`: permite que as entidades principais gerenciem as próprias configurações da Quick Setup.
- `ec2`: permite que o Systems Manager determine sua Regiões da AWS habilitada e o status da instância do Amazon EC2.
- `cloudformation`: permite que as entidades principais leiam as próprias pilhas da Quick Setup.
- `organizations`: permite que as entidades principais leiam a estrutura de uma organização no AWS Organizations e gerenciem os administradores delegados ao se integrarem ao Systems Manager como uma organização.
- `s3`: permite que as entidades principais listem e obtenham objetos em um bucket do Amazon S3 para diagnóstico, o qual é criado durante o processo de integração do Systems Manager.
- `iam:PassRole`: permite que as entidades principais passem perfis a serem assumidos para o Systems Manager quando eles executam automações para diagnóstico e correção de nós não gerenciados.
- `iam:GetRole`: permite que as entidades principais obtenham informações específicas de perfis do Quick Setup ao operar com o Systems Manager.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:*",
        "ssm-quicksetup:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowEC2DescribeActions",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstanceStatus",
```

```
        "ec2:DescribeInstances",
        "ec2:DescribeRegions"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CfnAccess",
      "Effect": "Allow",
      "Action": [
        "cloudformation:ListStacks",
        "cloudformation:ListStackSets",
        "cloudformation:ListStackInstances",
        "cloudformation:ListStackSetOperations",
        "cloudformation:ListStackSetOperationResults",
        "cloudformation:DescribeStacks",
        "cloudformation:DescribeStackSet",
        "cloudformation:DescribeStackSetOperation",
        "cloudformation:DescribeOrganizationsAccess",
        "cloudformation:DescribeStackInstance",
        "cloudformation:DetectStackSetDrift",
        "cloudformation:ListStackInstanceResourceDrifts"
      ],
      "Resource": "*"
    },
    {
      "Sid": "OrgsReadOnly",
      "Effect": "Allow",
      "Action": [
        "organizations:DescribeAccount",
        "organizations:DescribeOrganization",
        "organizations:ListDelegatedAdministrators",
        "organizations:ListRoots",
        "organizations:ListParents",
        "organizations:ListOrganizationalUnitsForParent",
        "organizations:DescribeOrganizationalUnit",
        "organizations:ListAWSServiceAccessForOrganization"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowKMSOperations",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
```

```
        "kms:GenerateDataKey"
      ],
      "Resource": "arn:aws:kms:*:*:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/SystemsManagerManaged": "true"
        },
        "ArnLike": {
          "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::do-not-delete-ssm-
diagnosis-*"
        },
        "StringLike": {
          "kms:ViaService": "s3.*.amazonaws.com"
        },
        "Bool": {
          "aws:ViaAWSService": "true"
        }
      }
    },
    {
      "Sid": "AllowReadS3BucketFromOrganization",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:::do-not-delete-ssm-diagnosis*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceOrgId": "${aws:PrincipalOrgId}"
        }
      }
    },
    {
      "Sid": "AllowReadS3BucketFromSingleAccount",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:::do-not-delete-ssm-diagnosis*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
```

```
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": [
        "arn:aws:iam::*:role/AWS-SSM-DiagnosisAdminRole*",
        "arn:aws:iam::*:role/AWS-SSM-DiagnosisExecutionRole*",
        "arn:aws:iam::*:role/AWS-SSM-RemediationAdminRole*",
        "arn:aws:iam::*:role/AWS-SSM-RemediationExecutionRole*"
      ],
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "ssm.amazonaws.com"
        }
      }
    },
    {
      "Sid": "IamReadOnly",
      "Effect": "Allow",
      "Action": "iam:GetRole",
      "Resource": [
        "arn:aws:iam::*:role/AWS-QuickSetup-*",
        "arn:aws:iam::*:role/service-role/AWS-QuickSetup-*"
      ]
    }
  ]
}
```

Política somente leitura do operador do console do AWS Systems Manager

Você pode criar uma política do IAM como a mostrada no exemplo a seguir e anexar a política às suas identidades do IAM. Essa política concede acesso somente leitura ao Systems Manager.

- `ssm`: permite que as entidades principais acessem as APIs somente leitura do Systems Manager.
- `ssm-quicksetup`: permite que as entidades principais leiam as próprias configurações da Quick Setup.
- `cloudformation`: permite que as entidades principais leiam as próprias pilhas da Quick Setup.

- `iam:GetRole`: permite que as entidades principais obtenham informações de perfis específicas para perfis do Quick Setup quando utilizam o Systems Manager.
- `ec2:DescribeRegions`: permite que o Systems Manager determine suas Regiões da AWS habilitadas.
- `organizations`: permite que as entidades principais leiam a estrutura de uma organização no AWS Organizations ao se integrarem ao Systems Manager como uma organização.
- `s3`: permite que as entidades principais listem e obtenham objetos em um bucket do Amazon S3 que é criado durante o processo de integração do Systems Manager.

Detalhes das permissões

Esta política inclui as seguintes permissões.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:Describe*",
        "ssm:Get*",
        "ssm:List*",
        "ssm-quicksetup:List*",
        "ssm-quicksetup:Get*",
        "cloudformation:Describe*",
        "cloudformation:Get*",
        "cloudformation:List*",
        "iam:GetRole",
        "ec2:DescribeRegions",
        "organizations:Describe*",
        "organizations:List*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowKMSOperations",
      "Effect": "Allow",
      "Action": [
```

```
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:*:*:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/SystemsManagerManaged": "true"
        },
        "ArnLike": {
          "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::do-not-delete-ssm-
diagnosis-*"
        },
        "StringLike": {
          "kms:ViaService": "s3.*.amazonaws.com"
        },
        "Bool": {
          "aws:ViaAWSService": "true"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:::do-not-delete-ssm-diagnosis*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceOrgId": "${aws:PrincipalOrgId}"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:::do-not-delete-ssm-diagnosis*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
```

```
    }
  ]
}
```

Configurar o console unificado do Systems Manager para uma organização

O processo de configuração da experiência do console unificado do Systems Manager é concluído via AWS Management Console com apenas alguns cliques. Para configurar o Systems Manager para uma organização do AWS Organizations, é necessário ter acesso à conta de gerenciamento da sua organização e a outra conta na sua organização para usar como administrador delegado. O acesso à conta de gerenciamento só é necessário para habilitar ou desabilitar o Systems Manager. Para gerenciar seus nós, você usará a conta de administrador delegado.

Pré-requisitos

Ao gerenciar nós em uma organização, o Systems Manager usa vários serviços dependentes para configurar e aprimorar a funcionalidade do console unificado. Como resultado, o Systems Manager deve habilitar o acesso confiável e registrar uma conta de administrador delegado para os seguintes serviços:

- AWS CloudFormation: implanta os recursos necessários para o funcionamento do Systems Manager em suas contas.
- Explorador de recursos da AWS: pesquisa e filtragem de instâncias do EC2 em suas contas.
- AWS Systems Manager Explorer: monitoramento e solução de problemas da integridade dos recursos implantados pelo Systems Manager em suas contas.
- AWS Systems Manager Quick Setup: implanta as configurações da Quick Setup necessários para o funcionamento do Systems Manager em suas contas.

Antes de começar, verifique se você já não ultrapassou a cota de administradores delegados para algum desses serviços dependentes. Caso contrário, não será possível registrar as contas de administrador delegado necessárias para habilitar o Systems Manager. Quando você habilita o Systems Manager para uma organização, todas as contas em sua organização são incluídas. No momento, não há previsão para a exclusão de contas do processo de configuração. Ao habilitar o Systems Manager, você pode escolher as Regiões da AWS que deseja incluir. Somente as

regiões que atualmente oferecem suporte ao console unificado do Systems Manager podem ser selecionadas. Para saber mais sobre as regiões em que a experiência do console está disponível, consulte Regiões da AWS com suporte.

Recursos do console unificado

O processo de configuração do console do Systems Manager conclui muitas tarefas de pré-requisitos para você. Dependendo dos recursos que você escolher configurar, isso inclui habilitar a Configuração de gerenciamento de host padrão para fornecer as permissões necessárias do IAM aos seus nós, e muito mais. A lista a seguir detalha os recursos criados pelo Systems Manager para o console unificado. Dependendo dos recursos que você escolher configurar, alguns recursos podem não ser criados.

Visualizações gerenciadas pela Explorador de recursos da AWS

- `AWSManagedViewForSSM`: permite que o Systems Manager acesse informações de recursos indexadas pelo Explorador de Recursos para sua organização. Essas visualizações gerenciadas só podem ser atualizadas ou excluídas pelo Systems Manager. Isso significa que, se você quiser excluir as visualizações gerenciadas ou desativar o Explorador de Recursos, será necessário desativar o console unificado. Para obter mais informações sobre como desabilitar o console unificado, consulte Desabilitar o console unificado do Systems Manager. Para obter mais informações sobre visualizações gerenciadas, consulte Visualizações gerenciadas da AWS no Guia do usuário do Explorador de Recursos.

> **Note**
>
> Se você criou um índice agregador para o Explorador de Recursos em uma região diferente da sua região de origem, o Systems Manager rebaixa o índice atual. Em seguida, o Systems Manager promove o índice local em sua região de origem como o novo índice agregador. Durante esse período, somente os nós da sua região de origem são exibidos. A conclusão desse processo poderá demorar até 24 horas.

Perfis do IAM

- `RoleForOnboardingAutomation`: permite que o Systems Manager gerencie recursos durante o processo de configuração. Para obter mais informações sobre a política, consulte AWSQuickSetupSSMManageResourcesExecutionPolicy.

- `RoleForLifecycleManagement`: permite que o Lambda gerencie o ciclo de vida dos recursos criados pelo processo de configuração. Para obter mais informações sobre a política, consulte AWSQuickSetupSSMLifecycleManagementExecutionPolicy.
- `RoleForAutomation`: um perfil de serviço para o Systems Manager Automation assumir para executar runbooks. Para obter mais informações, consulte Criar perfis de serviço para o Automation usando o console.
- `AWSSSMDiagnosisAdminRole`: um perfil administrativo usado para iniciar automações que usam runbooks de diagnóstico. Para obter mais informações sobre as políticas, consulte AWS-SSM-DiagnosisAutomation-AdministrationRolePolicy, AWS-SSM-Automation-DiagnosisBucketPolicy e AWS-SSM-DiagnosisAutomation-OperationalAccountAdministrationRolePolicy.
- `AWSSSMDiagnosisExecutionRole`: um perfil de execução de automação para o runbook de diagnóstico. Para obter mais informações sobre as políticas, consulte AWS-SSM-DiagnosisAutomation-ExecutionRolePolicy e AWS-SSM-Automation-DiagnosisBucketPolicy.
- `AWSSSMRemediationAdminRole`: um perfil administrativo usado para iniciar automações que usam runbooks de correção. Para obter mais informações sobre as políticas, consulte AWS-SSM-RemediationAutomation-AdministrationRolePolicy, AWS-SSM-Automation-DiagnosisBucketPolicy e AWS-SSM-RemediationAutomation-OperationalAccountAdministrationRolePolicy.
- `AWSSSMRemediationExecutionRole`: um perfil de execução de automação para o runbook de correção. Para obter mais informações sobre as políticas, consulte AWS-SSM-RemediationAutomation-ExecutionRolePolicy e AWS-SSM-Automation-DiagnosisBucketPolicy.
- `ManagedInstanceCrossAccountManagementRole`: permite que o Systems Manager colete informações de nós gerenciados em todas as contas.

Associações do State Manager

- `EnableDHMCAssociation`: executa diariamente e garante que a Configuração de gerenciamento de host padrão esteja habilitada.
- `SystemAssociationForEnablingExplorer`: executa diariamente e garante que o Explorer esteja habilitado. O Explorer é usado para sincronizar dados dos seus nós gerenciados.
- `EnableAREXAssociation`: executa diariamente e garante que o Explorador de recursos da AWS esteja habilitado. O Resource Explorer é usado para determinar quais instâncias do Amazon EC2 em sua organização não são gerenciadas pelo Systems Manager.
- `SSMAgentUpdateAssociation`: executa a cada 14 dias e garante que a versão mais recente disponível do SSM Agent esteja instalada em seus nós gerenciados.

- `SystemAssociationForInventoryCollection`: executa a cada 12 horas e coleta dados de inventário dos seus nós gerenciados.

Buckets do S3

- `DiagnosisBucket`: armazena dados coletados da execução do runbook de diagnóstico.

Funções do Lambda

- `SSMLifecycleOperatorLambda`: permite que as entidades principais acessem todas as ações do Configuração rápida do AWS Systems Manager.
- `SSMLifecycleResource`: recurso personalizado para ajudar a gerenciar o ciclo de vida dos recursos criados pelo processo de configuração.

Além disso, após a conclusão do processo de configuração, você pode selecionar a tarefa Diagnosticar e corrigir nó para aplicar automaticamente as correções aos nós que não estão se identificando como gerenciados pelo Systems Manager. Isso pode incluir a identificação de problemas como falha de conectividade de rede com os endpoints do Systems Manager e muito mais. Para obter mais informações, consulte Diagnosticar e remediar.

Configurar o console unificado

Para configurar o console unificado do Systems Manager para uma organização

1. Faça login na conta de gerenciamento da sua organização.
2. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
3. Insira o ID da conta que deseja registrar como administrador delegado.
4. Depois que a conta de administrador delegado for registrada com sucesso, faça login nessa conta e retorne ao console do Systems Manager para concluir a configuração do Systems Manager.
5. Selecione Habilitar Systems Manager.
6. Na seção Região de origem, determine uma região na qual deseja que o Systems Manager agregue seus dados de nós. Por padrão, o Systems Manager seleciona a região que você está usando atualmente. Para escolher uma região de origem diferente, altere o console para a região que você deseja usar antes de configurar o Systems Manager. Os dados dos nós são replicados entre as contas e regiões da sua organização e armazenados na região de origem.

A região escolhida não poderá ser alterada depois que a configuração do Systems Manager for configurada. Para usar uma região diferente como a região de origem da sua organização, é necessário desabilitar o console unificado e concluir o processo de configuração novamente. Se sua organização usa o Centro de Identidade do IAM, é necessário selecionar a mesma região em que configurou o Centro de Identidade do IAM como região de origem.

7. Na seção Regiões, selecione as regiões nas quais você deseja habilitar o Systems Manager.
8. Na seção Configurações de recursos, escolha as opções que você deseja habilitar para a sua configuração:

   Habilitar a Configuração de gerenciamento de host padrão (DHMC)

   Permite que o Systems Manager configure a DHMC. Esse recurso permite que o Systems Manager use um perfil do IAM para garantir que todas as instâncias do Amazon EC2 na conta e na região tenham as permissões necessárias para serem gerenciadas pelo Systems Manager. Você também pode especificar a frequência da remediação de desvios. O desvio de configuração ocorre sempre que um usuário faz qualquer alteração em um serviço ou recurso que entre em conflito com as seleções feitas por meio da sua configuração. O Systems Manager verifica desvios na configuração e tenta remediá-los com base na frequência especificada. Você deve especificar um valor entre 1 e 31 dias. Se você já configurou a DHMC em uma região, o Systems Manager não vai alterar o perfil do IAM selecionado anteriormente. Para obter mais informações sobre a DHMC, consulte Gerenciar instâncias do EC2 automaticamente com a Configuração de gerenciamento de hosts padrão.

   A DHMC torna possível gerenciar instâncias do Amazon EC2 sem a necessidade de criar um perfil de instância do AWS Identity and Access Management (IAM) manualmente. Recomendamos escolher esta opção para garantir que as instâncias do EC2 tenham as permissões necessárias para serem gerenciadas pelo Systems Manager.

   Habilitar a coleta de metadados de inventário

   Habilita o Systems Manager para configurar a coleta dos seguintes tipos de metadados dos nós:

   - Componentes da AWS: driver do EC2, agentes, versões e muito mais.
   - Aplicações: nomes de aplicações, editores, versões e muito mais.
   - Detalhes de nós: nome do sistema, nome do sistema operacional (SO), versão do SO, última inicialização, DNS, domínio, grupo de trabalho, arquitetura do SO e muito mais.

- Configuração de rede:: endereço IP, endereço MAC, DNS, gateway, máscara de sub-rede e muito mais.
- Serviços: nome, nome de exibição, status, serviços dependentes, tipo de serviço, tipo de início e muito mais (somente instâncias do Windows Server).
- Funções do Windows: nome, nome de exibição, caminho, tipo de recurso, estado instalado e muito mais (somente nós do Windows Server).
- Atualizações do Windows: ID do hotfix, autor da instalação, data da instalação e muito mais (somente nós do Windows Server).

Especifique a frequência com que o inventário é coletado. Você deve especificar um valor entre 1 e 744 horas. Para obter mais informações sobre o Inventory, uma ferramenta do AWS Systems Manager, consulte AWS Systems Manager Inventory.

Habilitar as atualizações automáticas do agente do Systems Manager (SSM)

Habilita o Systems Manager para que verifique se há uma nova versão do agente na frequência que você especificar. O valor da frequência deve estar entre 1 e 31 dias. Se houver uma nova versão, o Systems Manager atualizará automaticamente o agente em seu nó gerenciado para a última versão liberada. O Systems Manager não instala o agente em instâncias em que ele ainda não esteja presente. Para obter informações sobre quais AMIs estão SSM Agent pré-instaladas, consulte Encontrar AMIs com o SSM Agent pré-instalado.

Recomendamos escolher essa opção para garantir que os nós estejam sempre executando a versão mais atualizada do SSM Agent. Para obter mais informações sobre o SSM Agent, incluindo informações sobre como instalar o agente manualmente, consulte Trabalhar com o SSM Agent.

9. Selecione Enviar.

Dependendo do tamanho da sua organização, a configuração da experiência do console unificado do Systems Manager pode ser demorada.

Configurar o console unificado do Systems Manager para uma única conta e região

Para configurar a experiência do console unificado do Systems Manager para uma Conta da AWS e uma Região da AWS, não é necessário usar o Organizations nem registrar uma conta de administrador delegado. O processo de configuração da experiência do console do Systems

Manager conclui muitas tarefas de pré-requisitos para você. Dependendo dos recursos que você escolher configurar, isso inclui habilitar a Configuração de gerenciamento de host padrão para fornecer as permissões necessárias do IAM aos seus nós, e muito mais. A lista a seguir detalha os recursos criados pelo Systems Manager para o console unificado.

Recursos do console unificado

Dependendo dos recursos que você escolher configurar, alguns recursos podem não ser criados.

Perfis do IAM

- `RoleForOnboardingAutomation`: permite que o Systems Manager gerencie recursos durante o processo de configuração. Para obter mais informações sobre a política, consulte AWSQuickSetupSSMManageResourcesExecutionPolicy.
- `RoleForLifecycleManagement`: permite que o Lambda gerencie o ciclo de vida dos recursos criados pelo processo de configuração. Para obter mais informações sobre a política, consulte AWSQuickSetupSSMLifecycleManagementExecutionPolicy.
- `RoleForAutomation`: um perfil de serviço para o Systems Manager Automation assumir para executar runbooks. Para obter mais informações, consulte Criar perfis de serviço para o Automation usando o console.
- `AWSSSMDiagnosisAdminRole`: um perfil de execução de automação para o runbook de diagnóstico. Para obter mais informações sobre as políticas, consulte AWS-SSM-DiagnosisAutomation-AdministrationRolePolicy, AWS-SSM-Automation-DiagnosisBucketPolicy e AWS-SSM-DiagnosisAutomation-OperationalAccountAdministrationRolePolicy.
- `AWSSSMRemediationAdminRole`: um perfil de execução de automação para o runbook de correção. Para obter mais informações sobre as políticas, consulte AWS-SSM-RemediationAutomation-AdministrationRolePolicy, AWS-SSM-Automation-DiagnosisBucketPolicy e AWS-SSM-RemediationAutomation-OperationalAccountAdministrationRolePolicy.
- `ManagedInstanceCrossAccountManagementRole`: permite que o Systems Manager colete informações de nós gerenciados em todas as contas.

Associações do State Manager

- `EnableDHMCAssociation`: executa diariamente e garante que a Configuração de gerenciamento de host padrão esteja habilitada.
- `SystemAssociationForEnablingExplorer`: executa diariamente e garante que o Explorer esteja habilitado. O Explorer é usado para sincronizar dados dos seus nós gerenciados.

- `EnableAREXAssociation`: executa diariamente e garante que o Explorador de recursos da AWS esteja habilitado. O Resource Explorer é usado para determinar quais instâncias do Amazon EC2 em sua organização não são gerenciadas pelo Systems Manager.
- `SSMAgentUpdateAssociation`: executa a cada 14 dias e garante que a versão mais recente disponível do SSM Agent esteja instalada em seus nós gerenciados.
- `SystemAssociationForInventoryCollection`: executa a cada 12 horas e coleta dados de inventário dos seus nós gerenciados.

Buckets do S3

- `DiagnosisBucket`: armazena dados coletados da execução do runbook de diagnóstico.

Funções do Lambda

- `SSMLifecycleOperatorLambda`: permite que as entidades principais acessem todas as ações do Configuração rápida do AWS Systems Manager.
- `SSMLifecycleResource`: recurso personalizado para ajudar a gerenciar o ciclo de vida dos recursos criados pelo processo de configuração.

Além disso, após a conclusão do processo de configuração, você pode selecionar a tarefa Diagnosticar e corrigir nó para aplicar automaticamente as correções aos nós que não estão se identificando como gerenciados pelo Systems Manager. Isso pode incluir a identificação de problemas como falha de conectividade de rede com os endpoints do Systems Manager e muito mais.

Configurar o console unificado

Para configurar o Systems Manager para uma única conta e região

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. Selecione Habilitar Systems Manager.
3. Na seção Configurações de recursos, escolha as opções que você deseja habilitar para a sua configuração:

Habilitar a Configuração de gerenciamento de host padrão (DHMC)

Permite que o Systems Manager configure a DHMC. Esse recurso permite que o Systems Manager use um perfil do IAM para garantir que todas as instâncias do Amazon EC2 na conta e na região tenham as permissões necessárias para serem gerenciadas pelo Systems Manager. Você também pode especificar a frequência da remediação de desvios. O desvio de configuração ocorre sempre que um usuário faz qualquer alteração em um serviço ou recurso que entre em conflito com as seleções feitas por meio da sua configuração. O Systems Manager verifica desvios na configuração e tenta remediá-los com base na frequência especificada. Você deve especificar um valor entre 1 e 31 dias. Se você já configurou a DHMC em uma região, o Systems Manager não vai alterar o perfil do IAM selecionado anteriormente. Para obter mais informações sobre a DHMC, consulte Gerenciar instâncias do EC2 automaticamente com a Configuração de gerenciamento de hosts padrão.

A DHMC torna possível gerenciar instâncias do Amazon EC2 sem a necessidade de criar um perfil de instância do AWS Identity and Access Management (IAM) manualmente. Recomendamos escolher esta opção para garantir que as instâncias do EC2 tenham as permissões necessárias para serem gerenciadas pelo Systems Manager.

Habilitar a coleta de metadados de inventário

Habilita o Systems Manager para configurar a coleta dos seguintes tipos de metadados dos nós:

- Componentes da AWS: driver do EC2, agentes, versões e muito mais.
- Aplicações: nomes de aplicações, editores, versões e muito mais.
- Detalhes de nós: nome do sistema, nome do sistema operacional (SO), versão do SO, última inicialização, DNS, domínio, grupo de trabalho, arquitetura do SO e muito mais.
- Configuração de rede:: endereço IP, endereço MAC, DNS, gateway, máscara de sub-rede e muito mais.
- Serviços: nome, nome de exibição, status, serviços dependentes, tipo de serviço, tipo de início e muito mais (somente instâncias do Windows Server).
- Funções do Windows: nome, nome de exibição, caminho, tipo de recurso, estado instalado e muito mais (somente nós do Windows Server).
- Atualizações do Windows: ID do hotfix, autor da instalação, data da instalação e muito mais (somente nós do Windows Server).

Especifique a frequência com que o inventário é coletado. Você deve especificar um valor entre 1 e 744 horas. Para obter mais informações sobre o Inventory, uma ferramenta do AWS Systems Manager, consulte AWS Systems Manager Inventory.

Habilitar as atualizações automáticas do agente do Systems Manager (SSM)

Habilita o Systems Manager para que verifique se há uma nova versão do agente na frequência que você especificar. O valor da frequência deve estar entre 1 e 31 dias. Se houver uma nova versão, o Systems Manager atualizará automaticamente o agente em seu nó gerenciado para a última versão liberada. O Systems Manager não instala o agente em instâncias em que ele ainda não esteja presente. Para obter informações sobre quais AMIs estão SSM Agent pré-instaladas, consulte Encontrar AMIs com o SSM Agent pré-instalado.

Recomendamos escolher essa opção para garantir que os nós estejam sempre executando a versão mais atualizada do SSM Agent. Para obter mais informações sobre o SSM Agent, incluindo informações sobre como instalar o agente manualmente, consulte Trabalhar com o SSM Agent.

4. Selecione Enviar.

Desabilitar o console unificado do Systems Manager

Para desabilitar o console unificado do Systems Manager para uma organização, será necessário acessar a conta que você registrou como administrador delegado do Systems Manager. Após fazer login na conta de administrador delegado da sua organização, você poderá desativar a configuração da sua organização na seção Configurações do console unificado. Quando você desabilita a configuração do console unificado para sua organização, o Systems Manager exclui os recursos criados, inclusive as visualizações gerenciadas do Explorador de Recursos, durante o processo de configuração. Desativar a configuração da sua organização não revoga o acesso confiável nem cancela o registro das contas de administrador delegado para serviços dependentes. O procedimento a seguir descreve como desabilitar a configuração para o console unificado.

Para desabilitar a configuração para o console unificado do Systems Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. Selecione Configurações no painel de navegação.

3. Selecione Desabilitar. Você deve confirmar que deseja desabilitar a configuração da sua organização. Essa ação exclui os recursos criados para o console unificado e não pode ser desfeita.

Se você não conseguir acessar a conta de administrador delegado do Systems Manager e quiser desabilitar a configuração do console unificado, também poderá fazer isso na conta de gerenciamento da sua organização. Usando a AWS CLI ou o SDK, chame a operação da API `DeleteConfigurationManager` e passe o valor `ManagerArn` para a configuração da organização em sua conta. O formato do ARN do gerenciador usado para configurar o console unificado é o seguinte:

`arn:aws:ssm-quicksetup:`*`account-id`*`:configuration-manager/`*`configuration-manager-id`*.

> **Note**
>
> Se você não souber o valor de *`configuration-manager-id`*, chame a ação da API `ListConfigurationManagers` e filtre os resultados usando o tipo `AWSQuickSetupType-SSM`.

Executar tarefas de gerenciamento de nós com o AWS Systems Manager

Os tópicos a seguir descrevem como concluir tarefas em nós comuns usando o console unificado do AWS Systems Manager integrado para uma organização do AWS Organizations e Contas da AWS individuais.

Tópicos

- Como revisar insights de nós
- Explorando nós
- Acesso a nós just-in-time usando o Systems Manager
- Diagnosticar e remediar
- Ajustar as configurações do Systems Manager

Como revisar insights de nós

É possível obter informações sobre o status geral dos nós gerenciados e das instâncias do EC2 não gerenciadas em sua organização ou conta usando o console unificado do Systems Manager.

O Systems Manager fornece uma visão geral visual dos nós gerenciados e das instâncias do EC2 que ainda não são gerenciadas pelo Systems Manager. (Um nó gerenciado é qualquer máquina configurada para uso com o Systems Manager em ambientes híbridos e multinuvem. Para obter informações sobre os tipos de máquinas compatíveis, consulte Tipos de máquinas compatíveis em ambientes híbridos e multinuvem.)

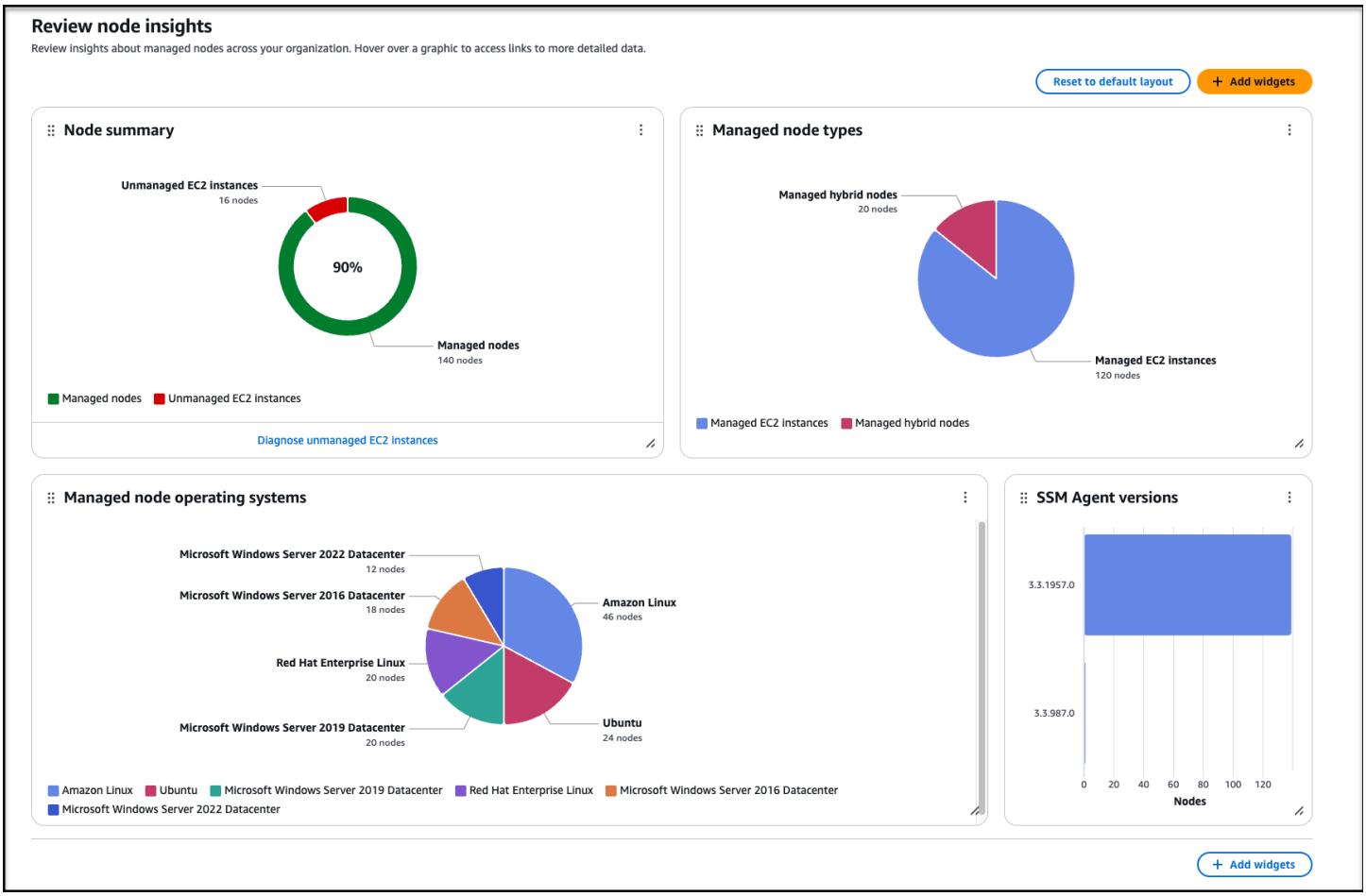
Essa visão geral é fornecida por meio de caixas de relatórios individuais, chamadas de widgets, que apresentam gráficos de pizza interativos e outros tipos de gráficos.

Antes de começar

Para revisar insights do nó, primeiro é necessário integrar sua organização ou conta ao console unificado do Systems Manager. Para obter mais informações, consulte Configurar o AWS Systems Manager.

Após a integração, abra o console do Systems Manager e escolha Revisar insights do nó.

A imagem a seguir mostra as caixas de relatório individuais, chamadas widgets, que estão disponíveis na página Revisar insigths do nó.



A exibição é compatível com widgets que fornecem a você as informações a seguir.

Resumo dos nós

Indica quantas instâncias do EC2 em sua organização ou conta não são nós gerenciados atualmente e quantos nós gerenciados fazem parte da frota da sua organização ou conta.

O que é uma instância não gerenciada?

Quando você interrompe uma instância EC2 gerenciada, ela é relatada como "Não gerenciada" no console do Systems Manager. Esse é o comportamento esperado porque o SSM Agent não tem uma conexão ativa com o serviço.

> **Note**
>
> Isso é diferente de como o AWS Config define uma instância como não gerenciada. Se uma instância estiver parada no momento, o AWS Config informa qual foi o status da instância na última vez em que uma conexão "heartbeat" foi feita entre o SSM Agent na instância e o serviço Systems Manager.

Quando a instância é reiniciada, ela se reconecta automaticamente ao serviço Systems Manager e seu status no console unificado é restaurado para "Gerenciada" em cinco minutos. Nenhuma intervenção manual é necessária e todas as configurações do Systems Manager para a instância são preservadas durante o ciclo de Parada/Início.

No entanto, se a instância ainda não for relatada como "Gerenciada" vários minutos após o início, é provável que a instância não esteja configurada adequadamente para o gerenciamento do Systems Manager. Nesse caso, recomendamos executar um diagnóstico para identificar por que a instância permanece em um estado não gerenciado. Para obter mais informações, consulte Diagnosticar e corrigir instâncias não gerenciadas do Amazon EC2 no Systems Manager.

Se a verificação de diagnóstico não conseguir determinar o problema, consulte os tópicos a seguir para verificar se os requisitos para o SSM Agent, perfis (IAM) AWS Identity and Access Management e os pré-requisitos do Systems Manager foram todos atendidos:

- Solução de problemas do SSM Agent
- Configurar permissões de instância obrigatórias para o Systems Manager
- Solução de problemas de disponibilidade do nó gerenciado

Tipos de nós gerenciados

Indica quantos nós gerenciados na sua frota são instâncias do EC2 e quantos são outros tipos de servidores, incluindo servidores em suas próprias instalações (servidores on-premises), dispositivos AWS IoT Greengrass básicos, dispositivos de borda AWS IoT e não AWS e máquinas virtuais (VMs), incluindo VMs em outros ambientes de nuvem. Mova o ponteiro do mouse sobre o gráfico Tipos de nós para acessar links com mais detalhes na página Explorar nós.

Para obter informações sobre suporte da AWS a ambientes híbridos e multinuvem, consulte Soluções da AWS para nuvem híbrida e multinuvem.

Versões do SSM Agent

Fornece informações sobre as instalações do AWS Systems Manager Agent (SSM Agent) em sua frota. O SSM Agent é um software da Amazon que é executado em seus nós gerenciados. O SSM Agent permite que o Systems Manager atualize, gerencie e configure esses recursos. O agente processa as solicitações do serviço Systems Manager na Nuvem AWS e as executa conforme especificado na solicitação.

Para nós gerenciados em sua frota, esse widget relata as versões do SSM Agent em sua frota, da mais nova para a mais antiga. Mova o ponteiro do mouse sobre o gráfico Versões do SSM Agent para acessar links com mais detalhes na página Explorar nós.

Para obter mais informações sobre o SSM Agent, consulte Trabalhar com o SSM Agent.

Sistemas operacionais de nós gerenciados

Fornece um detalhamento da porcentagem de cada sistema operacional nos nós gerenciados da sua frota. Mova o ponteiro do mouse sobre o gráfico Nós gerenciados por sistemas operacionais para acessar links com mais detalhes na página Explorar nós.

É possível personalizar o layout dos widgets na página Revisar insights do nó usando um recurso de arrastar e soltar e removendo e adicionando widgets à tela.

Use as informações dos tópicos a seguir para obter ajuda para trabalhar com os widgets de insights do Systems Manager.

Tópicos

- Adicionar ou remover widgets da página Revisar insights do nó
- Reorganizar widgets na página Revisar insights do nó

Adicionar ou remover widgets da página Revisar insights do nó

É possível personalizar o layout da página Revisar insights do nó do Systems Manager adicionando e removendo widgets.

> **Note**
>
> Por padrão, a página exibe todos os widgets disponíveis.

Para adicionar ou remover widgets da página Revisar insights do nó

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. Na navegação à esquerda, selecione Revisar insights do nó.
3. Para remover um widget da exibição, faça o seguinte:
   a. Escolha o menu Mais opções () para o widget.
   b. Selecione Remover widget.
4. Para adicionar um widget à exibição, faça o seguinte:
   a. Escolha Adicionar widgets.
   b. No painel Adicionar widgets, clique e segure a alça de arrastar () do widget para adicionar à exibição.
   c. Arraste o widget e solte-o no painel principal.

Reorganizar widgets na página Revisar insights do nó

É possível personalizar o layout na página Revisar insights do nó reorganizando os widgets.

Para reorganizar os widgets na página Revisar insights do nó

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Revisar insights do nó.
3. Para personalizar o layout de widgets, escolha um widget que gostaria de mover. Clique e mantenha pressionado a alça de arrastar () do widget e arraste-o até o novo local.
4. Repita este processo para cada widget que você deseja reposicionar.

Se você decidir que não gosta do novo layout, escolha Redefinir para layout padrão para mover todos os widgets de volta para as posições originais.

Explorando nós

Você pode usar a página Explorar nós no Systems Manager para revisar os detalhes dos nós gerenciados em sua organização ou conta de acordo com os critérios especificados nos filtros. Você também pode usar a integração do Systems Manager com o Amazon Q Developer (Amazon Q), uma solução de IA generativa da AWS, para pesquisar usando prompts de texto.

Antes de começar

Para usar o recurso Explorar nós, primeiro é necessário integrar sua organização ou conta ao console unificado do Systems Manager. Para ter mais informações, consulte Configurar o console unificado do Systems Manager para uma organização.

Após a integração, abra o Console do Systems Manager e escolha Explorar nós.

> **Note**
>
> Se você criou um índice agregador para o Explorador de Recursos em uma região diferente da sua região de origem, o Systems Manager rebaixa o índice atual. Em seguida, o Systems Manager promove o índice local em sua região de origem como o novo índice agregador. Durante esse período, somente os nós da sua região de origem são exibidos. A conclusão desse processo poderá demorar até 24 horas.

Tópicos

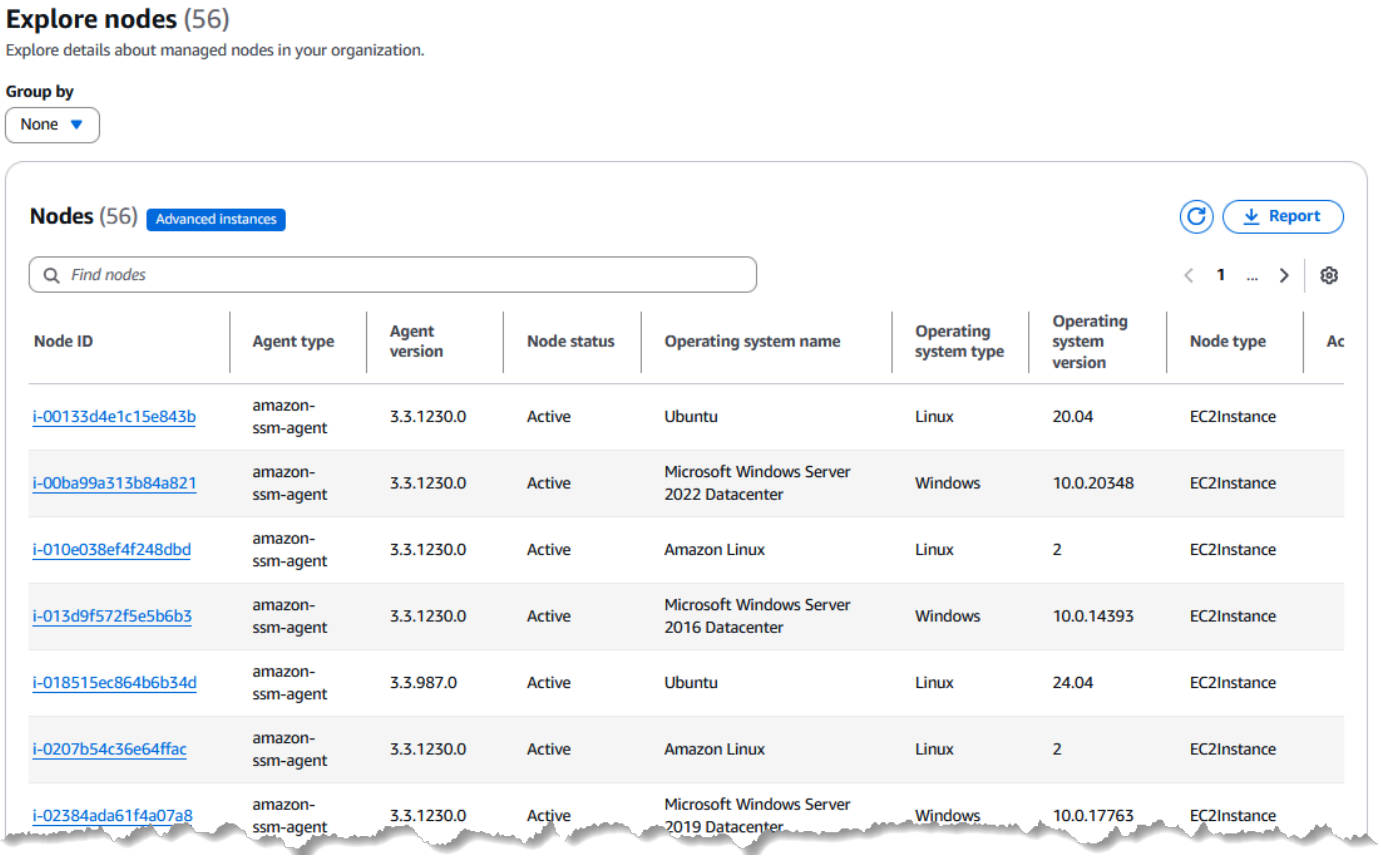
- Explorar nós usando filtros de console
- Explorar nós usando prompts de texto no Amazon Q
- Visualizar detalhes individuais do nó e realizando ações em um nó
- Baixar ou exportar um relatório de nó gerenciado
- Gerenciar o conteúdo e a aparência de relatórios de nós

Explorar nós usando filtros de console

No console do Systems Manager, é possível agrupar seus nós gerenciados de acordo com as seguintes visualizações:

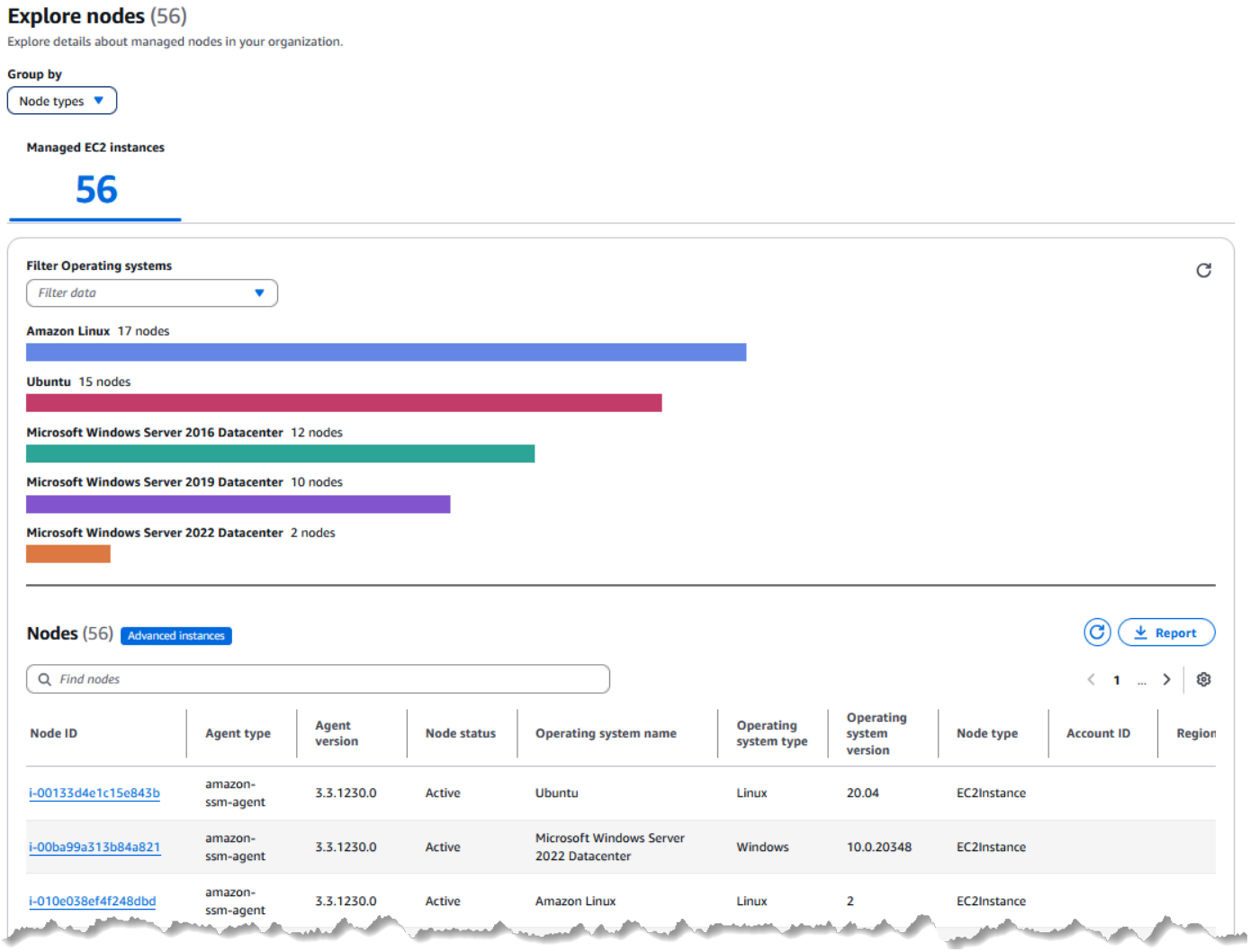
All nodes (No filter)

Lista de todos os nós gerenciados em sua organização ou conta.



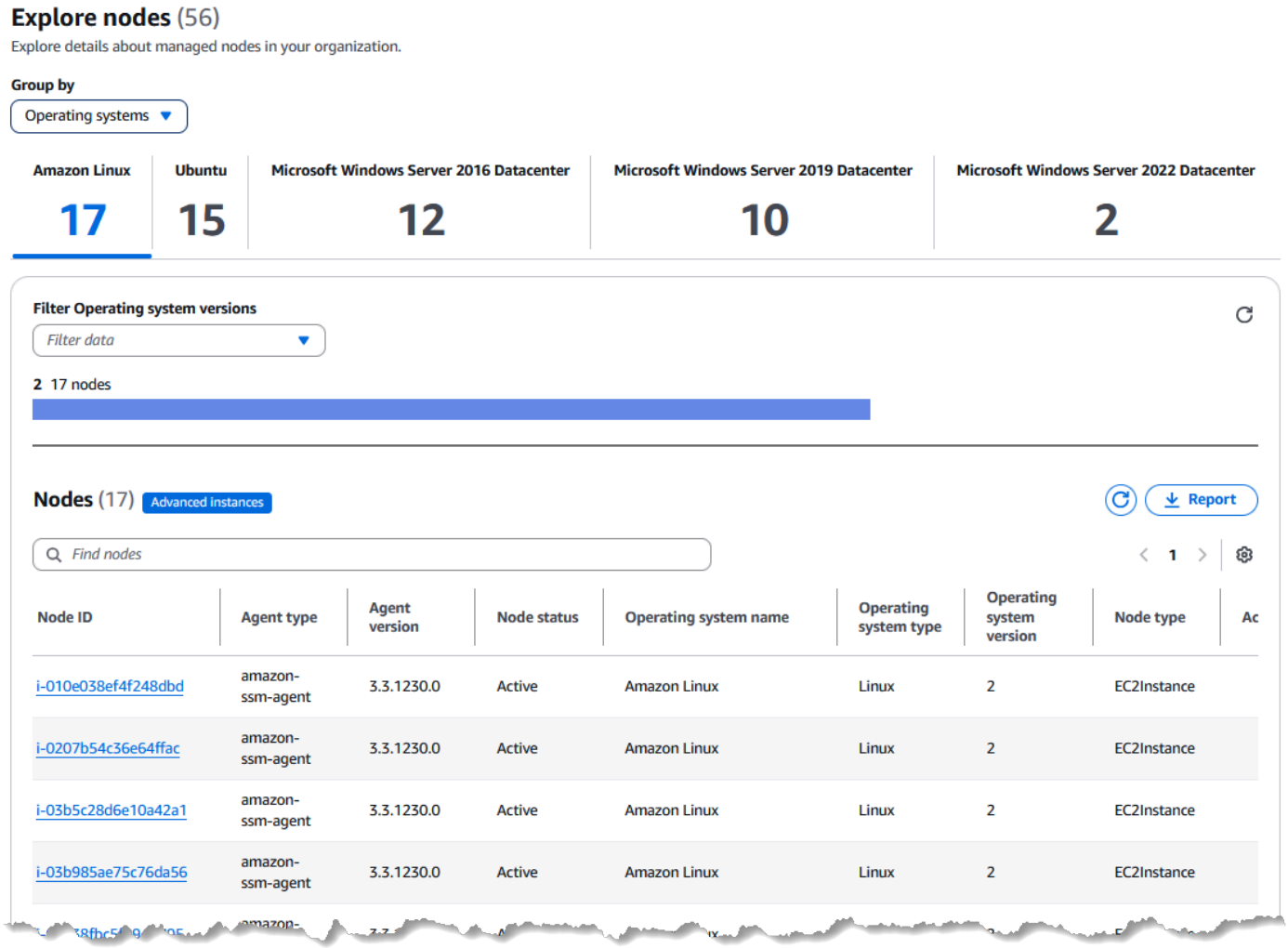
Node types

Fornece guias para visualização de dados separadamente para instâncias do Amazon Elastic Compute Cloud (Amazon EC2) e outros tipos de máquina, incluindo servidores em suas próprias dependências (servidores on-premises), dispositivos AWS IoT Greengrass básicos, dispositivos de borda AWS IoT e não AWS e máquinas virtuais (VMs), incluindo VMs em outros ambientes de nuvem.



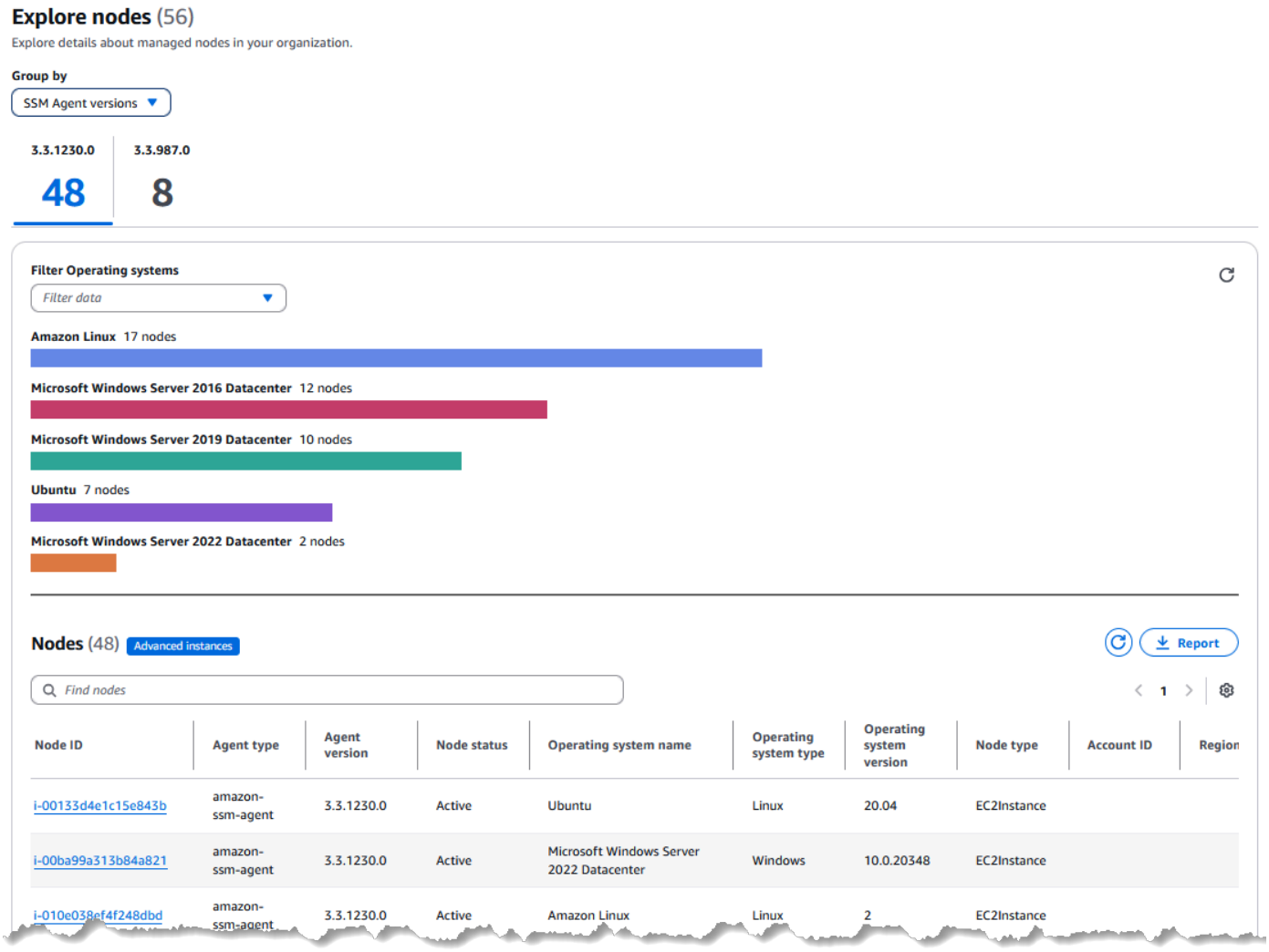
Operating systems

Fornece uma guia para cada tipo de sistema operacional em sua organização ou conta, como Amazon Linux e Microsoft Windows Server 2022 Datacenter. Em cada guia, é possível filtrar ainda mais a lista selecionando somente versões específicas dos sistemas operacionais, como Amazon Linux 2 e Amazon Linux 2023.



SSM Agent versions

Fornece uma guia para cada versão do SSM Agent instalada nos nós gerenciados da sua frota. Em cada guia, é possível filtrar ainda mais a lista selecionando somente sistemas operacionais específicos, como Amazon Linux e Microsoft Windows Server 2022 Datacenter.



Além disso, para cada uma dessas visualizações, você pode refinar ainda mais a lista de nós relatados optando por visualizar somente os nós de uma determinada propriedade, como status do nó, ID do Conta da AWS, ID da unidade organizacional e mais.

É possível personalizar a exibição do relatório escolhendo quais das colunas de dados disponíveis serão exibidas na página Explorar nós. Você também pode baixar relatórios nos formatos `CSV` ou `JSON` ou exportar relatórios para o Amazon S3 no formato `CSV`.

Tópicos

- Escolher uma visualização de filtro para resumos de nós gerenciados

Escolher uma visualização de filtro para resumos de nós gerenciados

A página Explorar nós no Systems Manager permite visualizar dados agregados sobre sua frota de acordo com várias visualizações de filtro disponíveis.

Para escolher uma visualização de filtro para resumos de nós gerenciados

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Explorar nós.
3. Em Visualização de filtro, selecione uma das opções de filtro e, opcionalmente, refine ainda mais o relatório:

   - Nós gerenciados: na caixa de pesquisa (), é possível selecionar uma propriedade e um delimitador, como `Node type = Managed EC2 instances`.
   - Sistemas operacionais: na lista Filtrar versões do sistema operacional, é possível selecionar um número de versão do sistema operacional. Na caixa de pesquisa (), é possível selecionar uma propriedade e um delimitador, como `Node type = Managed EC2 instances`.
   - Versões do SSM Agent: na lista Filtrar sistemas operacionais, é possível selecionar um nome de sistema operacional. Na caixa de pesquisa (), é possível selecionar uma propriedade e um delimitador, como `Node type = Managed EC2 instances`.
   - Tipos de nós: na lista Filtrar sistemas operacionais, é possível selecionar um nome de sistema operacional. Na caixa de pesquisa (), é possível selecionar uma propriedade e um delimitador, como `Node type = Managed EC2 instances`.

Depois de filtrar a lista em caráter opcional, é possível visualizar detalhes sobre um nó gerenciado específico escolhendo seu ID na coluna ID do nó. A partir dessa visão detalhada, você pode realizar várias ações no nó.

Explorar nós usando prompts de texto no Amazon Q

Usando a integração do Systems Manager com o Amazon Q Developer, você pode usar solicitações de texto para visualizar informações criadas por IA generativa sobre seus nós gerenciados.

O Amazon Q Developer é um assistente conversacional baseado em IA generativa que pode ajudar você a entender, criar, estender e operar aplicações da AWS. Para acelerar o desenvolvimento na AWS, o modelo que impulsiona o Amazon Q foi aprimorado com conteúdo de alta qualidade da AWS para fornecer respostas mais completas, acionáveis e referenciadas. Para ter mais informações, consulte What is Amazon Q Developer no Guia do usuário do Amazon Q Developer.

A integração entre o Systems Manager e o Amazon Q permite que você obtenha rapidamente visibilidade e controle sobre ambientes grandes e distribuídos em várias regiões e Contas da AWS. É possível usar consultas em linguagem natural para pesquisar rapidamente dados de nós, identificar problemas e agir com mais rapidez.

Quando você faz uma pergunta em linguagem natural sobre nós gerenciados ou instâncias gerenciadas, o Amazon Q usa a ação `ListNodes` do Systems Manager e cria filtros com base na sua entrada de texto para recuperar os resultados.

Por exemplo, digamos que você forneça o seguinte prompt à Amazon Q:

`List my managed nodes running Red Hat Enterprise Linux 9.2`

O Amazon Q determina quais filtros incluir em uma solicitação e, em seguida, executa uma consulta semelhante à seguinte:

```
aws ssm list-nodes \
    --filters Key=PlatformName,Values='Red Hat Enterprise Linux',Type=Equal
 Key=PlatformVersion,Values=9.2,Type=Equal
```

Em seguida, o Amazon Q gera um relatório sobre instâncias do Red Hat Enterprise Linux em sua conta, listando informações como o número de instâncias, seus IDs e suas regiões.

Você também pode ver um resumo JSON dos detalhes de cada instância, bem como abrir um link para ver as listas completas de instâncias do EC2 ou nós gerenciados na página Explorar nós do Systems Manager. Explorar nós exibe resultados que correspondem aos critérios de filtro que você incluiu no prompt. A partir daí, você pode alterar ou refinar os filtros da sua solicitação, conforme descrito em Explorando nós.

Tópicos

- Como criar prompts eficazes para perguntar ao Amazon Q sobre sua frota
- Explorar nós gerenciados com o Amazon Q

Como criar prompts eficazes para perguntar ao Amazon Q sobre sua frota

Quanto melhor a qualidade da pergunta, ou prompt, que você fornecer ao Amazon Q, melhor será o resultado que ela fornecerá.

Dicas para solicitações de consulta

Lembre-se das dicas a seguir ao consultar o Amazon Q sobre sua frota:

1. Para ajudar a melhorar a precisão dos resultados, use os termos "nós gerenciados" e "instâncias gerenciadas" em seus prompts, em vez de apenas "nós" e "instâncias".
2. Para consultar resultados em várias contas que fazem parte de uma organização, conforme configurado em AWS Organizations, você deverá estar conectado à conta de administrador delegado na região de origem designada.
3. Na conta de administrador delegado, use termos para ajudar a Amazon Q a entender que você está perguntando sobre nós e instâncias em toda a organização usando especificamente termos como "na minha organização" ou "na minha conta 123456789012".

Tópicos

- Exemplos de perguntas para o Amazon Q
- Nomes e versões de sistemas operacionais compatíveis para prompts

Exemplos de perguntas para o Amazon Q

Na tabela a seguir, fornecemos exemplos de perguntas que demonstram algumas maneiras pelas quais você pode criar consultas do Amazon Q que levam a melhores resultados.

Também fornecemos exemplos dos filtros que o Amazon Q aplicará ao executar o comando ListNodes, os quais são gerados a partir do conteúdo do seu prompt.

Exemplo de pergunta em linguagem natural	Filtros aplicados do Amazon Q
`Show me my Windows managed nodes.`	`PlatformType = Windows`

Exemplo de pergunta em linguagem natural	Filtros aplicados do Amazon Q
List my managed instances in account 123456789012.	`AccountId = 123456789012`
Show me all managed nodes running Amazon Linux 1 across my organization.	`PlatformName = Amazon Linux` `PlatformVersion = 1`
Show me all managed instances running Microsoft Windows Server 2019 Datacenter in my organization.	`PlatformName = Microsoft Windows Server 2019 Datacenter`
Can you show me all managed nodes with SSM Agent version 3.3.1142.0?	`AgentType = amazon-ssm-agent` `AgentVersion = 3.3.1142.0`

List all Amazon Linux 2 managed instances in account 123456789012 that have SSM Agent version 3.3.1230.0.	`PlatformName = Amazon Linux`
	`PlatformVersion = 2`
	`AccountId = 123456789012`
	`AgentType = amazon-ssm-agent`
	`AgentVersion = 3.3.1230.0`

What Microsoft Windows Server 2008 R2 Enterprise managed nodes are running in the eu-central-1 region across my entire organization?	`PlatformName = Microsoft Windows Server 2008 R2 Enterprise` `Region = eu-central-1`

Show me all managed instances running CentOS Linux 7 in ou-d6ty-gxdma6vm.	`PlatformName = CentOS Linux`
	`PlatformVersion = 7`
	`OrganizationalUnitId = ou-d6ty-gxdma6vm`

What Ubuntu managed instances are in account 123456789012?	`PlatformName = Ubuntu` `AccountId = 123456789012`

Exemplo de pergunta em linguagem natural	Filtros aplicados do Amazon Q
List my Linux managed instances.	`PlatformType = Linux`
Find my macOS managed nodes.	`PlatformType = macOS`
Show me all Amazon Linux managed nodes in my org.	`PlatformName = Amazon Linux`
List managed nodes running Amazon Linux 2 .	`PlatformName = Amazon Linux` `PlatformVersion = 2`

List the managed nodes with Ubuntu 16.04 in account 123456789012.	`PlatformName = Ubuntu`
	`PlatformVersion = 16.04`
	`AccountId = 123456789012`

Find all managed nodes that have an SSM Agent version that is not 3.3.987.0.	`AgentType = amazon-ssm-agent` `AgentVersion != 3.3.987.0`
List all managed instances that are not running a Linux operating system.	`PlatformType != Linux`

Nomes e versões de sistemas operacionais compatíveis para prompts

Quando você pergunta ao Amazon Q sobre os nós gerenciados em sua conta, é útil fornecer o nome de um sistema operacional conforme rotulado no Systems Manager. Você também pode fornecer números de versão para restringir ainda mais os resultados. Por exemplo, conforme representado nas tabelas a seguir, você pode solicitar resultados específicos sobre **`macOS 14.5`**, **`Microsoft Windows Server 2019 Datacenter`** e **`AlmaLinux 9.2 through 9.4`**, apenas para citar alguns exemplos.

Essas listas podem não ser completas e são oferecidas apenas como exemplos.

macOS

nome-da-plataforma	Números de versão
macOS	11.6.7, 11.7.10, 12.6.6, 12.7.6, 13.2, 13.4, 13.7, 14.1, 14.5, 14.6.1, 15.0

Windows

Versões	Números de versão
Microsoft Windows Server 2008 Enterprise	6.0.6003
Microsoft Windows Server 2008 R2 Datacenter	6.1.7601
Microsoft Windows Server 2008 R2 Enterprise	6.1.7601
Microsoft Windows Server 2008 R2 Standard	6.1.7601
Microsoft Windows Server 2012 R2 Datacenter	6.3.9600
Microsoft Windows Server 2012 R2 Standard	6.3.9600
Microsoft Windows Server 2012 Standard	6.2.9200
Microsoft Windows Server 2016 Datacenter	N/D
Microsoft Windows Server 2016 Standard	10.0.14393
Microsoft Windows Server 2019 Datacenter	N/D
Microsoft Windows Server 2019 Standard	N/D
Microsoft Windows Server 2022 Datacenter	N/D
Microsoft Windows Server 2022 Standard	10.0.20348

Linux

Nomes de plataformas	Números de versão
AlmaLinux	8.10, 9.2, 9.3, 9.4
Amazon Linux 2	2.0 e posteriores
Amazon Linux 2023	2023.0.20230315.0 e posteriores
Amazon Linux	2015.03, 2015.09, 2016.03, 2016.09, 2017.03, 2017.09, 2018.03
BottleRocket	1.14.3, 1.16.1, 1.18.0, 1.19.1, 1.19.2, 1.19.5, 1.20.0, 1.20.1, 1.20.2, 1.20.3, 1.20.5, 1.21.1, 1.23.0, 1.24.0, 1.24.1, 1.25.0, 1.26.1,
CentOS	6.7, 6.8, 6.9 (Final), 6.10, 6.10 (Final)
CentOS Linux	7, 7.2.1511, 7.2.1511 (Básico), 7.3.1611, 7.3.1611 (Básico), 7.4.1708, 7.5.1804, 7.5.1804 (Básico), 7.6.1810, 7.7.1908, 7.8.2003, 7.9.2009, 7.9.2009 (Básico), 8.2.2004, 8.5.2111
CentOS Stream	8, 9
Debian GNU/Linux	8 – 12
Oracle Linux Server	7.8, 8.2, 8.3, 8.8, 8.9, 8.10, 9.4
Red Hat Enterprise Linux	8.2, 8.3, 8.4, 8.5, 8.6, 8.7, 8.8, 8.9, 8.10, 9.2, 9.3, 9.4
Servidor do Red Hat Enterprise Linux	6.9 (Santiago), 6.10 (Santiago), 7.3, 7.6, 7.7, 7.8,7.9
Rocky Linux	8.6, 8.7, 8.8, 8.9, 8.10, 9.1, 9.2, 9.3, 9.4
SLES	12.4, 15, 15.1, 15.2, 15.3, 15.4, 15.5
Ubuntu	14.04, 16.04, 18.04, 20.04, 22.04, 24.04

Explorar nós gerenciados com o Amazon Q

A integração do Systems Manager com o Amazon Q Developer permite fazer perguntas sobre nós gerenciados em sua frota de qualquer lugar no AWS Management Console em que a interface do Amazon Q esteja disponível.

Para obter mais informações sobre como interagir com o Amazon Q, consulte Conversando com o Amazon Q Developer sobre AWS no Guia do usuário do Amazon Q Developer.

Para explorar nós gerenciados com o Amazon Q

1. Em qualquer lugar do AWS Management Console, escolha o ícone do Amazon Q ().
2. No campo de prompt na parte inferior do painel do Amazon Q, faça uma pergunta sobre os nós gerenciados em sua conta ou organização.

   > **Tip**
   >
   > Para obter dicas de como criar prompts eficazes, revise as informações em Como criar prompts eficazes para perguntar ao Amazon Q sobre sua frota.

3. Examine as informações sobre nós específicos ou escolha Abrir console do AWS Systems Manager para continuar explorando.

Visualizar detalhes individuais do nó e realizando ações em um nó

Em uma lista na página Explorar nós no Systems Manager, é possível selecionar um nó individual para visualizar detalhes abrangentes sobre a máquina ou realizar diversas ações no nó. A página Geral na página de detalhes apresenta informações abrangentes sobre o nó.

Para visualizar detalhes individuais do nó e realizar ações em um nó

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Explorar nós.
3. (Opcional) Siga as etapas em Escolher uma visualização de filtro para resumos de nós gerenciados para refinar a lista de nós gerenciados exibidos para sua organização ou conta.
4. Na coluna ID do nó, escolha o ID vinculado de um nó.

5. Para ver mais detalhes sobre o nó, na navegação à esquerda, na lista Propriedades, escolha uma propriedade para visualizar mais informações sobre:

   - Tags: visualize uma lista de tags aplicadas ao nó. Também é possível adicionar ou remover tags.
   - Inventário: escolha um tipo de inventário, como AWS:Application ou AWS:Network, para visualizar os detalhes do inventário do nó.
   - Associações: visualize detalhes sobre todas as associações do State Manager aplicadas ao nó, incluindo detalhes como status e nome do documento do SSM associado.
   - Patches: visualize informações resumidas sobre os patches e o status do patch para o nó.
   - Conformidade da configuração: visualize detalhes de conformidade do nó, como status de conformidade e gravidade do problema de conformidade.

6. Para realizar ações no nó, use as seguintes opções no menu Ações do nó:

   > **Note**
   >
   > Essas ações estão disponíveis somente para nós gerenciados na região e na Conta da AWS em que você está trabalhando no momento. Para nós gerenciados aos quais você pode ter acesso em outras contas ou regiões, é possível acessar uma lista de Propriedades.

   - Conectar, iniciar sessão do terminal: conecte ao nó usando o AWS Systems Manager Session Manager.
   - Ferramentas
     - Visualizar sistema de arquivos: navegue pelo conteúdo da estrutura de diretórios do nó. Adicione, renomeie e remova diretórios. Recorte ou copie e cole arquivos.
     - Visualizar contadores de performance: visualize informações de performance sobre o nó, como utilização da CPU, tráfego de rede e outros tipos de utilização.
     - Processos gerenciados: visualize informações sobre o uso de recursos no nó. Inicie ou interrompa processos no nó.
     - Gerenciar usuários e grupos: visualize, adicione ou exclua contas de usuário e grupos de usuários no nó.
     - Executar comando de execução: use AWS Systems Manager Run Command para gerenciar a configuração do nó. Run Command usa documentos do Systems Manager para

realizar alterações sob demanda, como atualizar aplicações ou executar scripts de shell do Linux e comandos do Windows PowerShell.

- Aplicar patches em nós: use o recurso Aplicar patch agora no AWS Systems Manager Patch Manager para executar uma operação de correção sob demanda no nó via console.

> **Note**
>
> As tarefas anteriores também podem ser iniciadas no menu Ferramentas na navegação à esquerda.

- Configurações do nó
  - Adicionar tags: aplique pares de chave-valor de tag ao nó.
  - Redefinir senha do usuário do nó: defina uma nova senha para um usuário especificado no nó.
  - Modificar perfil do IAM: altere o perfil do IAM associado ao nó. Crie um perfil do IAM para associar ao nó.

Baixar ou exportar um relatório de nó gerenciado

É possível usar o recurso Explorar nós do Systems Manager para visualizar listas filtradas ou não filtradas de nós gerenciados para sua organização ou conta da AWS no console do Systems Manager. Nos casos em que você deseja visualizar os dados offline ou processá-los em outra aplicação, é possível salvar o relatório como um arquivo `CSV` ou `JSON`.

Dependendo do tamanho do relatório, você será solicitado a baixar o relatório para sua máquina local ou exportá-lo para um bucket do Amazon S3. Os relatórios são salvos nos buckets do S3 somente no formato `CSV`.

Para baixar ou exportar um relatório de nó gerenciado

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Explorar nós.
3. (Opcional) Siga as etapas em Escolher uma visualização de filtro para resumos de nós gerenciados para refinar a lista de nós gerenciados exibidos para sua organização ou conta.
4. Escolha Relatório
   ( ).

5. Se a caixa de diálogo Baixar relatório for exibida, faça o seguinte:

   a. Em Nome do arquivo, insira um nome para o arquivo. Recomendamos especificar um nome que represente o escopo do relatório, como `all-organization-nodes` ou `ec2-instances-out-of-date-agent`.

   b. Em Colunas incluídas, especifique se deseja incluir colunas para todos os detalhes dos nós disponíveis ou somente aqueles que você selecionou para sua exibição atual.

   > **Tip**
   >
   > Para obter informações sobre o gerenciamento de colunas na exibição do relatório, consulte Gerenciar o conteúdo e a aparência de relatórios de nós.

   c. Em Formato do arquivo, selecione CSV ou JSON, dependendo de como você usará o arquivo.

   d. Em Título da planilha, para incluir uma linha de cabeçalhos de coluna em um arquivo `CSV`, selecione Incluir linha de nomes de colunas.

   e. Escolha Baixar.

   O relatório é salvo no local de download padrão de acordo com as configurações do seu navegador.

6. Se a caixa de diálogo Exportar para o Amazon S3 for exibida, faça o seguinte:

   a. Em URI do S3, insira o URI do bucket para o qual exportar o relatório.

   > **Tip**
   >
   > Para visualizar uma lista dos buckets do console do Amazon S3, escolha Visualizar. Para selecionar em uma lista de buckets em sua conta Navegar pelo S3.

   b. Para Método de autorização, especifique o perfil de serviço a ser usado para fornecer permissões para exportar o relatório para o bucket.

   Se você optar por permitir que o Systems Manager crie o perfil para você, ele fornecerá todas as permissões e declarações de confiança necessárias para a operação.

   Se você quiser usar ou criar seu próprio perfil, o perfil deverá incluir as permissões e declarações de confiança necessárias. Para obter informações sobre como criar essa

função, consulte Criar um perfil de serviço personalizado para exportar relatórios de diagnóstico para o S3.

c. Selecione Enviar.

Criar um perfil de serviço personalizado para exportar relatórios de diagnóstico para o S3

Ao visualizar listas filtradas ou não filtradas de nós gerenciados para sua organização ou conta da AWS na página Explorar nós do Systems Manager, é possível exportar a lista como um relatório para um bucket do Amazon S3 na forma de um arquivo CSV.

Para fazer isso, é necessário especificar um perfil de serviço com as permissões necessárias e a política de confiança para a operação. É possível fazer com que o Systems Manager crie o perfil para você durante o processo de download do relatório. Opcionalmente, você mesmo pode criar o perfil e a política necessários.

Para criar um perfil de serviço personalizado para exportar relatórios de diagnóstico para o S3

1. Siga as etapas em Criar políticas usando o editor JSON no Guia do usuário do IAM.

   - Use o conteúdo a seguir para a política, certificando-se de substituir os *valores do espaço reservado* por suas próprias informações.

     JSON

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

```
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketAcl",
        "s3:ListBucket",
        "s3:PutLifecycleConfiguration",
        "s3:GetLifecycleConfiguration"
      ],
      "Resource": "arn:aws:s3:::s3-bucket-name",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "account-id"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:ListNodes"
      ],
      "Resource": "*"
    }
  ]
}
```

- Atribua um nome à política para obter ajuda para reconhecê-la facilmente na próxima etapa.

2. Siga o procedimento Criar um perfil do IAM usando políticas de confiança personalizadas (console) no Guia do usuário do IAM.

   - Na etapa 4, insira a política de confiança a seguir, certificando-se de substituir os *valores do espaço reservado* por suas próprias informações.

     JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SSMAssumeRole",
      "Effect": "Allow",
      "Principal": {
        "Service": "ssm.amazonaws.com"
      },
```

```
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "account-id"
        }
      }
    }
  ]
}
```

3. Para a etapa 10, escolha Etapa 2: adicionar permissões e selecione o nome da política que você criou na etapa anterior.

Após criar a função, você poderá selecioná-la seguindo as etapas em Baixar ou exportar um relatório de nó gerenciado.

Gerenciar o conteúdo e a aparência de relatórios de nós

É possível usar o recurso Explorar nós do Systems Manager para visualizar listas filtradas ou não filtradas de nós gerenciados para sua organização ou conta da AWS no console do Systems Manager. É possível escolher entre mais de uma dúzia de campos para incluir em suas listas de nós, como ID do nó, Nome do sistema operacional, Região e muito mais. Também é possível reordenar as colunas de suas listas e relatórios e alterar a forma como a lista é exibida no console.

Para gerenciar o conteúdo e a aparência de um relatório de nó

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Explorar nós.
3. Na área Nós, escolha o ícone de engrenagem de preferências ().
4. Na caixa de diálogo Preferências faça o seguinte:
   a. Em Tamanho da página, escolha quantas linhas estão incluídas em cada exibição do console, 10, 25 ou 50.
   b. Em Quebrar linhas, selecione a caixa para exibir todo o conteúdo de uma célula na largura disponível da coluna.
   c. Em Linhas listradas, selecione a caixa para exibir linhas alternadas de planos de fundo claros e sombreados.

d. Em Selecionar conteúdo visível, faça o seguinte:

- Ative ou desative colunas individuais para a exibição da lista e os relatórios.
- Para alterar a ordem das colunas, clique e segure a alça de arrastar
  ()
  do nome de uma coluna e arraste-a para cima ou para baixo na lista.

5. Escolha Confirmar.

Acesso a nós just-in-time usando o Systems Manager

O Systems Manager ajuda você a melhorar a segurança de seus nós oferecendo suporte ao acesso just-in-time. O acesso a nós just-in-time permite que os usuários solicitem acesso temporário e com limite de tempo aos nós, que você poderá aprovar somente quando o acesso for realmente necessário. Isso elimina a necessidade de fornecer acesso prolongado aos nós gerenciados pelas políticas do IAM. Além disso, o Systems Manager fornece gravação de sessões para sessões RDP em nós do Windows Server para ajudar você a atender aos requisitos de conformidade, realizar análises de causa raiz e muito mais. Para usar o acesso a nós just-in-time, você deve configurar o console unificado do Systems Manager.

Com o acesso a nós just-in-time, você cria políticas granulares do IAM para garantir que somente os usuários permitidos possam enviar solicitações de acesso aos seus nós. Em seguida, você cria políticas de aprovação que definem as aprovações necessárias para se conectar aos seus nós. Para o acesso a nós just-in-time, existem políticas de aprovação automática e de aprovação manual. Uma política de aprovação automática define a quais nós os usuários podem se conectar automaticamente. As políticas de aprovação manual definem o número e os níveis de aprovações manuais que devem ser fornecidas para acessar os nós que você especificar. Além disso, você pode criar uma política de negação de acesso. Uma política de negação de acesso impede explicitamente a aprovação automática de solicitações de acesso aos nós que você especificar. Uma política de negação de acesso se aplica a todas as contas em uma organização do AWS Organizations. As políticas de aprovação automática e manual se aplicam somente às Contas da AWS e às Regiões da AWS onde foram criadas.

Quando um usuário tenta se conectar a um nó, ele é solicitado a inserir um motivo para acessá-lo. Depois, suas políticas de aprovação são avaliadas. Dependendo de suas políticas, os usuários se conectam automaticamente ao nó de destino ou o Systems Manager cria automaticamente uma solicitação de aprovação manual em nome do solicitante. Os aprovadores especificados na

política de aprovação manual que se aplica ao nó são então notificados da solicitação de acesso e podem aprovar ou negar a solicitação. Os aprovadores e solicitantes podem ser notificados por e-mail ou por meio do Amazon Q Developer na integração de aplicações de chat com o Slack ou o Microsoft Teams. O Systems Manager só concede acesso aos nós solicitados quando os aprovadores especificados fornecem todas as aprovações necessárias. Depois que todas as aprovações necessárias forem recebidas, o usuário poderá iniciar quantas sessões no nó forem necessárias durante a janela de acesso especificada na política de aprovação. O Systems Manager não encerra automaticamente as sessões de acesso a nós just-in-time. Como prática recomendada, especifique valores para as preferências de duração máxima da sessão e tempo limite de sessão ociosa da sessão. Essas preferências evitam que os usuários permaneçam conectados aos nós além da janela de acesso aprovada.

Recomendamos o uso de uma combinação de políticas de aprovação para ajudar você a proteger os nós com dados mais críticos e permitir que os usuários se conectem a nós menos críticos sem intervenção. Por exemplo, você pode exigir aprovações manuais para solicitações de acesso aos nós do banco de dados e aprovar automaticamente sessões para nós não persistentes da camada de apresentação.

O Systems Manager é compatível com o acesso a nós just-in-time para usuários federados com o Centro de Identidade do IAM ou o IAM. Quando um usuário federado envia uma solicitação de acesso, ele especifica o nó de destino e o motivo da necessidade de se conectar a ele. O Systems Manager compara a identidade do usuário com os parâmetros definidos nas políticas de aprovação da sua organização. Quando as condições da política de aprovação automática são atendidas, ou os aprovadores fornecem aprovações manualmente, o solicitante pode se conectar ao nó de destino. Quando um usuário tenta se conectar a um nó aprovado, o Systems Manager cria e usa um token temporário para estabelecer a sessão.

Como o serviço do Systems Manager gerencia a autenticação para solicitações de acesso e o estabelecimento de sessões, você não precisa usar políticas do IAM para gerenciar o acesso aos seus nós. Ao usar o acesso a nós just-in-time, o Systems Manager ajuda sua organização a ficar mais próxima de privilégios permanentes zero, já que você só precisa permitir que os usuários criem solicitações de acesso em vez de permitir que eles iniciem sessões com permissões prolongadas em seus nós. Para ajudar você a atender aos requisitos de conformidade, o Systems Manager retém todas as solicitações de acesso por um ano. O Systems Manager também emite eventos do EventBridge para acesso a nós just-in-time para solicitações de acesso com falha e atualizações de status para solicitações de acesso para aprovações manuais. Para obter mais informações, consulte, Monitorar eventos do Systems Manager com o Amazon EventBridge.

Tópicos

- Configurar o acesso just-in-time com o Systems Manager
- Iniciar uma sessão de acesso a nós just-in-time
- Gerenciar solicitações de acesso just-in-time
- Migrar para o acesso a nós just-in-time do Session Manager
- Desabilitar o acesso just-in-time com o Systems Manager
- Perguntas frequentes sobre acesso a nós just-in-time

Configurar o acesso just-in-time com o Systems Manager

A configuração do acesso a nós just-in-time com o Systems Manager envolve várias etapas. Primeiro, você escolhe os destinos em que deseja configurar o acesso a nós just-in-time. Os destinos consistem em unidades organizacionais (UOs) do AWS Organizations e Regiões da AWS. Por padrão, os mesmos destinos que você escolheu ao configurar o console unificado do Systems Manager são selecionados para acesso a nós just-in-time. Você pode optar por configurar o acesso a nós just-in-time para todos os mesmos destinos ou para um subconjunto dos destinos que você especificou ao configurar o console unificado do Systems Manager. Não há suporte para a adição de novos destinos que não foram selecionados quando você configurou o console unificado do Systems Manager.

Em seguida, você criará políticas de aprovação para determinar quando as conexões com os nós exigem aprovação manual e quando são aprovadas automaticamente. As políticas de aprovação são gerenciadas por cada conta em sua organização. Você também pode compartilhar uma política da conta do administrador delegado para negar explicitamente a aprovação automática de conexões com nós específicos.

> **Note**
>
> A configuração do acesso a nós just-in-time não afeta as preferências ou políticas existentes do IAM que você configurou para o Session Manager. Você deve remover as permissões para as ações do Session Manager, como `StartSession`, de suas políticas do IAM para garantir que somente o acesso a nós just-in-time seja usado quando os usuários tentarem se conectar aos nós. Depois de configurar o acesso a nós just-in-time, recomendamos testar suas políticas de aprovação com um subconjunto de usuários e nós para verificar se suas

políticas estão funcionando conforme desejado, antes de remover as permissões para o Session Manager.

Compatibilidade com autenticação

Observe os seguintes detalhes sobre o suporte à autenticação usado para acesso a nós just-in-time:

- O acesso a nós just-in-time não é compatível com o tipo de autenticação de login único ao se conectar a instâncias Windows Server com o Remote Desktop.
- Há suporte somente a credenciais de segurança temporárias `AssumeRole` do AWS Security Token Service (AWS STS). Para obter mais informações, consulte os seguintes tópicos no Guia do usuário do IAM:
- • Credenciais de segurança temporárias no IAM
  - Comparar credenciais do AWS STS
  - Solicitar credenciais de segurança temporárias

As políticas do IAM a seguir descrevem as permissões necessárias para administrar e permitir que os usuários criem solicitações de acesso a nós just-in-time com o Systems Manager. Depois de verificar se você tem as permissões necessárias para usar o acesso a nós just-in-time com o Systems Manager, você pode continuar o processo de configuração. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

Política do IAM para habilitar o acesso a nós just-in-time

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "QuickSetupConfigurationManagers",
            "Effect": "Allow",
            "Action": [
                "ssm-quicksetup:CreateConfigurationManager",
                "ssm-quicksetup:DeleteConfigurationManager",
                "ssm-quicksetup:GetConfiguration",
                "ssm-quicksetup:GetConfigurationManager",
```

```
                "ssm-quicksetup:GetServiceSettings",
                "ssm-quicksetup:ListConfigurationManagers",
                "ssm-quicksetup:ListConfigurations",
                "ssm-quicksetup:ListQuickSetupTypes",
                "ssm-quicksetup:ListTagsForResource",
                "ssm-quicksetup:TagResource",
                "ssm-quicksetup:UntagResource",
                "ssm-quicksetup:UpdateConfigurationDefinition",
                "ssm-quicksetup:UpdateConfigurationManager",
                "ssm-quicksetup:UpdateServiceSettings"
            ],
            "Resource": "*"
        },
        {
            "Sid": "QuickSetupDeployments",
            "Effect": "Allow",
            "Action": [
                "cloudformation:DescribeStackSetOperation",
                "cloudformation:ListStacks",
                "cloudformation:DescribeStacks",
                "cloudformation:DescribeStackResources",
                "cloudformation:ListStackSetOperations",
                "cloudformation:ListStackInstances",
                "cloudformation:DescribeStackSet",
                "cloudformation:ListStackSets",
                "cloudformation:DescribeStackInstance",
                "cloudformation:DescribeOrganizationsAccess",
                "cloudformation:ActivateOrganizationsAccess",
                "cloudformation:GetTemplate",
                "cloudformation:ListStackSetOperationResults",
                "cloudformation:DescribeStackEvents",
                "cloudformation:UntagResource",
                "ssm:DescribeAutomationExecutions",
                "ssm:GetAutomationExecution",
                "ssm:ListAssociations",
                "ssm:DescribeAssociation",
                "ssm:GetDocument",
                "ssm:ListDocuments",
                "ssm:DescribeDocument",
                "ssm:GetOpsSummary",
                "organizations:DeregisterDelegatedAdministrator",
                "organizations:DescribeAccount",
                "organizations:DescribeOrganization",
                "organizations:ListDelegatedAdministrators",
```

```
                "organizations:ListRoots",
                "organizations:ListParents",
                "organizations:ListOrganizationalUnitsForParent",
                "organizations:DescribeOrganizationalUnit",
                "organizations:ListAWSServiceAccessForOrganization",
                "iam:ListRoles",
                "iam:ListRolePolicies",
                "iam:GetRole",
                "iam:CreatePolicy",
                "cloudformation:TagResource"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudformation:RollbackStack",
                "cloudformation:CreateStack",
                "cloudformation:UpdateStack",
                "cloudformation:DeleteStack"
            ],
            "Resource": [
                "arn:aws:cloudformation:*:*:stack/StackSet-AWS-QuickSetup-
JITNA*",
                "arn:aws:cloudformation:*:*:stack/AWS-QuickSetup-*",
                "arn:aws:cloudformation:*:*:type/resource/*",
                "arn:aws:cloudformation:*:*:stack/StackSet-SSMQuickSetup"
            ]
        },
        {
            "Sid": "StackSetOperations",
            "Effect": "Allow",
            "Action": [
                "cloudformation:CreateStackSet",
                "cloudformation:UpdateStackSet",
                "cloudformation:DeleteStackSet",
                "cloudformation:DeleteStackInstances",
                "cloudformation:CreateStackInstances",
                "cloudformation:StopStackSetOperation"
            ],
            "Resource": [
                "arn:aws:cloudformation:*:*:stackset/AWS-QuickSetup-JITNA*",
                "arn:aws:cloudformation:*:*:type/resource/*",
```

```
                "arn:aws:cloudformation:*:*:stackset-target/AWS-QuickSetup-
JITNA*:*"
            ]
        },
        {
            "Sid": "IamRolesMgmt",
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:GetRole",
                "iam:AttachRolePolicy",
                "iam:PutRolePolicy",
                "iam:DetachRolePolicy",
                "iam:GetRolePolicy",
                "iam:ListRolePolicies"
            ],
            "Resource": [
                "arn:aws:iam::*:role/AWS-QuickSetup-JITNA*",
                "arn:aws:iam::*:role/service-role/AWS-QuickSetup-JITNA*"
            ]
        },
        {
            "Sid": "IamPassRole",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/AWS-QuickSetup-JITNA*",
                "arn:aws:iam::*:role/service-role/AWS-QuickSetup-JITNA*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ssm.amazonaws.com",
                        "ssm-quicksetup.amazonaws.com",
                        "cloudformation.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "SSMAutomationExecution",
```

```
            "Effect": "Allow",
            "Action": "ssm:StartAutomationExecution",
            "Resource": "arn:aws:ssm:us-east-1:111122223333:automation-
definition/AWS-EnableExplorer:*"
        },
        {
            "Sid": "SSMAssociationPermissions",
            "Effect": "Allow",
            "Action": [
                "ssm:DeleteAssociation",
                "ssm:CreateAssociation",
                "ssm:StartAssociationsOnce"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:association/*"
        },
        {
            "Sid": "SSMResourceDataSync",
            "Effect": "Allow",
            "Action": [
                "ssm:CreateResourceDataSync",
                "ssm:UpdateResourceDataSync"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:resource-data-sync/
AWS-QuickSetup-*"
        },
        {
            "Sid": "ListResourceDataSync",
            "Effect": "Allow",
            "Action": [
                "ssm:ListResourceDataSync"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CreateServiceLinkedRoles",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "accountdiscovery.ssm.amazonaws.com",
                        "ssm.amazonaws.com",
```

```
                        "ssm-quicksetup.amazonaws.com",
                        "stacksets.cloudformation.amazonaws.com"
                    ]
                }
            },
            "Resource": "*"
        },
        {
            "Sid": "CreateStackSetsServiceLinkedRole",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::*:role/aws-
service-role/stacksets.cloudformation.amazonaws.com/
AWSServiceRoleForCloudFormationStackSetsOrgAdmin"
        },
        {
            "Sid": "AllowSsmJitnaPoliciesCrudOperations",
            "Effect": "Allow",
            "Action": [
                "ssm:CreateDocument",
                "ssm:UpdateDocument",
                "ssm:UpdateDocumentDefaultVersion",
                "ssm:GetDocument",
                "ssm:DescribeDocument",
                "ssm:DeleteDocument"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:document/SSM-
JustInTimeAccessDenyAccessOrgPolicy"
            ],
            "Condition": {
                "StringEquals": {
                    "ssm:DocumentType": [
                        "AutoApprovalPolicy"
                    ]
                }
            }
        },
        {
            "Sid": "AllowAccessRequestOpsItemOperations",
            "Effect": "Allow",
            "Action": [
```

```
                "ssm:GetOpsItem",
                "ssm:DescribeOpsItems",
                "ssm:GetOpsSummary",
                "ssm:DeleteOpsItem",
                "ssm:ListOpsItemEvents"
            ],
            "Resource": "*"
        },
        {
            "Sid": "IdentityCenterPermissions",
            "Effect": "Allow",
            "Action": [
                "sso:DescribeRegisteredRegions",
                "sso:ListDirectoryAssociations",
                "identitystore:GetUserId",
                "identitystore:DescribeUser",
                "identitystore:DescribeGroup",
                "identitystore:ListGroupMembershipsForMember"
            ],
            "Resource": "*"
        }
    ]
}
```

Política do IAM para configurar o acesso a nós just-in-time

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSsmJitnaPoliciesCrudOperations",
            "Effect": "Allow",
            "Action": [
                "ssm:CreateDocument",
                "ssm:UpdateDocument",
                "ssm:UpdateDocumentDefaultVersion",
                "ssm:GetDocument",
                "ssm:DescribeDocument",
                "ssm:DeleteDocument"
            ],
```

```
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:document/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ssm:DocumentType": [
                        "ManualApprovalPolicy",
                        "AutoApprovalPolicy"
                    ]
                }
            }
        },
        {
            "Sid": "AllowSsmJitnaPoliciesListOperations",
            "Effect": "Allow",
            "Action": [
                "ssm:ListDocuments",
                "ssm:ListDocumentVersions"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/SSM-
JustInTimeAccessTokenRole",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "justintimeaccess.ssm.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "AllowAccessRequestOpsItemOperations",
            "Effect": "Allow",
            "Action": [
                "ssm:GetOpsItem",
                "ssm:DescribeOpsItems",
                "ssm:GetOpsSummary",
                "ssm:DeleteOpsItem",
                "ssm:ListOpsItemEvents"
            ],
```

```
            "Resource": "*"
        },
        {
            "Sid": "AllowSessionManagerPreferencesOperation",
            "Effect": "Allow",
            "Action": [
                "ssm:CreateDocument",
                "ssm:GetDocument",
                "ssm:DescribeDocument",
                "ssm:UpdateDocument",
                "ssm:DeleteDocument"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:document/SSM-
SessionManagerRunShell",
            "Condition": {
                "StringEquals": {
                    "ssm:DocumentType": "Session"
                }
            }
        },
        {
            "Sid": "AllowSessionManagerOperations",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeSessions",
                "ssm:GetConnectionStatus",
                "ssm:TerminateSession"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowRDPConnectionRecordingOperations",
            "Effect": "Allow",
            "Action": [
                "ssm-guiconnect:UpdateConnectionRecordingPreferences",
                "ssm-guiconnect:GetConnectionRecordingPreferences",
                "ssm-guiconnect:DeleteConnectionRecordingPreferences"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowRDPConnectionRecordingKmsOperation",
            "Effect": "Allow",
            "Action": [
```

```
                "kms:CreateGrant"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/SystemsManagerJustInTimeNodeAccessManaged":
"true"
                },
                "StringLike": {
                    "kms:ViaService": "ssm-guiconnect.*.amazonaws.com"
                },
                "Bool": {
                    "aws:ViaAWSService": "true"
                }
            }
        },
        {
            "Sid": "AllowFleetManagerOperations",
            "Effect": "Allow",
            "Action": [
                "ssm-guiconnect:GetConnection",
                "ssm-guiconnect:ListConnections"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SNSTopicManagement",
            "Effect": "Allow",
            "Action": [
                "sns:CreateTopic",
                "sns:SetTopicAttributes"
            ],
            "Resource": [
                "arn:aws:sns:us-east-1:111122223333:SSM-JITNA*"
            ]
        },
        {
            "Sid": "SNSListTopics",
            "Effect": "Allow",
            "Action": [
                "sns:ListTopics"
            ],
            "Resource": "*"
        },
```

```
        {
            "Sid": "EventBridgeRuleManagement",
            "Effect": "Allow",
            "Action": [
                "events:PutRule",
                "events:PutTargets"
            ],
            "Resource": [
                "arn:aws:events:us-east-1::rule/SSM-JITNA*"
            ]
        },
        {
            "Sid": "ChatbotSlackManagement",
            "Effect": "Allow",
            "Action": [
                "chatbot:CreateSlackChannelConfiguration",
                "chatbot:UpdateSlackChannelConfiguration",
                "chatbot:DescribeSlackChannelConfigurations",
                "chatbot:DescribeSlackWorkspaces",
                "chatbot:DeleteSlackChannelConfiguration",
                "chatbot:RedeemSlackOauthCode",
                "chatbot:DeleteSlackWorkspaceAuthorization",
                "chatbot:GetSlackOauthParameters"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ChatbotTeamsManagement",
            "Effect": "Allow",
            "Action": [
                "chatbot:ListMicrosoftTeamsChannelConfigurations",
                "chatbot:CreateMicrosoftTeamsChannelConfiguration",
                "chatbot:UpdateMicrosoftTeamsChannelConfiguration",
                "chatbot:ListMicrosoftTeamsConfiguredTeams",
                "chatbot:DeleteMicrosoftTeamsChannelConfiguration",
                "chatbot:RedeemMicrosoftTeamsOauthCode",
                "chatbot:DeleteMicrosoftTeamsConfiguredTeam",
                "chatbot:GetMicrosoftTeamsOauthParameters",
                "chatbot:TagResource"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SSMEmailSettings",
```

```
            "Effect": "Allow",
            "Action": [
                "ssm:UpdateServiceSetting",
                "ssm:GetServiceSetting"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:servicesetting/ssm/access-
request/email-role-mapping",
                "arn:aws:ssm:us-east-1:111122223333:servicesetting/ssm/access-
request/enabled-email-notifications"
            ]
        },
        {
            "Sid": "AllowViewingJitnaCloudWatchMetrics",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "AWS/SSM/JustInTimeAccess"
                }
            }
        },
        {
            "Sid": "QuickSetupConfigurationManagers",
            "Effect": "Allow",
            "Action": [
                "ssm-quicksetup:ListConfigurationManagers",
                "ssm-quicksetup:ListConfigurations",
                "ssm-quicksetup:ListQuickSetupTypes",
                "ssm-quicksetup:GetConfiguration",
                "ssm-quicksetup:GetConfigurationManager"
            ],
            "Resource": "*"
        },
        {
            "Sid": "QuickSetupDeployments",
            "Effect": "Allow",
            "Action": [
                "cloudformation:ListStacks",
```

```
                "cloudformation:DescribeStacks",
                "organizations:DescribeOrganization",
                "organizations:ListDelegatedAdministrators"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ManualPolicy",
            "Effect": "Allow",
            "Action": [
                "sso:DescribeRegisteredRegions",
                "ssm:GetServiceSetting",
                "iam:ListRoles"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SessionPreference",
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowIamListForKMS",
            "Effect": "Allow",
            "Action": [
                "iam:ListUsers"
            ],
            "Resource": "arn:aws:iam::111122223333:user/*"
        },
        {
            "Sid": "KMSPermission",
            "Effect": "Allow",
            "Action": [
                "kms:TagResource",
                "kms:ListAliases",
                "kms:CreateAlias"
            ],
            "Resource": "*"
        },
        {
            "Sid": "KMSCreateKey",
```

```
            "Effect": "Allow",
            "Action": [
                "kms:CreateKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/SystemsManagerJustInTimeNodeAccessManaged":
 "true"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": [
                        "SystemsManagerJustInTimeNodeAccessManaged"
                    ]
                }
            }
        },
        {
            "Sid": "AllowIamRoleForChatbotAction",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/role name",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "chatbot.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "AllowIamServiceRoleForChat",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/aws-service-role/
management.chatbot.amazonaws.com/AWSServiceRoleForAWSChatbot"
        },
        {
            "Sid": "CloudWatchLogs",
            "Effect": "Allow",
```

```
            "Action": [
                "logs:DescribeLogGroups"
            ],
            "Resource": "arn:aws:logs:*:111122223333:log-group::log-stream:"
        },
        {
            "Sid": "IdentityStorePermissions",
            "Effect": "Allow",
            "Action": [
                "sso:ListDirectoryAssociations",
                "identitystore:GetUserId",
                "sso-directory:SearchUsers",
                "sso-directory:SearchGroups",
                "identitystore:DescribeGroup",
                "identitystore:DescribeUser"
            ],
            "Resource": "*"
        }
    ]
}
```

Política do IAM para aprovadores de solicitações de acesso

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAccessRequestDescriptions",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeOpsItems",
                "ssm:GetOpsSummary",
                "ssm:ListOpsItemEvents"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowGetSpecificAccessRequest",
            "Effect": "Allow",
            "Action": [
```

```
                "ssm:GetOpsItem"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:opsitem/*"
        },
        {
            "Sid": "AllowApprovalRejectionSignal",
            "Effect": "Allow",
            "Action": [
                "ssm:SendAutomationSignal"
            ],
            "Resource": "arn:aws:ssm:*:*:automation-execution/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/SystemsManagerJustInTimeNodeAccessManaged":
"true"
                }
            }
        },
        {
            "Sid": "QuickSetupConfigurationManagers",
            "Effect": "Allow",
            "Action": [
                "ssm-quicksetup:ListConfigurationManagers",
                "ssm-quicksetup:ListConfigurations",
                "ssm-quicksetup:GetConfigurationManager",
                "ssm-quicksetup:ListQuickSetupTypes",
                "ssm-quicksetup:GetConfiguration"
            ],
            "Resource": "*"
        },
        {
            "Sid": "QuickSetupDeployments",
            "Effect": "Allow",
            "Action": [
                "cloudformation:ListStacks",
                "cloudformation:DescribeStacks",
                "organizations:DescribeOrganization",
                "organizations:ListDelegatedAdministrators"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowSsmJitnaPoliciesCrudOperations",
            "Effect": "Allow",
```

```
            "Action": [
                "ssm:GetDocument",
                "ssm:DescribeDocument"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:document/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ssm:DocumentType": [
                        "ManualApprovalPolicy",
                        "AutoApprovalPolicy"
                    ]
                }
            }
        },
        {
            "Sid": "AllowSsmJitnaPoliciesListOperations",
            "Effect": "Allow",
            "Action": [
                "ssm:ListDocuments",
                "ssm:ListDocumentVersions"
            ],
            "Resource": "*"
        },
        {
            "Sid": "IDCPermissions",
            "Effect": "Allow",
            "Action": [
                "sso:DescribeRegisteredRegions",
                "sso:ListDirectoryAssociations",
                "identitystore:GetUserId",
                "identitystore:DescribeUser",
                "identitystore:DescribeGroup",
                "identitystore:ListGroupMembershipsForMember"
            ],
            "Resource": "*"
        }
    ]
}
```

Política do IAM para usuários de acesso a nós just-in-time

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowJITNAOperations",
            "Effect": "Allow",
            "Action": [
                "ssm:StartAccessRequest",
                "ssm:GetAccessToken"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowOpsItemCreationAndRetrieval",
            "Effect": "Allow",
            "Action": [
                "ssm:CreateOpsItem",
                "ssm:GetOpsItem"
            ],
            "Resource": "arn:aws:ssm:*:*:opsitem/*"
        },
        {
            "Sid": "AllowListAccessRequests",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeOpsItems",
                "ssm:GetOpsSummary",
                "ssm:ListOpsItemEvents",
                "ssm:DescribeSessions"
            ],
            "Resource": "*"
        },
        {
            "Sid": "RequestManualApprovals",
            "Action": "ssm:StartAutomationExecution",
            "Effect": "Allow",
            "Resource": "arn:aws:ssm:*:*:document/*",
            "Condition": {
                "StringEquals": {
```

```
                    "ssm:DocumentType": "ManualApprovalPolicy"
                }
            }
        },
        {
            "Sid": "StartManualApprovalsAutomationExecution",
            "Effect": "Allow",
            "Action": "ssm:StartAutomationExecution",
            "Resource": "arn:aws:ssm:*:*:automation-execution/*"
        },
        {
            "Sid": "AllowManualApprovalAutomationExecutionTagging",
            "Effect": "Allow",
            "Action": [
                "ssm:AddTagsToResource"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:automation-execution/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/SystemsManagerJustInTimeNodeAccessManaged":
"true"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": [
                        "SystemsManagerJustInTimeNodeAccessManaged"
                    ]
                }
            }
        },
        {
            "Sid": "CancelAccessRequestManualApproval",
            "Effect": "Allow",
            "Action": "ssm:StopAutomationExecution",
            "Resource": "arn:aws:ssm:*:*:automation-execution/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/SystemsManagerJustInTimeNodeAccessManaged":
"true"
                }
            }
        },
        {
```

```
            "Sid": "DescribeEC2Instances",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeTags",
                "ec2:GetPasswordData"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowListSSMManagedNodesAndTags",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeInstanceInformation",
                "ssm:ListTagsForResource"
            ],
            "Resource": "*"
        },
        {
            "Sid": "QuickSetupConfigurationManagers",
            "Effect": "Allow",
            "Action": [
                "ssm-quicksetup:ListConfigurationManagers",
                "ssm-quicksetup:GetConfigurationManager",
                "ssm-quicksetup:ListConfigurations",
                "ssm-quicksetup:ListQuickSetupTypes",
                "ssm-quicksetup:GetConfiguration"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowSessionManagerOperations",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeSessions",
                "ssm:GetConnectionStatus"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowRDPOperations",
            "Effect": "Allow",
            "Action": [
                "ssm-guiconnect:ListConnections",
```

```
                "ssm:GetConnectionStatus"
            ],
            "Resource": "*"
        },
        {
            "Sid": "QuickSetupDeployments",
            "Effect": "Allow",
            "Action": [
                "cloudformation:ListStacks",
                "cloudformation:DescribeStacks",
                "organizations:DescribeOrganization",
                "organizations:ListDelegatedAdministrators"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowSsmJitnaPoliciesReadOnly",
            "Effect": "Allow",
            "Action": [
                "ssm:GetDocument",
                "ssm:DescribeDocument"
            ],
            "Resource": [
                "arn:aws:ssm:*:111122223333:document/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ssm:DocumentType": [
                        "ManualApprovalPolicy",
                        "AutoApprovalPolicy"
                    ]
                }
            }
        },
        {
            "Sid": "AllowSsmJitnaPoliciesListOperations",
            "Effect": "Allow",
            "Action": [
                "ssm:ListDocuments",
                "ssm:ListDocumentVersions"
            ],
            "Resource": "*"
        },
        {
```

```
            "Sid": "ExploreNodes",
            "Effect": "Allow",
            "Action": [
                "ssm:ListNodesSummary",
                "ssm:ListNodes",
                "ssm:DescribeInstanceProperties"
            ],
            "Resource": "*"
        },
        {
            "Sid": "IdentityStorePermissions",
            "Effect": "Allow",
            "Action": [
                "sso:DescribeRegisteredRegions",
                "sso:ListDirectoryAssociations",
                "identitystore:GetUserId",
                "identitystore:DescribeUser",
                "identitystore:DescribeGroup"
            ],
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Para restringir o acesso às operações de API que criam, atualizam ou excluem políticas de aprovação, use a chave de condição `ssm:DocumentType` para os tipos de documento `AutoApprovalPolicy` e `ManualApprovalPolicy` . As operações de API `StartAccessRequest` e `GetAccessToken` não são compatíveis com as seguintes chaves de contexto globais:
>
> - aws:ViaAwsService
> - aws:MultiFactorAuthPresent
> - aws:SourceVpce
> - aws:UserAgent

Para obter mais informações sobre chaves de contexto de condição para o Systems Manager, consulte Condition keys for AWS Systems Manager na Referência de autorização do serviço.

O procedimento a seguir descreve como concluir a primeira etapa de configuração para o acesso a nós just-in-time.

Para configurar o acesso a nós just-in-time

1. Faça login na conta de administrador delegado do Systems Manager da sua organização.
2. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
3. Selecione Acesso ao nó Just-in-time no painel de navegação.
4. Escolha Habilitar nova experiência.
5. Escolha as regiões em que você deseja habilitar o acesso a nós just-in-time. Por padrão, as mesmas regiões que você escolheu ao configurar o console unificado do Systems Manager são selecionadas para acesso a nós just-in-time. Não há suporte para a escolha de novas regiões que não foram selecionadas quando você configurou o console unificado do Systems Manager.
6. Selecione Habilitar acesso a nós just-in-time.

Não há cobrança para usar o acesso a nós just-in-time por 30 dias após a habilitação do recurso. Após o período de teste de 30 dias, há uma cobrança pelo uso do acesso a nós just-in-time. Para obter mais informações, consulte Preços do AWS Systems Manager.

Criar políticas de aprovação para os nós

As políticas de aprovação definem quais aprovações os usuários precisam ter para acessar um nó. Como o acesso a nós just-in-time elimina a necessidade de permissões prolongadas para os nós por meio de políticas do IAM, você deve criar políticas de aprovação para permitir o acesso aos nós. Se não houver políticas de aprovação que se apliquem a um nó, os usuários não poderão solicitar acesso ao nó.

No acesso a nós just-in-time, há três tipos de políticas. Os tipos de política são: aprovação automática, negação de acesso e aprovação manual.

Tipos de política de acesso a nós just-in-time

- Uma política de aprovação automática define a quais nós os usuários podem se conectar automaticamente.

- As políticas de aprovação manual definem o número e os níveis de aprovações manuais que devem ser fornecidas para acessar os nós que você especificar.
- Uma política de negação de acesso impede explicitamente a aprovação automática de solicitações de acesso aos nós que você especificar.

Uma política de negação de acesso se aplica a todas as contas em uma organização do AWS Organizations. Por exemplo, você pode negar explicitamente as aprovações automáticas para o grupo `Intern` aos nós marcados com a chave `Production`. As políticas de aprovação automática e manual se aplicam somente às Contas da AWS e às Regiões da AWS onde foram criadas. Cada conta de membro em sua organização gerencia suas próprias políticas de aprovação. As políticas de aprovação são avaliadas na seguinte ordem:

1. Negar acesso
2. Aprovação automática
3. Manual

Embora você só possa ter uma política de negação de acesso por organização e uma política de aprovação automática por conta e região, você provavelmente terá várias políticas de aprovação manual em uma conta. Ao avaliar as políticas de aprovação manual, o acesso a nós just-in-time sempre favorece a política mais específica de um nó. As políticas de aprovação manual são avaliadas na seguinte ordem:

1. Tags específicas para destino
2. Todos os nós para destino

Por exemplo, você tem um nó marcado com a chave `Demo`. Na mesma conta, você tem uma política de aprovação manual que visa todos os nós e exige uma aprovação de um nível. Você também tem uma política de aprovação manual que exige duas aprovações de dois níveis para nós marcados com a chave `Demo`. O Systems Manager aplica a política que direciona a tag `Demo` ao nó, pois ela é mais específica do que a política que visa todos os nós. Isso permite que você crie uma política geral para todos os nós na sua conta, garantindo que os usuários possam enviar solicitações de acesso e, ao mesmo tempo, permitindo que você crie políticas mais granulares conforme necessário.

Dependendo da sua organização, pode haver várias tags aplicadas aos nós. Nesse cenário, se várias políticas de aprovação manual se aplicarem a um nó, as solicitações de acesso falharão. Por exemplo, um nó é marcado com as chaves `Production` e `Database`. Na mesma conta, você

tem uma política de aprovação manual que se aplica aos nós marcados com a chave `Production` e outra política de aprovação manual que se aplica aos nós marcados com a chave `Database`. Isso resulta em um conflito para o nó marcado com ambas as chaves, e as solicitações de acesso falham. O Systems Manager redireciona o usuário para a solicitação que falhou. Lá, é possível ver os detalhes sobre as políticas e tags conflitantes para que o usuário possa fazer os ajustes necessários caso tenha as permissões necessárias. Do contrário, o usuário pode notificar um colega em sua organização com as permissões necessárias para modificar as políticas. Conflitos de políticas que resultam em falhas nas solicitações de acesso emitem eventos do EventBridge, permitindo flexibilidade na criação de seus próprios fluxos de trabalho de resposta. Além disso, o Systems Manager envia notificações por e-mail sobre conflitos de políticas que resultam em solicitações de acesso com falha para os destinatários que você especificar. Para obter mais informações sobre como configurar notificações por e-mail de conflitos de políticas, consulte Configurar notificações para solicitações de acesso just-in-time.

Em uma política de negação de acesso, você usa a linguagem de política Cedar para definir a quais nós os usuários explicitamente não podem se conectar automaticamente em sua organização. Essa política é criada e compartilhada da conta do administrador delegado da sua organização. A política de negação de acesso substitui todas as políticas de aprovação automática. Você só pode ter uma política de negação de acesso por organização.

Em uma política de aprovação automática, você usa a linguagem de política Cedar para definir quais usuários podem se conectar automaticamente aos nós especificados sem aprovação manual. A duração do acesso para uma solicitação de acesso aprovada automaticamente é de uma hora. Esse valor não pode ser alterado. Você só pode ter uma política de aprovação automática por conta e região.

Em uma política de aprovação manual, você especifica a duração do acesso, quantos níveis de aprovações são necessários, o número de aprovadores necessários por nível e os nós para os quais eles podem aprovar solicitações de acesso just-in-time. A duração do acesso para uma política de aprovação manual deve estar entre 1 e 336 horas. Se você especificar vários níveis de aprovações, as aprovações para a solicitação de acesso processarão um nível por vez. Isso significa que todas as aprovações necessárias para um nível deverão ser fornecidas antes que o processo de aprovação passe para os níveis subsequentes. Se você especificar várias tags em uma política de aprovação manual, elas serão avaliadas como instruções `or`, não instruções `and`. Por exemplo, se você criar uma política de aprovação manual que inclua as tags `Application`, `Web` e `Test`, a política se aplicará a qualquer nó que esteja marcado com uma dessas chaves. A política não se aplica somente aos nós marcados com as três chaves.

Recomendamos usar uma combinação de políticas manuais com sua política de aprovação automática para ajudar você a proteger nós com dados mais críticos e, ao mesmo tempo, permitir que os usuários se conectem a nós menos críticos sem intervenção. Por exemplo, você pode exigir aprovações manuais para solicitações de acesso aos nós do banco de dados e aprovar automaticamente sessões para nós não persistentes da camada de apresentação.

Os procedimentos a seguir descrevem como criar políticas de aprovação para o acesso a nós just-in-time.

Tópicos

- Criar políticas de aprovação manual para acesso a nós just-in-time
- Estrutura de instruções e operadores integrados para políticas de aprovação automática e negação de acesso
- Criar uma política de aprovação automática para acesso a nós just-in-time
- Criar uma política de negação de acesso para acesso a nós just-in-time
- Criar políticas de aprovação para acesso a nós just-in-time com o Amazon Q

Criar políticas de aprovação manual para acesso a nós just-in-time

O procedimento a seguir descreve como criar políticas de aprovação manual. O Systems Manager permite que você crie até 50 políticas de aprovação manual por Conta da AWS e Região da AWS.

Para criar uma política de aprovação manual

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. Selecione Gerenciar acesso a nós no painel de navegação.
3. Na seção Detalhes da política da etapa Criar política de aprovação manual, insira um nome e uma descrição para a política de aprovação.
4. Insira um valor para a Duração do acesso. Este é o tempo máximo que um usuário pode iniciar sessões em um nó após a aprovação de uma solicitação de acesso. O valor deve ser entre 1 e 336 horas.
5. Na seção Destinos do nó, insira os pares de chave/valor da tag associados aos nós aos quais você deseja que a política se aplique. Se nenhuma das tags especificadas na política estiver associada a um nó, a política não será aplicada ao nó.
6. Na seção Aprovadores de solicitações de acesso, insira os usuários ou grupos que você deseja que possam aprovar solicitações de acesso aos destinos de nós na política. Os aprovadores

de solicitações de acesso podem ser usuários e grupos do Centro de Identidade do IAM ou perfis do IAM. Você pode especificar até cinco aprovadores por nível e até cinco níveis de aprovadores.

7. Selecione Criar política de aprovação manual.

Estrutura de instruções e operadores integrados para políticas de aprovação automática e negação de acesso

A tabela a seguir mostra a estrutura das políticas de aprovação automática e negação de acesso.

Componente	Sintaxe
efeito;	`permit | forbid`
scope	`(principal, action, resource)`
Cláusula de condição	`when { principal or resource has attribute name };`

Componentes da política

Uma política de aprovação automática ou negação de acesso contém os seguintes componentes:

- Efeito: permite (`permit`) ou nega (`forbid`) o acesso.
- Escopo: a entidade principal, as ações e os recursos aos quais o efeito se aplica. Você pode deixar o escopo no Cedar indefinido ao não identificar entidades principais, ações ou recursos específicos. Nesse caso, a política se aplica a todos os principais, ações e recursos possíveis. Para acesso a nós just-in-time, o `action` é sempre `AWS::SSM::Action::"getTokenForInstanceAccess"`.
- Cláusula de condição: o contexto no qual o efeito se aplica.

Comentários

Você pode incluir comentários nas políticas do . Os comentários são definidos como uma linha que começa com `//` e termina com um caractere de nova linha.

O exemplo a seguir mostra comentários em uma política.

```
// Allows users in the Engineering group from the Platform org to automatically connect
 to nodes tagged with Engineering and Production keys.
permit (
    principal in AWS::IdentityStore::Group::"d4q81745-r081-7079-d789-14da1EXAMPLE",
    action == AWS::SSM::Action::"getTokenForInstanceAccess",
    resource
)
when {
    principal has organization && resource.hasTag("Engineering") &&
 resource.hasTag("Production") && principal.organization == "Platform"
};
```

Cláusulas múltiplas

Você pode usar mais de uma cláusula de condição em uma declaração de política usando o operador &&.

```
// Allow access if node has tag where the tag key is Environment
// & tag value is Development

permit(principal, action == AWS::SSM::getTokenForInstanceAccess, resource)
when {
    resource.hasTag("Environment") &&
    resource.getTag("Environment") == "Development"
};
```

Caracteres reservados

O exemplo a seguir mostra como escrever uma política se uma propriedade de contexto usar : (ponto e vírgula), que é um caractere reservado na linguagem da política.

```
permit (
    principal,
    action == AWS::SSM::Action::"getTokenForInstanceAccess",
    resource
)
when {
    principal has employeeNumber && principal.employeeNumber like "E-1*" &&
 resource.hasTag("Purpose") && resource.getTag("Purpose") == "Testing"
```

```
}
```

Para obter exemplos adicionais, consulte Exemplo de instruções de políticas.

Esquema de acesso a nós just-in-time

Confira abaixo o esquema Cedar para acesso a nós just-in-time.

```
namespace AWS::EC2 {
    entity Instance tags String;
}


namespace AWS::IdentityStore {
    entity Group;

    entity User in [Group] {
    employeeNumber?: String,
    costCenter?: String,
    organization?: String,
    division?: String,
    };

}


namespace AWS::IAM {

    entity Role;

    type AuthorizationContext = {
        principalTags: PrincipalTags,
    };

    entity PrincipalTags tags String;
}

namespace AWS::SSM {

    entity ManagedInstance tags String;

    action "getTokenForInstanceAccess" appliesTo {
    principal: [AWS::IdentityStore::User],
```

```
    resource: [AWS::EC2::Instance, AWS::SSM::ManagedInstance],
    context: {
        "iam": AWS::IAM::AuthorizationContext
        }
    };
}
```

Operadores integrados

Ao criar o contexto de uma política de aprovação automática ou negação de acesso usando várias condições, você pode usar o operador && para adicionar outras condições. Há também muitos outros operadores integrados que você pode usar para adicionar mais poder expressivo às condições da sua política. A tabela a seguir contém todos os operadores integrados para referência.

Operador	Tipos e sobrecargas	Descrição
!	Booleano → Booleano	Lógico que não.
==	qualquer → qualquer	Igualdade. Funciona com argumentos de qualquer tipo, mesmo que os tipos não correspondam. Valores de tipos diferentes nunca são iguais entre si.
!=	qualquer → qualquer	Desigualdade; o inverso exato da igualdade (veja acima).
<	(longo, longo) → Booleano	Número inteiro longo menor que.
<=	(longo, longo) → Booleano	Número inteiro longo menor ou igual a.
>	(longo, longo) → Booleano	Número inteiro longo maior que.
>=	(longo, longo) → Booleano	Número inteiro longo maior que ou igual a.

Operador	Tipos e sobrecargas	Descrição
em	(entidade, entidade) → Booleano	Associação hierárquica (reflexiva: A em A é sempre verdadeiro).
	(entidade, conjunto (entidade)) → Booleano	Associação à hierarquia: A em [B, C,...] é verdadeiro se (A e B) || (A em C) ||... erro se o conjunto não contiver uma entidade.
&&	(Booleano, Booleano) → Booleano	Lógico e (curto-circuito).
||	(Booleano, Booleano) → Booleano	Lógico ou (curto-circuito).
.exists()	entidade→ Booleano	Existência de entidades.
tem	(entidade, atributo) → Booleano	Operador infixo. `e has f` testa se o registro ou a entidade `e` tem uma associação para o atributo `f`. Retorna `false` se `e` não existe ou se `e` existe, mas não tem o atributo `f`. Os atributos podem ser expressos como identificadores ou literais de sequência de caracteres.

Operador	Tipos e sobrecargas	Descrição
como	(string, string) → Booleano	Operador infixo. `t like p` verifica se o texto `t` corresponde ao padrão `p`, que pode incluir caracteres curinga `*` que correspondam a 0 ou mais de qualquer caractere. Para combinar literalmente um caractere estrela `t`, você pode usar a sequência `\*` especial de caracteres escapados em `p`.
.hasTag()	(entity, string) → Boolean	Verifica se a entidade tem a tag especificada aplicada.
.getTag()	(entity, string) → Boolean	Retorna o valor na chave de tag especificada.
.contém()	(conjunto, todos) → Booleano	Defina a associação (B é um elemento de A).
.contém tudo()	(conjunto, conjunto) → Booleano	Testa se o conjunto A contém todos os elementos do conjunto B.
.contém qualquer()	(conjunto, conjunto) → Booleano	Testa se o conjunto A contém algum dos elementos do conjunto B.

Exemplo de instruções de políticas

Confira abaixo exemplos de instruções de política.

```
// Users assuming IAM roles with a principal tag of "Elevated" can automatically access
 nodes tagged with the "Environment" key when the value equals "prod"
permit(principal, action == AWS::SSM::getTokenForInstanceAccess, resource)
```

```
when {
    // Verify IAM role principal tag
    context.iam.principalTags.getTag("AccessLevel") == "Elevated" &&

    // Verify the node has a tag with "Environment" tag key and a tag value of "prod"
    resource.hasTag("Environment") &&
    resource.getTag("Environment") == "prod"
};
```

```
// Identity Center users in the "Contractor" division can automatically access nodes
 tagged with the "Environment" key when the value equals "dev"
permit(principal, action == AWS::SSM::getTokenForInstanceAccess, resource)
when {
    // Verify that the user is part of the "Contractor" division
    principal.division == "Contractor" &&

    // Verify the node has a tag with "Environment" tag key and a tag value of "dev"
    resource.hasTag("Environment") &&
    resource.getTag("Environment") == "dev"
};
```

```
// Identity Center users in a specified group can automatically access nodes tagged
 with the "Environment" key when the value equals "Production"
permit(principal in AWS::IdentityStore::Group::"d4q81745-r081-7079-d789-14da1EXAMPLE",
    action == AWS::SSM::getTokenForInstanceAccess,
    resource)
when {
    resource.hasTag("Environment") &&
    resource.getTag("Environment") == "Production"
};
```

Criar uma política de aprovação automática para acesso a nós just-in-time

As políticas de aprovação automática usam a linguagem de política Cedar para definir quais usuários podem se conectar automaticamente aos nós especificados sem aprovação manual. Uma política de aprovação automática contém várias declarações `permit` especificando a `principal` e o `resource`. Cada instrução inclui uma cláusula `when` que define as condições para a aprovação automática.

Confira abaixo um exemplo de política de aprovação automática.

```
permit (
    principal in AWS::IdentityStore::Group::"e8c17310-e011-7089-d989-10da1EXAMPLE",
    action == AWS::SSM::Action::"getTokenForInstanceAccess",
    resource
)
when {
    principal has costCenter && resource.hasTag("CostCenter") && principal.costCenter
 == resource.getTag("CostCenter")
};

permit (
    principal in AWS::IdentityStore::Group::"d4q81745-r081-7079-d789-14da1EXAMPLE",
    action == AWS::SSM::Action::"getTokenForInstanceAccess",
    resource
)
when {
    principal has organization && resource.hasTag("Engineering") &&
 resource.hasTag("Production") && principal.organization == "Platform"
};

permit (
    principal,
    action == AWS::SSM::Action::"getTokenForInstanceAccess",
    resource
)
when {
    principal has employeeNumber && principal.employeeNumber like "E-1*" &&
 resource.hasTag("Purpose") && resource.getTag("Purpose") == "Testing"
};
```

O procedimento a seguir descreve como criar uma política de aprovação automática para acesso a nós just-in-time. A duração do acesso para uma solicitação de acesso aprovada automaticamente é de uma hora. Esse valor não pode ser alterado. Você só pode ter uma política de aprovação automática por Conta da AWS e Região da AWS. Para obter informações sobre como criar instruções de política, consulte Estrutura de instruções e operadores integrados para políticas de aprovação automática e negação de acesso.

Para criar uma política de aprovação automática

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. Selecione Gerenciar acesso a nós no painel de navegação.

3. Na guia Políticas de aprovação, selecione Criar uma política de aprovação automática.
4. Insira sua instrução de política para a política de aprovação automática na seção Instrução de política. Você pode usar as Instruções de exemplo fornecidas para ajudar a criar sua política.
5. Selecione Criar política de aprovação automática.

Criar uma política de negação de acesso para acesso a nós just-in-time

As políticas de negação de acesso usam a linguagem de política Cedar para definir a quais nós os usuários não podem se conectar automaticamente sem aprovação manual. Uma política de negação de acesso contém várias instruções `forbid` especificando a `principal` e o `resource`. Cada instrução inclui uma cláusula `when` que define as condições para negar explicitamente a aprovação automática.

Confira abaixo um exemplo de política de negação de acesso.

```
forbid (
    principal in AWS::IdentityStore::Group::"e8c17310-e011-7089-d989-10da1EXAMPLE",
    action == AWS::SSM::Action::"getTokenForInstanceAccess",
    resource
)
when {
    resource.hasTag("Environment") && resource.getTag("Environment") == "Production"
};

forbid (
    principal,
    action == AWS::SSM::Action::"getTokenForInstanceAccess",
    resource
)
when {
    principal has division && principal.division != "Finance" &&
 resource.hasTag("DataClassification") && resource.getTag("DataClassification") ==
 "Financial"
};


forbid (
    principal,
    action == AWS::SSM::Action::"getTokenForInstanceAccess",
    resource
)
```

```
when {

    principal has employeeNumber && principal.employeeNumber like "TEMP-*" &&
 resource.hasTag("Criticality") && resource.getTag("Criticality") == "High"
};
```

O procedimento a seguir descreve como criar uma política de negação de acesso para acesso a nós just-in-time. Para obter informações sobre como criar instruções de política, consulte Estrutura de instruções e operadores integrados para políticas de aprovação automática e negação de acesso.

> **Note**
>
> Observe as seguintes informações:
>
> - Você poderá criar políticas de negação de acesso enquanto estiver conectado à conta de gerenciamento da AWS ou à conta de administrador delegado. Sua organização do AWS Organizations pode ter apenas uma política de negação de acesso.
> - O acesso ao nó just-in-time faz uso do AWS Resource Access Manager (AWS RAM) para compartilhar sua política de negação de acesso com as contas de membros na sua organização. Se você quiser compartilhar sua política de negação de acesso com as contas de membros da sua organização, o compartilhamento de recursos deve ser habilitado na conta gerencial da sua organização. Para obter mais informações, consulte Habilitar o compartilhamento de recursos no AWS Organizations no Guia do usuário do AWS RAM.

Para criar uma política de negação de acesso

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. Selecione Gerenciar acesso a nós no painel de navegação.
3. Na guia Políticas de aprovação, selecione Criar uma política de negação de acesso.
4. Insira sua instrução de política para a política de negação de acesso na seção Instrução de política. Você pode usar as Instruções de exemplo fornecidas para ajudar a criar sua política.
5. Selecione Criar política de negação de acesso.

Criar políticas de aprovação para acesso a nós just-in-time com o Amazon Q

O uso do Amazon Q Developer para a linha de comando fornece orientação e suporte em vários aspectos do desenvolvimento de software. Para acesso a nós just-in-time, o Amazon Q ajuda você a criar políticas de aprovação gerando e atualizando o código das políticas, analisando declarações de políticas e muito mais. As informações a seguir descrevem como criar políticas de aprovação usando o Amazon Q para a linha de comando.

Identifique seu caso de uso

A primeira etapa na criação de políticas de aprovação é definir claramente o caso de uso. Por exemplo, na sua organização, talvez você queira aprovar automaticamente as solicitações de acesso a nós com uma tag `Environment:Testing`. Talvez você também queira negar explicitamente as aprovações automáticas aos nós com uma tag `Environment:Production` se o ID do funcionário começar com `TEMP`. Para nós com uma tag `Tier:Database`, talvez você queira exigir dois níveis de aprovações manuais.

Em qualquer cenário específico, você pode preferir uma política ou condição em vez de outra. Portanto, recomendamos que você defina claramente os comportamentos da política que deseja para determinar quais instruções melhor se adaptam ao seu caso de uso e preferências.

Configuração do ambiente de desenvolvimento

Instale o Amazon Q para a linha de comando em que você deseja desenvolver suas políticas de aprovação. Para obter informações sobre a instalação do Amazon Q para a linha de comando, consulte Installing Amazon Q for command line no Guia do usuário do Amazon Q Developer.

Também recomendamos instalar o servidor MCP da documentação da AWS. Esse servidor MCP conecta o Amazon Q da linha de comando aos recursos de documentação mais atuais. Para obter informações sobre o uso do MCP com o Amazon Q para a linha de comando, consulte Using MCP with Amazon Q Developer no Guia do usuário do Amazon Q Developer.

Para obter mais informações sobre o servidor MCP da documentação da AWS, consulte AWS Documentation MCP Server.

Instale e configure a AWS CLI, caso ainda não tenha feito isso. Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

Desenvolver o conteúdo da política de aprovação

Com o caso de uso identificado e o ambiente configurado, você está pronto para desenvolver o conteúdo das políticas. Seu caso de uso e suas preferências ditarão em grande parte os tipos de políticas e declarações de aprovação que você usa.

Caso não tenha certeza de como usar uma política específica ou precise de mais informações sobre o esquema de uma política, consulte Criar políticas de aprovação para os nós e os tópicos subsequentes. Esses tópicos explicam em detalhes como as políticas são avaliadas e fornecem exemplos práticos de instruções.

O procedimento a seguir descreve como criar políticas de aprovação com o Amazon Q da linha de comando.

> **Note**
>
> O Amazon Q Developer usa IA generativa. Talvez seja necessário verificar as respostas. Consulte a Política de uso responsável de IA da AWS.

Para criar políticas de aprovação usando o Amazon Q da linha de comando

1. Abra seu terminal e inicie uma sessão com o Amazon Q.

```
$ q chat
```

2. Insira um prompt e descreva o tipo de instrução de política que você deseja criar. Por exemplo, usando o cenário anterior para uma política de aprovação, você pode inserir o seguinte prompt:

```
Create a Cedar policy statement that allows a principal to use the
 “getTokenForInstanceAccess" action for a resource
that has a tag key “Environment” and tag value of “Testing”. Use the following
 example statement for reference to verify the syntax.

permit (
    principal in AWS::IdentityStore::Group::"e8c17310-e011-7089-d989-10da1EXAMPLE",
    action == AWS::SSM::Action::"getTokenForInstanceAccess",
    resource
)
when {
```

```
    principal has organization && resource.hasTag("Engineering") &&
 resource.hasTag("Production") && principal.organization == "Platform"
};

The Cedar policy statement you create must be valid. Validate the Cedar statement
 using the AWS documentation provided for just-in-time node access policy statement
 structure and make sure you abide by the schema.
```

3. A inserção desse prompt de exemplo resultou na seguinte saída:

```
Based on the AWS documentation for Cedar policy statements and the examples
 provided, I'll create a valid Cedar policy
statement that allows a principal to use the "getTokenForInstanceAccess" action for
 resources with the tag key "Environment" and tag value "Testing".

Here's the Cedar policy statement:

permit (
    principal,
    action == AWS::SSM::Action::"getTokenForInstanceAccess",
    resource
)
when {
    resource.hasTag("Environment") &&
    resource.getTag("Environment") == "Testing"
};


This policy statement:
1. Uses the permit effect to allow access
2. Applies to any principal (since no specific principal is defined)
3. Specifically targets the "getTokenForInstanceAccess" action
4. Applies to resources that have a tag with key "Environment" and value "Testing"
5. Uses the proper syntax for the condition clause with the hasTag() and getTag()
 operators

The statement follows the Cedar policy language structure as documented in the AWS
 Systems Manager documentation for just-in-time node access policies.
```

4. Cole a instrução de política em um novo arquivo Cedar para validar a sintaxe. Você também pode validar a instrução usando o playground Cedar.
5. Depois de validar a instrução, salve o arquivo Cedar.

6. No seu terminal, execute o comando da AWS CLI a seguir para criar um documento do SSM usando o conteúdo da sua política. Antes de usar políticas de aprovação em um ambiente de produção, teste suas políticas de aprovação com um subconjunto de identidades e nós em uma Conta da AWS e Região da AWS.

> **Note**
>
> Para uma política de aprovação automática, o nome do documento deve ser `SSM-JustInTimeAccessAutoApprovalPolicy`. Só pode haver uma política de aprovação automática por Conta da AWS e Região da AWS. Para uma política de negação de acesso, o nome do documento deve ser `SSM-JustInTimeAccessDenyAccessOrgPolicy`. Só pode haver uma política de negação de acesso por organização do AWS Organizations, e a política deve ser criada na conta de administrador delegado do Systems Manager. As restrições de nomenclatura das políticas de aprovação manual são as mesmas de outros documentos do SSM. Para obter mais informações, consulte CreateDocument.

Linux & macOS

```
aws ssm create-document \
    --content file://path/to/file/policyContent.cedar \
    --name "SSM-JustInTimeAccessAutoApprovalPolicy" \
    --document-type "AutoApproval"
```

Windows

```
aws ssm create-document ^
    --content file://C:\path\to\file\policyContent.cedar ^
    --name "SSM-JustInTimeAccessAutoApprovalPolicy" ^
    --document-type "AutoApproval"
```

PowerShell

```
$cedar = Get-Content -Path "C:\path\to\file\policyContent.cedar" | Out-String
New-SSMDocument `
    -Content $cedar `
    -Name "SSM-JustInTimeAccessAutoApprovalPolicy" `
```

```
    -DocumentType "AutoApproval"
```

Atualizar as preferências da sessão de acesso a nós just-in-time

Com o acesso a nós just-in-time, você pode especificar as preferências gerais de sessão e registro em log em cada Conta da AWS e Região da AWS em sua organização. Como alternativa, você pode usar o AWS CloudFormation StackSets para criar um documento de preferências de sessão em várias contas e regiões para ajudar você a ter preferências de sessão consistentes. Para obter informações sobre o esquema para documentos de preferências de sessão, consulte Esquema do documento de sessão.

Para fins de registro em log, recomendamos usar a opção de streaming com o Amazon CloudWatch Logs. Esse recurso permite que você envie uma transmissão contínua de logs de dados de sessão ao Amazon CloudWatch Logs. Detalhes essenciais, como os comandos que um usuário executou em uma sessão, o ID do usuário que executou os comandos e carimbos de data/hora para quando os dados da sessão são transmitidos para o CloudWatch Logs, são incluídos ao transmitir dados da sessão. Ao transmitir dados de sessão, os logs são formatados em JSON para ajudar você a integrar com suas soluções de log existentes.

O Systems Manager não encerra automaticamente as sessões de acesso a nós just-in-time. Como prática recomendada, especifique os valores para as configurações de duração máxima da sessão e tempo limite de sessão ociosa. O uso dessas configurações ajuda a evitar que um usuário permaneça conectado a um nó por mais tempo do que a janela de tempo aprovada em uma solicitação de acesso. O procedimento a seguir descreve como atualizar as preferências de sessão para acesso a nós just-in-time.

> **Important**
>
> É necessário marcar com tags as chaves do AWS KMS usadas para criptografia e gravação RDP no acesso a nós just-in-time no Session Manager com a chave de tag `SystemsManagerJustInTimeNodeAccessManaged` e o valor de tag `true`.
> Para obter mais informações sobre tags de chaves do KMS, consulte Tags no AWS KMS no Guia do desenvolvedor do AWS Key Management Service.

Para atualizar as preferências da sessão

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.

2. Selecione Configurações no painel de navegação.
3. Selecione a guia Acesso a nós just-in-time.
4. Na seção Preferências da sessão, selecione Editar.
5. Atualize suas preferências gerais e de registro em log conforme necessário e selecione Salvar.

Configurar notificações para solicitações de acesso just-in-time

Você pode configurar o Systems Manager para enviar notificações quando um usuário criar uma solicitação de acesso a nós just-in-time para os endereços de e-mail ou cliente de chat dos aprovadores e do solicitante. A notificação contém o motivo da solicitação de acesso fornecido pelo solicitante, a Conta da AWS, a Região da AWS, o status da solicitação e o ID do nó de destino. Atualmente, o Systems Manager oferece suporte aos clientes do Slack e do Microsoft Teams por meio da integração com o Amazon Q Developer em aplicações de chat. Ao usar notificações por meio de clientes de chat, os aprovadores de solicitações de acesso podem interagir diretamente com as solicitações de acesso. Isso elimina a necessidade de fazer login no console para executar ações em relação às solicitações de acesso.

Antes de começar

Antes de configurar um cliente de chat para notificações de acesso a nós just-in-time, observe o seguinte requisito:

- Caso esteja usando perfis do IAM para gerenciar identidades de usuários em sua conta, você deverá associar manualmente os endereços de e-mail dos aprovadores ou solicitantes aos quais deseja enviar notificações ao perfil associado. Caso contrário, os destinatários pretendidos não poderão ser notificados por e-mail.

Os procedimentos a seguir descrevem como configurar notificações para solicitações de acesso a nós just-in-time.

Para configurar um cliente de chat para notificações de acesso a nós just-in-time

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. Selecione Configurações no painel de navegação.
3. Selecione a guia Acesso a nós just-in-time.
4. Na seção Chat, selecione Configurar novo cliente.

5. No menu suspenso Selecionar tipo de cliente, escolha o tipo de cliente de chat que você deseja configurar e selecione Próximo.
6. Você será solicitado a permitir que o Amazon Q Developer em aplicações de chat acesse seu cliente de chat. Selecione Permitir.
7. Na seção Configurar canal, insira as informações do canal do cliente de chat e selecione os tipos de notificações que você deseja receber.
8. Se você estiver configurando as notificações do Slack, convide "@Amazon Q" para cada canal do Slack em que as notificações estão sendo configuradas.
9. Selecione Configurar canal.

> **Note**
>
> Para permitir a aprovação/rejeição de solicitações de acesso diretamente de um canal do Slack, certifique-se de que o perfil do IAM configurado com o canal do Slack tenha permissões `ssm:SendAutomationSignal` e tenha uma política de confiança que inclua o chatbot:
>
> ```
> {
>             "Effect": "Allow",
>             "Principal": {
>                 "Service": "chatbot.amazonaws.com"
>             },
>             "Action": "sts:AssumeRole"
> }
> ```

Para configurar notificações por e-mail para notificações de acesso a nós just-in-time

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. Selecione Configurações no painel de navegação.
3. Selecione a guia Acesso a nós just-in-time.
4. Na seção E-mail, selecione Editar.
5. Selecione Adicionar e-mails, escolha o perfil do IAM ao qual você deseja associar manualmente os endereços de e-mail.

6. No campo Endereço de e-mail, insira um endereço de e-mail. Sempre que uma solicitação de acesso é criada e exige aprovação do perfil do IAM que você especificou, os endereços de e-mail que você associou ao perfil são notificados.
7. Selecione Adicionar endereço de e-mail.

Gravar conexões RDP

O acesso a nós just-in-time inclui a capacidade de gravar conexões RDP realizadas com seus nós do Windows Server. A gravação de conexões RDP requer um bucket do S3 e uma chave do AWS Key Management Service (AWS KMS) gerenciada pelo cliente. A AWS KMS key é usada para criptografar temporariamente os dados de gravação enquanto são gerados e armazenados nos recursos do Systems Manager. A chave gerenciada pelo cliente deve ser uma chave simétrica com o uso da chave de criptografia e descriptografia. Você pode usar uma chave multirregional para sua organização, ou você deve criar uma chave gerenciada pelo cliente em cada região onde você habilitou o acesso a nós just-in-time.

Se você habilitou a criptografia do KMS no bucket do S3 no qual armazena gravações, será necessário fornecer acesso à chave gerenciada pelo cliente utilizada para a criptografia de buckets à entidade principal de serviço `ssm-guiconnect`. Essa chave gerenciada pelo cliente pode ser diferente da especificada nas configurações de gravação, que precisa incluir a permissão `kms:CreateGrant` que é necessária para o estabelecimento de conexões.

Configurar a criptografia de buckets do S3 para gravações de RDP

Suas gravações de conexão são armazenadas no bucket do S3 especificado ao habilitar a gravação de RDP.

Se você utilizar uma chave do KMS como mecanismo de criptografia padrão para o bucket do S3 (SSE-KMS), deverá permitir que a entidade principal de serviço `ssm-guiconnect` acesse a ação `kms:GenerateDataKey` nessa chave. Recomendamos o uso de uma chave gerenciada pelo cliente ao usar a criptografia SSE-KMS com um bucket do S3. Isso porque é possível atualizar a política de chave associada de uma chave gerenciada pelo cliente. Não é possível atualizar as políticas de chave para Chaves gerenciadas pela AWS.

> **Important**
>
> É necessário marcar com tags as chaves do AWS KMS usadas para criptografia e gravação RDP no acesso a nós just-in-time no Session Manager com a chave de tag `SystemsManagerJustInTimeNodeAccessManaged` e o valor de tag `true`.

Para obter mais informações sobre tags de chaves do KMS, consulte Tags no AWS KMS no Guia do desenvolvedor do AWS Key Management Service.

Use a política de chave gerenciada pelo cliente a seguir para permitir que o serviço `ssm-guiconnect` acesse a chave do KMS para armazenamento no S3. Para obter informações sobre como modificar uma chave gerenciada pelo cliente, consulte Alterar uma política de chave no Guia do desenvolvedor do AWS Key Management Service.

Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações:

- *account-id* representa o ID da Conta da AWS que inicia a conexão.
- *region* representa a Região da AWS em que o bucket do S3 está localizado. (Você poderá usar * se o bucket for receber gravações de várias regiões. Exemplo: `s3.*.amazonaws.com`).

Note

Você poderá usar `aws:SourceOrgID` na política em vez de `aws:SourceAccount` se a conta pertencer a uma organização no AWS Organizations.

```
{
    "Sid": "Allow the GUI Connect service principal to access S3",
    "Effect": "Allow",
    "Principal": {
        "Service": "ssm-guiconnect.amazonaws.com"
    },
    "Action": [
        "kms:GenerateDataKey*"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": "account-id"
        },
        "StringLike": {
            "kms:ViaService": "s3.region.amazonaws.com"
        }
    }
```

```
}
```

Configurar permissões do IAM para gravar conexões RDP

Além das permissões do IAM necessárias para o acesso a nós just-in-time, o usuário ou perfil que você utilizar deverá ter as permissões a seguir com base na tarefa que você precisará realizar.

Permissões para configurar a gravação de conexão

Para configurar o registro de conexões RDP, as seguintes permissões são necessárias:

- `ssm-guiconnect:UpdateConnectionRecordingPreferences`
- `ssm-guiconnect:GetConnectionRecordingPreferences`
- `ssm-guiconnect:DeleteConnectionRecordingPreferences`
- `kms:CreateGrant`

Permissões para iniciar conexões

Para fazer conexões RDP com o acesso a nós just-in-time, as seguintes permissões são necessárias:

- `ssm-guiconnect:CancelConnection`
- `ssm-guiconnect:GetConnection`
- `ssm-guiconnect:StartConnection`
- `kms:CreateGrant`

Antes de começar

Para armazenar as gravações de conexão, primeiro você deve criar um bucket do S3 e adicionar política de bucket a seguir. Substitua cada *`espaço reservado para recurso de exemplo`* por suas próprias informações.

(Para obter mais informações sobre como adicionar uma política de bucket, consulte Adicionar uma política de bucket usando o console do Amazon S3 no Guia do usuário do Amazon Simple Storage Service.)

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ConnectionRecording",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "ssm-guiconnect.amazonaws.com"
                ]
            },
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::",
                "arn:aws:s3:::/*"
            ],
            "Condition":{
            "StringEquals":{
                "aws:SourceAccount":"111122223333"
                }
            }
        }
    ]
}
```

Habilitar e configurar a gravação de conexões RDP

O procedimento a seguir descreve como habilitar e configurar o registro de conexões RDP.

Para habilitar e configurar a gravação de conexões RDP

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. Selecione Configurações no painel de navegação.
3. Selecione a guia Acesso a nós just-in-time.
4. Na seção Gravação RDP, selecione Habilitar gravação RDP.
5. Escolha o bucket do S3 no qual você deseja fazer upload das gravações da sessão.

6. Escolha a chave gerenciada pelo cliente que você deseja usar para criptografar temporariamente os dados de gravação enquanto eles são gerados e armazenados nos recursos do Systems Manager. (Pode ser uma chave gerenciada pelo cliente diferente da que você utiliza para criptografar o bucket.)
7. Selecione Salvar.

Valores de status de gravação de conexões RDP

Os valores de status válidos para gravações de conexões RDP incluem o seguinte:

- `Recording`: a conexão está em processo de ser gravada
- `Processing`: o vídeo está sendo processado após o encerramento da conexão.
- `Finished`: estado de terminal com êxito: o vídeo de gravação da conexão foi processado com êxito e carregado no bucket especificado.
- `Failed`: estado de terminal com falha. A conexão não foi gravada com êxito.
- `ProcessingError`: uma ou mais falhas/erros intermediários ocorreram durante o processamento do vídeo. Possíveis causas incluem falhas na dependência de serviços ou permissões ausentes em decorrência de uma configuração incorreta no bucket do S3 especificado para o armazenamento de gravações. O serviço continua a fazer tentativasd de processamento quando a gravação se encontra nesse estado.

> **Note**
>
> `ProcessingError` pode ocorrer porque a entidade principal de serviço `ssm-guiconnect` não tem permissão para fazer upload de objetos no bucket do S3 depois que a conexão é estabelecida. Outra causa possível é a ausência de permissões do KMS na chave do KMS utilizada para a criptografia do bucket do S3.

Modificar destinos

Ao configurar o acesso a nós just-in-time, você escolhe os destinos em que deseja configurar o acesso a nós just-in-time. Os destinos consistem em unidades organizacionais (UOs) do AWS Organizations e Regiões da AWS. Por padrão, os mesmos destinos que você escolheu ao configurar o console unificado do Systems Manager são selecionados para acesso a nós just-in-time. Você pode optar por configurar o acesso a nós just-in-time para todos os mesmos destinos ou para

um subconjunto dos destinos que você especificou ao configurar o console unificado do Systems Manager. Não há suporte para a adição de novos destinos que não foram selecionados quando você configurou o console unificado do Systems Manager. Você pode alterar os destinos selecionados após configurar o acesso a nós just-in-time.

O procedimento a seguir descreve como modificar os destinos para acesso a nós just-in-time.

Para modificar destinos

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. Selecione Configurações no painel de navegação.
3. Selecione a guia Acesso a nós just-in-time.
4. Na seção Destinos, selecione Editar.
5. Selecione as Unidades organizacionais e as Regiões em que você deseja usar o acesso a nós just-in-time.
6. Selecione Salvar.

Alterar provedores de identidade

Por padrão, o acesso a nós just-in-time usa o IAM como provedor de identidade. Depois de habilitar o acesso a nós just-in-time, os clientes que usam o console unificado com uma organização podem modificar essa configuração para usar o Centro de Identidade do IAM. O acesso a nós just-in-time não é compatível com o Centro de Identidade do IAM como provedor de identidade quando configurado para uma única conta e região.

O procedimento a seguir descreve como modificar o provedor de identidade para acesso a nós just-in-time.

Para modificar provedores de identidade

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. Selecione Configurações no painel de navegação.
3. Selecione a guia Acesso a nós just-in-time.
4. Na seção Identidade do usuário, selecione Editar.
5. Escolha Habilitar o Centro de Identidade do AWS IAM.
6. Selecione Salvar.

Iniciar uma sessão de acesso a nós just-in-time

Depois de habilitar e definir o acesso a nós just-in-time e configurar as preferências de sessão e notificação, os usuários estarão prontos para iniciar as sessões de acesso a nós just-in-time. Você pode iniciar sessões usando o acesso a nós just-in-time no console do Systems Manager ou da AWS Command Line Interface usando o plug-in do Session Manager. As sessões de acesso a nós just-in-time podem ser iniciadas em nós na mesma conta e região. Os procedimentos a seguir descrevem como iniciar sessões com acesso a nós just-in-time.

> **Note**
>
> Se os seus usuários usavam anteriormente o Session Manager para se conectar aos nós, você deve remover as permissões do Session Manager, por exemplo `ssm:StartSession`, das políticas do IAM para iniciar sessões usando o acesso a nós just-in-time. Do contrário, ao se conectar aos nós, eles continuarão a usar o Session Manager.

Para iniciar uma sessão com acesso a nós just-in-time usando o console

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, selecione Explorar nós.
3. Selecione o nó ao qual deseja se conectar.
4. No menu suspenso Ações, selecione Conectar.

Se as políticas de aprovação da sua organização não permitirem que você se conecte automaticamente ao nó, você será solicitado a enviar uma solicitação de acesso. Depois de preencher as informações solicitadas e enviar a solicitação de acesso, você poderá iniciar as sessões no nó assim que todas as aprovações necessárias forem recebidas.

Para iniciar uma sessão com acesso a nós just-in-time usando a AWS CLI

1. Execute o comando a seguir para iniciar o fluxo de trabalho da solicitação de acesso, certificando-se de substituir os *valores de espaço reservado* por suas próprias informações.

```
aws ssm start-access-request \
    --targets  Key=InstanceIds,Values=i-02573cafcfEXAMPLE
```

```
    --reason "Troubleshooting networking performance issue"
```

Dependendo das políticas de aprovação da sua organização, você será conectado automaticamente ao nó ou o processo de aprovação manual será iniciado. Para solicitações que exigem aprovações manuais, anote o ID da solicitação de acesso que é retornado na resposta.

2. Aguarde até que todas as aprovações necessárias sejam fornecidas.
3. Depois que todas as aprovações necessárias tiverem sido fornecidas, execute o comando a seguir para obter um token de acesso contendo credenciais temporárias. Substitua os *valores de espaço reservado* por suas próprias informações.

```
aws ssm get-access-token \
    --access-request-id oi-12345abcdef
```

Anote o token de acesso retornado na resposta.

4. Execute o comando a seguir para usar a credencial temporária na AWS CLI, certificando-se de substituir os *valores de espaço reservado* por suas próprias informações.

```
export AWS_SESSION_TOKEN=AQoDYXdzEJr...<remainder of session token>
```

5. Execute o comando a seguir para iniciar uma sessão no nó, certificando-se de substituir os *valores de espaço reservado* por suas próprias informações.

```
aws ssm start-session \
    --target i-02573cafcfEXAMPLE
```

Gerenciar solicitações de acesso just-in-time

Para maior visibilidade em toda a organização, o Systems Manager replica as solicitações de acesso para a conta de administrador delegado da sua organização. Para ajudar você a atender aos requisitos de conformidade, o Systems Manager retém todas as solicitações de acesso por um ano. Os tópicos a seguir descrevem como gerenciar solicitações de acesso a nós just-in-time. Essas informações destinam-se aos aprovadores de solicitações de acesso. Antes de começar, recomendamos revisar suas políticas do IAM e garantir que você tenha as permissões necessárias para administrar o acesso a nós just-in-time. Para obter mais informações, consulte, Configurar o acesso just-in-time com o Systems Manager.

Tópicos

- Aprovar e negar solicitações de acesso a nós just-in-time

Aprovar e negar solicitações de acesso a nós just-in-time

Os aprovadores de solicitações de acesso podem aprovar ou negar solicitações de acesso a nós just-in-time no console unificado do Systems Manager, ou podem usar a ferramenta de linha de comando preferencial. Essas informações destinam-se aos aprovadores de solicitações de acesso. Caso não tenha as permissões necessárias para aprovar ou rejeitar solicitações de acesso, entre em contato com o administrador. Os procedimentos a seguir descrevem como aprovar ou negar solicitações de acesso a nós just-in-time.

Para aprovar ou negar solicitações de acesso a nós just-in-time usando o console

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. Selecione Gerenciar acesso a nós no painel de navegação.
3. Selecione a guia Solicitações de acesso.
4. Selecione a alternância Solicitações para mim.
5. Marque a caixa de seleção ao lado da solicitação de acesso que deseja aprovar ou negar.
6. Selecione Aprovar ou Rejeitar.

Depois de aprovar uma solicitação de acesso, você pode revogar sua aprovação a qualquer momento selecionando Revogar.

Para aprovar ou negar solicitações de acesso a nós just-in-time usando o console

1. Anote o ID da solicitação de acesso da notificação. Por exemplo, *oi-12345abcdef*.
2. Execute o comando a seguir para retornar detalhes sobre o fluxo de trabalho de aprovação da solicitação de acesso, certificando-se de substituir os *valores do espaço reservado* por suas próprias informações.

```
aws ssm get-ops-item \
    --ops-item-id oi-12345abcdef
```

   Observe o valor `automationExecutionId` no campo `/aws/accessrequest` para `OperationalData`. Por exemplo, *9231944f-61c6-40be-8bce-8ee2bEXAMPLE*.

3. Execute o comando a seguir para aprovar ou negar a solicitação de acesso. Use o tipo de sinal `Approve` para aprovar a solicitação e `Deny` para negar a solicitação. Certifique-se de substituir os *valores do espaço reservado* por suas próprias informações.

```
aws ssm send-automation-signal \
    --automation-execution-id 9231944f-61c6-40be-8bce-8ee2bEXAMPLE \
    --signal-type "Approve"
```

Migrar para o acesso a nós just-in-time do Session Manager

Quando você habilita o acesso a nós just-in-time, o Systems Manager não faz nenhuma alteração em seus recursos existentes do Session Manager. Isso garante que não haja interrupções em seu ambiente atual e que os usuários possam continuar iniciando sessões enquanto você cria e valida políticas de aprovação. Quando tudo estiver pronto para você testar as políticas de aprovação, você deve modificar as políticas existentes do IAM para concluir a transição para o acesso a nós just-in-time. Isso inclui adicionar as permissões necessárias para o acesso a nós just-in-time às identidades e remover a permissão da operação da API `StartSession` do Session Manager. Recomendamos testar as políticas de aprovação com um subconjunto de identidades e nós em uma Conta da AWS e Região da AWS.

Para obter mais informações sobre as permissões necessárias para o acesso a nós just-in-time, consulte Configurar o acesso just-in-time com o Systems Manager.

Para obter mais informações sobre como modificar e as permissões do IAM da identidade, consulte Adicionar e remover permissões de identidade do IAM no Guia do usuário do IAM.

A seguir, descrevemos um método detalhado de como você pode migrar para o acesso a nós just-in-time do Session Manager.

Migrar do Session Manager para o acesso a nós just-in-time exige planejamento e testes cuidadosos para garantir uma transição tranquila sem interromper suas operações. As seções a seguir descrevem como você pode concluir esse processo.

Pré-requisitos

Antes de começar, certifique-se de ter concluído as seguintes tarefas:

- Configure o console unificado do Systems Manager.
- Verifique se você tem permissões para modificar as políticas do IAM em sua conta.

- Identificou todas as políticas e perfis do IAM que atualmente concedem permissões do Session Manager.
- Documentou sua configuração atual do Session Manager, incluindo preferências de sessão e configurações de registros em log.

Avaliação

Avalie seu ambiente atual e descreva os comportamentos de aprovação desejados concluindo as seguintes tarefas:

1. Faça o inventário de seus nós: identifique todos os nós que os usuários acessam atualmente por meio do Session Manager.
2. Identifique padrões de acesso do usuário: documente quais usuários ou perfis precisam acessar quais nós e sob quais circunstâncias.
3. Mapeie fluxos de trabalho de aprovação: determine quem deve aprovar solicitações de acesso para diferentes tipos de nós.
4. Revise a estratégia de marcação: garanta que seus nós estejam devidamente marcados para apoiar suas políticas de aprovação planejadas.
5. Audite as políticas existentes do IAM: identifique todas as políticas que incluem permissões do Session Manager.

Planejamento

Estratégia em fases

Ao migrar do Session Manager para o acesso a nós just-in-time, recomendamos usar uma abordagem em fases, como a seguinte:

1. Fase 1: configuração: habilite o acesso a nós just-in-time sem modificar as permissões existentes do Session Manager.
2. Fase 2: desenvolvimento de políticas: crie e teste políticas de aprovação para seus nós.
3. Fase 3: migração piloto: modifique um pequeno grupo de nós não críticos e usuários ou perfis do Session Manager para o acesso a nós just-in-time.
4. Fase 4: migração completa: migre gradualmente todos os nós, usuários ou perfis restantes.

Considerações sobre o cronograma

Considere os seguintes fatores ao criar seu cronograma para migrar do Session Manager para o acesso a nós just-in-time:

- Reserve um tempo para o treinamento do usuário e a adaptação ao novo fluxo de trabalho de aprovação.
- Agende as migrações durante períodos de menor atividade operacional.
- Inclua tempo de buffer para a solução de problemas e ajustes.
- Planeje um período de operação paralela em que ambos os sistemas estejam disponíveis.

Etapas de implementação

Fase 1: configuração

1. Habilite o acesso a nós just-in-time no console do Systems Manager. Para obter detalhes das etapas, consulte, Configurar o acesso just-in-time com o Systems Manager.
2. Configure as preferências da sessão para acesso a nós just-in-time de acordo com suas configurações atuais do Session Manager. Para obter mais informações, consulte Atualizar as preferências da sessão de acesso a nós just-in-time.
3. Configure as preferências de notificação das solicitações de acesso. Para obter mais informações, consulte Configurar notificações para solicitações de acesso just-in-time.
4. Se você usar conexões RDP para nós do Windows Server, configure a gravação RDP. Para obter mais informações, consulte Gravar conexões RDP.

Fase 2: desenvolvimento de políticas

1. Crie políticas do IAM para administradores e usuários de acesso a nós just-in-time.
2. Desenvolva políticas de aprovação com base nos seus requisitos de segurança e no seu caso de uso.
3. Teste suas políticas em um ambiente de não produção para garantir que funcionem conforme o esperado.

Fase 3: migração piloto

1. Selecione um pequeno grupo de usuários e nós não críticos para o piloto.

2. Crie novas políticas do IAM para usuários piloto que incluam permissões de acesso a nós just-in-time.
3. Remova as permissões do Session Manager (`ssm:StartSession`) das políticas do IAM dos usuários piloto.
4. Treine os usuários piloto no novo fluxo de trabalho de solicitação de acesso.
5. Monitore o piloto em busca de problemas e colete feedback.
6. Ajuste as políticas e procedimentos com base nos resultados do piloto.

Exemplo de modificação da política do IAM para usuários piloto

Política original com permissões do Session Manager:

JSON

```
{
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Action": [
                                "ssm:StartSession",
                                "ssm:ResumeSession",
                                "ssm:TerminateSession"
                            ],
                            "Resource": "*"
                        }
                    ]
                }
```

Política modificada para acesso a nós just-in-time:

JSON

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

```
                "Effect": "Allow",
                "Action": [
                    "ssm:StartAccessRequest",
                    "ssm:GetAccessToken",
                    "ssm:ResumeSession",
                    "ssm:TerminateSession"
                ],
                "Resource": "*"
            }
        ]
    }
```

Fase 4: migração completa

Desenvolva um cronograma para migrar os usuários e nós restantes em lotes.

Metodologia de testes

Durante todo o processo de migração, realize os seguintes testes:

- Validação da política: verifique se as políticas de aprovação se aplicam corretamente aos nós e usuários pretendidos.
- Fluxo de trabalho da solicitação de acesso: teste o fluxo de trabalho completo, desde a solicitação de acesso até o estabelecimento da sessão, para cenários de aprovação automática e aprovação manual.
- Notificações: verifique se os aprovadores recebem notificações pelos canais configurados (e-mail, Slack, Microsoft Teams).
- Registro em log e monitoramento: verifique se os logs da sessão e as solicitações de acesso são capturados e armazenados adequadamente.

Práticas recomendadas para uma migração com êxito

- Comunique com antecedência e frequência: informe os usuários sobre o cronograma de migração e os benefícios do acesso a nós just-in-time.
- Comece com sistemas não críticos: inicie a migração com ambientes de desenvolvimento ou de testes antes de passar para o de produção.
- Documente tudo: mantenha registros detalhados de suas políticas de aprovação, alterações na política do IAM e definições de configuração.

- Monitore e ajuste: monitore continuamente as solicitações de acesso e os fluxos de trabalho de aprovação, ajustando as políticas conforme necessário.
- Estabeleça a governança: crie um processo para revisar e atualizar regularmente as políticas de aprovação à medida que seu ambiente muda.

Desabilitar o acesso just-in-time com o Systems Manager

O procedimento a seguir descreve como desabilitar o acesso a nós just-in-time. Depois de desabilitar o acesso a nós just-in-time, os usuários da sua organização talvez não consigam se conectar aos seus nós, a menos que você já tenha implementado outros métodos de conexão.

Para desabilitar o acesso a nós just-in-time

1. Faça login na conta de administrador delegado do Systems Manager da sua organização.
2. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
3. Selecione Configurações no painel de navegação.
4. Na guia Acesso a nós just-in-time, selecione Desabilitar.

Perguntas frequentes sobre acesso a nós just-in-time

Como faço para migrar do Session Manager para o acesso a nós just-in-time?

Depois de configurar o console unificado e habilitar o acesso a nós just-in-time, você deve modificar as políticas existentes do IAM para concluir a migração para o acesso a nós just-in-time. Isso inclui adicionar as permissões necessárias para o acesso a nós just-in-time e remover a permissão para a operação da API `StartSession` do Session Manager. Para obter mais informações sobre as políticas do IAM para acesso a nós just-in-time, consulte Configurar o acesso just-in-time com o Systems Manager.

Preciso configurar o console unificado para usar o acesso a nós just-in-time?

Sim, configurar o console unificado é um pré-requisito para o acesso a nós just-in-time. No entanto, depois de configurar o console unificado e habilitar o acesso a nós just-in-time, há vários métodos para se conectar aos seus nós. Por exemplo, você pode iniciar sessões de acesso a nós just-in-time no console do Amazon EC2 e na AWS CLI. Para obter mais informações sobre a configuração do console unificado, consulte Configurar o console unificado do Systems Manager para uma organização.

Há algum custo associado ao acesso a nós just-in-time?

O Systems Manager oferece um teste gratuito de 30 dias para o acesso a nós just-in-time. Após o teste, o acesso a nós just-in-time gera custos. Para obter mais informações, consulte Preços do AWS Systems Manager.

Qual é a precedência das políticas de aprovação do acesso a nós just-in-time?

As políticas de aprovação são avaliadas na seguinte ordem:

1. Negar acesso
2. Aprovação automática
3. Manual

Como as políticas de aprovação manual são avaliadas?

O acesso a nós just-in-time sempre favorece a política mais específica para um nó. As políticas de aprovação manual são avaliadas na seguinte ordem:

1. Tags específicas para destino
2. Todos os nós para destino

O que acontecerá se não houver uma política de aprovação que se aplique a um nó?

Para se conectar a um nó usando o acesso a nós just-in-time, uma política de aprovação deve ser aplicada ao nó. Se não houver políticas de aprovação que se apliquem a um nó, os usuários não poderão solicitar acesso ao nó.

Várias políticas de aprovação podem ter como destino uma tag?

Uma tag só pode ser de destino uma vez em suas políticas de aprovação.

O que acontecerá se várias políticas de aprovação manual se aplicarem a um nó como resultado da sobreposição de tags?

Quando várias políticas de aprovação manual se aplicam a um nó, isso resulta em um conflito, e os usuários não conseguem solicitar acesso ao nó. Lembre-se disso ao criar suas políticas de aprovação manual, pois algumas instâncias podem ter várias tags, dependendo do seu caso.

Posso usar o acesso a nós just-in-time para solicitar acesso e iniciar sessões em nós em todas as contas e regiões?

O acesso a nós just-in-time oferece suporte à solicitação de acesso e ao início de sessões em nós na mesma conta e região do solicitante.

Posso usar o acesso a nós just-in-time para solicitar acesso e iniciar sessões em nós registrados com uma ativação híbrida?

Sim, o acesso a nós just-in-time oferece suporte à solicitação de acesso e ao início de sessões em nós registrados com uma ativação híbrida. O nó precisa estar registrado na mesma conta e região que o solicitante.

Diagnosticar e remediar

Usando o console unificado do Systems Manager, é possível identificar problemas em toda a sua frota em uma única operação de diagnóstico. Para organizações, você pode então tentar remediar todos os alvos ou apenas selecionar alvos usando uma única operação do Automation. Para uma organização, na condição de administrador de conta delegado, você pode selecionar alvos em todas as contas e regiões. Se você estiver trabalhando em uma única conta, poderá selecionar alvos em uma única região por vez.

O Systems Manager pode diagnosticar e ajudar você a corrigir vários tipos de falhas de implantação, bem como configurações desviadas. O Systems Manager também pode identificar instâncias do Amazon Elastic Compute Cloud (Amazon EC2) em sua conta ou organização que o Systems Manager não pode tratar como um nó gerenciado. O processo de diagnóstico de instâncias do EC2 pode identificar problemas relacionados a configurações incorretas em uma nuvem privada virtual (VPC), em uma configuração do Domain Name Service (DNS) ou em um grupo de segurança do Amazon Elastic Compute Cloud (Amazon EC2).

> **Note**
>
> O Systems Manager oferece suporte a instâncias do EC2 e outros tipos de máquinas em um ambiente híbrido e multinuvem como nós gerenciados. Para ser um nó gerenciado, o AWS Systems Manager Agent (SSM Agent) deve estar instalado na máquina e o Systems Manager deve ter permissão para realizar ações na máquina.
> Para instâncias do EC2, essa permissão pode ser fornecida no nível da conta usando um perfil do AWS Identity and Access Management (IAM) ou no nível da instância usando um

perfil de instância. Para ter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager.
Para máquinas não EC2, essa permissão é fornecida usando um perfil de serviço do IAM. Para ter mais informações, consulte Criar o perfil de serviço do IAM obrigatório para o Systems Manager em ambientes híbridos e multinuvem.

Antes de começar

Para usar o recurso Diagnosticar e corrigir para detectar instâncias do EC2 não gerenciadas, primeiro é necessário integrar sua organização ou conta ao console unificado do Systems Manager. Durante esse processo, você deve escolher a opção de criar os perfis do IAM e as políticas gerenciadas necessárias para essas operações. Para ter mais informações, consulte Configurar o console unificado do Systems Manager para uma organização.

Use os tópicos a seguir para obter ajuda para identificar e corrigir certos tipos comuns de implantações com falha, configurações derivadas e instâncias do EC2 não gerenciadas.

Tópicos

- Diagnosticar e corrigir implantações com falha
- Diagnosticar e revisar configurações com desvios
- Diagnosticar e corrigir instâncias não gerenciadas do Amazon EC2 no Systems Manager
- Tipos de impacto de remediação de ações do runbook
- Verificar o progresso da execução e o histórico de correções no Systems Manager

Diagnosticar e corrigir implantações com falha

O Systems Manager pode diagnosticar e ajudar a corrigir os seguintes tipos de implantações com falha:

- Configuração básica para contas-membro da organização
- Configuração básica para conta de administrador delegado
- Configuração básica para sua conta

Use o procedimento a seguir para tentar corrigir esses tipos de problemas.

Para diagnosticar e corrigir implantações com falha

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Diagnosticar e corrigir.
3. Escolha a guia Problemas de implantação.
4. Na seção Implantações com falha, revise a lista de descobertas de implantações que falharam.
5. Na coluna Etapa de configuração, escolha o nome de uma descoberta para revisar detalhes adicionais sobre o problema. Por exemplo: Configuração básica para contas-membro da organização.
6. Na página de detalhes dessa implantação com falha, é possível visualizar uma lista de contas e quantas regiões em cada uma sofreram falhas de implantação.
7. Selecione um ID de conta para visualizar informações sobre o motivo das falhas nessa conta.
8. Na área Regiões com falha, examine as informações fornecidas pelo Motivo do status. Essas informações podem indicar o motivo da falha na implantação, o que pode fornecer informações sobre as alterações de configuração que precisam ser feitas.
9. Se desejar repetir a implantação sem fazer alterações na configuração, escolha Reimplantar.

Diagnosticar e revisar configurações com desvios

O Systems Manager pode diagnosticar e, em seguida, ajudar a corrigir os seguintes tipos de configurações com desvios:

- Configuração básica para contas-membro da organização
- Configuração básica para conta de administrador delegado
- Configuração básica para sua conta

Use o procedimento a seguir para tentar corrigir esses tipos de configurações com desvios.

Para diagnosticar e revisar configurações com desvios

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Diagnosticar e corrigir.
3. Escolha a guia Problemas de implantação.
4. Na seção Implantações com desvios, revise a lista de descobertas de implantações que falharam.

- ou -

Para executar um novo diagnóstico, escolha Detectar desvio.

5. Na coluna Etapa de configuração, escolha o nome de uma descoberta para revisar detalhes adicionais sobre o problema. Por exemplo: Configuração básica para contas-membro da organização.
6. Na página de detalhes dessa implantação malsucedida, você pode visualizar uma lista de contas e quantas regiões em cada uma sofreram alterações na configuração.
7. Selecione um ID de conta para visualizar informações sobre o motivo dos desvios de configuração nessa conta.
8. Na área Recursos com desvio, a coluna Recursos relata os nomes dos recursos que sofreram desvios. A coluna Tipo de desvio informa se o recurso foi modificado ou excluído.
9. Para reimplantar a configuração pretendida, escolha Reimplantar.

Diagnosticar e corrigir instâncias não gerenciadas do Amazon EC2 no Systems Manager

Para ajudar a gerenciar suas instâncias do Amazon Elastic Compute Cloud (Amazon EC2) com o Systems Manager, você pode usar o console unificado do Systems Manager para fazer o seguinte:

1. Executar um processo de diagnóstico manual ou programado para identificar as instâncias do EC2 em sua conta ou organização que não são gerenciadas pelo Systems Manager no momento.
2. Identificar a rede ou outros problemas que estão impedindo o Systems Manager de assumir o gerenciamento das instâncias.
3. Executar uma execução do Automation para corrigir automaticamente o problema ou acessar informações que podem ajudar a resolver o problema manualmente.

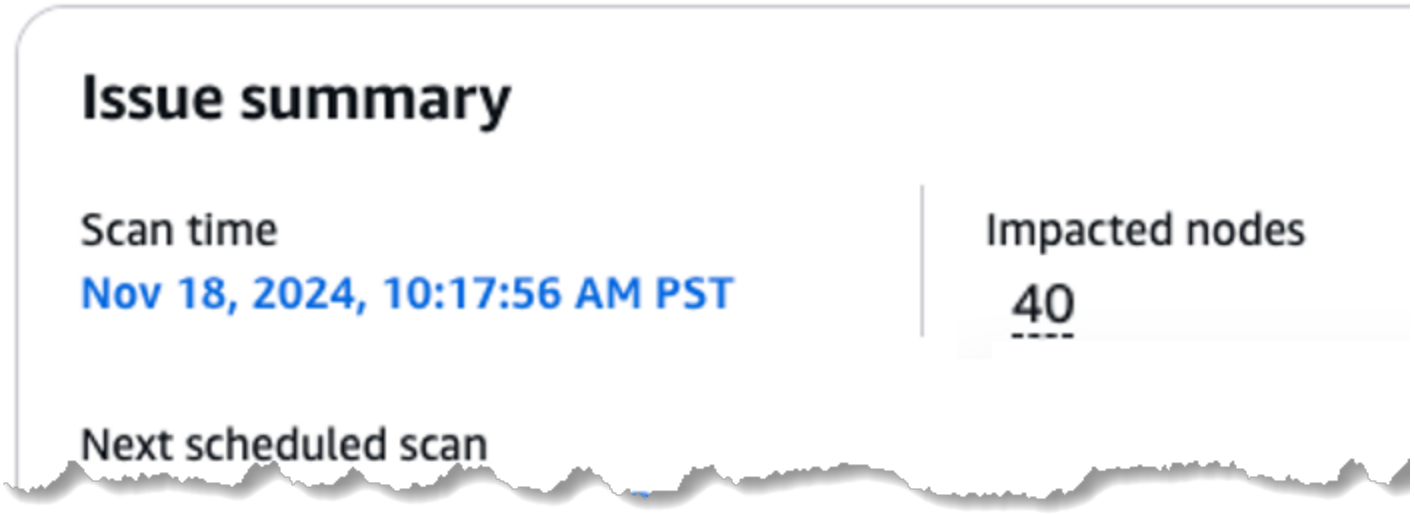
Use as informações nos tópicos a seguir para ajudar no diagnóstico e na correção de problemas que impedem o Systems Manager de gerenciar as instâncias do EC2.

Como o Systems Manager conta os nós afetados para a lista "Problemas de instâncias do EC2 não gerenciadas"

O número de nós relatados como não gerenciados na guia Problemas de instâncias do EC2 não gerenciadas representa o número total de instâncias com qualquer um dos seguintes valores de status no momento da verificação do diagnóstico:

- `Running`
- `Stopped`
- `Stopping`

Esse número é relatado como Nós afetados na área Resumo do problema. Na imagem a seguir, esse número de nós afetados não gerenciados no momento pelo Systems Manager é `40`.



Ao contrário do relatório de instâncias do EC2 não gerenciadas na página Revisar insights do nó, essa contagem de instâncias do EC2 não é dinâmica. Ele representa as descobertas feitas durante a última verificação de diagnóstico relatada, mostrada como o valor de Tempo de verificação. Portanto, recomendamos executar uma verificação de diagnóstico para instâncias do EC2 não gerenciadas em uma programação regular para manter atualizado esse número relatado de nós afetados.

Para obter informações sobre número de instâncias não gerenciadas na página Revisar insights do nó, consulte O que é uma instância não gerenciada? no tópico Como revisar insights de nós.

Tópicos

- Categorias de problemas diagnosticáveis de instâncias do EC2 não gerenciadas

- Executar um diagnóstico e uma correção opcional para instâncias do EC2 não gerenciadas
- Programar uma verificação recorrente para instâncias do EC2 não gerenciadas

Categorias de problemas diagnosticáveis de instâncias do EC2 não gerenciadas

Este tópico lista as principais categorias de problemas de gerenciamento do EC2 e os problemas específicos em cada categoria que o Systems Manager pode ajudar a diagnosticar e corrigir. Observe que, para alguns dos problemas, o Systems Manager pode identificar o problema, mas não fornecer uma correção automática. Nesses casos, o console do Systems Manager direciona você às informações necessárias para ajud a resolver o problema manualmente.

O processo de diagnóstico examina cada grupo de instâncias do EC2 de uma só vez de acordo com a nuvem privada virtual (VPC) à qual elas pertencem.

Tipos de problemas

- Categoria do problema: configuração do grupo de segurança e comunicações HTTPS
- Categoria do problema: configuração do DNS ou do nome de host DNS
- Categoria do problema: configuração do endpoint da VPC

Categoria do problema: configuração do grupo de segurança e comunicações HTTPS

Uma operação de diagnóstico pode descobrir que o SSM Agent não consegue se comunicar com o serviço Systems Manager via HTTPS. Nesses casos, você pode optar por executar um runbook do Automation que tente atualizar os grupos de segurança anexados às instâncias.

> **Note**
>
> Ocasionalmente, o Systems Manager pode não conseguir corrigir automaticamente esses problemas, mas você pode editar manualmente os grupos de segurança afetados.

Tipos de problemas com suporte

- Grupo de segurança da instância: tráfego de saída não é permitido na porta 443
- Grupo de segurança do endpoint da VPC do **`ssm`**: tráfego de entrada não é permitido na porta 443
- Grupo de segurança do endpoint da VPC do **`ssmmessages`**: tráfego de entrada não permitido na porta 443

- Grupo de segurança do endpoint da VPC do **`ec2messages`**: tráfego de entrada não permitido na porta 443

Para obter mais informações, consulte Verificar regras de entrada e saída em grupos de segurança de endpoints no tópico Solução de problemas do SSM Agent.

Categoria do problema: configuração do DNS ou do nome de host DNS

Uma operação de diagnóstico pode descobrir que o Domain Name System (DNS) ou os nomes de host DNS não estão configurados corretamente para a VPC. Nesses casos, você pode optar por executar um runbook do Automation que tente ativar os atributos `enableDnsSupport` e `enableDnsHostnames` da VPC afetada.

Tipos de problemas com suporte

- O suporte ao DNS está desabilitado em uma VPC.
- Um nome de host DNS está desabilitado em uma VPC.

Para obter mais informações, consulte Verificar seus atributos relacionados a DNS da VPC no tópico Solução de problemas do SSM Agent.

Categoria do problema: configuração do endpoint da VPC

Uma operação de diagnóstico pode descobrir que os endpoints da VPC não estão configurados adequadamente para a VPC.

Se os endpoints da VPC exigidos pelo SSM Agent não existirem, o Systems Manager tentará executar um runbook do Automation para criar os endpoints da VPC e associá-los a uma sub-rede em cada zona de disponibilidade regional (AZ) relevante. Se os endpoints da VPC necessários existirem, mas não estiverem associados a uma sub-rede na qual o problema foi encontrado, o runbook associará os endpoints da VPC à sub-rede afetada.

> **Note**
>
> O Systems Manager não oferece suporte a todos os problemas de VPC com configuração incorreta de endpoints da VPC. Nesses casos, o Systems Manager direcionará você para instruções de correção manual em vez de executar um runbook do Automation .

Tipos de problemas com suporte

- Nenhum endpoint `ssm.region.amazonaws.com` para o PrivateLink foi encontrado.
- Nenhum endpoint `ssmmessages.region.amazonaws.com` para o PrivateLink foi encontrado.
- Nenhum endpoint `ec2messages.region.amazonaws.com` para o PrivateLink foi encontrado.

Tipos de problemas diagnosticáveis

O Systems Manager pode diagnosticar os tipos de problemas a seguir, mas, no momento, não há um runbook disponível para corrigi-los. É possível editar sua configuração manualmente para esses problemas.

- A sub-rede de uma instância não está anexada a um endpoint `ssm.region.amazonaws.com`.
- A sub-rede de uma instância não está anexada a um endpoint `ssmmessages.region.amazonaws.com`.
- Sub-rede de uma instância não anexada a um endpoint `ec2messages.region.amazonaws.com`.

Para obter mais informações, consulte Verificar sua configuração de VPC no tópico Solução de problemas do SSM Agent.

Executar um diagnóstico e uma correção opcional para instâncias do EC2 não gerenciadas

Use o procedimento a seguir para diagnosticar os problemas relacionados à rede e à VPC que podem estar impedindo o Systems Manager de gerenciar suas instâncias do EC2.

A operação de diagnóstico pode detectar e agrupar problemas dos seguintes tipos:

- Problemas de configuração de rede: tipos de problemas de rede que podem estar impedindo que as instâncias do EC2 se comuniquem com o serviço Systems Manager na nuvem. Operações de correção podem estar disponíveis para esses problemas. Para obter mais informações sobre os problemas de configuração de rede, consulte Categorias de problemas diagnosticáveis de instâncias do EC2 não gerenciadas.
- Problemas não identificados: uma lista de descobertas de casos em que a operação de diagnóstico não conseguiu determinar por que as instâncias do EC2 não conseguem se comunicar com o serviço Systems Manager na nuvem.

Para executar um diagnóstico e uma correção para instâncias do EC2 não gerenciadas

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Diagnosticar e corrigir.
3. Escolha a guia Problema de instâncias do EC2 não gerenciadas.
4. Na seção Resumo do problema, escolha Executar novo diagnóstico.

   - ou -

   Se esta for a primeira vez que você diagnostica problemas do EC2 não gerenciados, na seção Diagnosticar instâncias do EC2 não gerenciadas, escolha Executar.

   > **Tip**
   >
   > Enquanto o diagnóstico está em execução, escolha Visualizar progresso ou Visualizar execuções para monitorar o estado atual da execução. Para ter mais informações, consulte Verificar o progresso da execução e o histórico de correções no Systems Manager.

5. Após concluir o diagnóstico, faça o seguinte:

   - Para quaisquer problemas relatados na seção Problemas não identificados, escolha o link Saiba mais para obter informações sobre como resolver o problema.
   - Para problemas relatados na seção Problemas de configurações de rede, prossiga para a próxima etapa.

6. Na lista de tipos de descoberta, na coluna Recomendações, para um problema específico, escolha o link, como Duas recomendações.
7. No painel Recomendações que é aberto, escolha entre as mitigações disponíveis:

   - Saiba mais: abra um tópico com informações sobre como resolver um problema manualmente.
   - Visualizar runbook: abra um painel com informações sobre o runbook do Automation que você pode executar para resolver o problema com suas instâncias do EC2, bem como opções para gerar uma prévia das ações que o runbook executaria. Continue na próxima etapa.

8. No painel do runbook, faça o seguinte:

a. Em Descrição do documento, revise o conteúdo, que fornece uma visão geral das ações que o runbook pode realizar para remediar seus problemas de instância do EC2 não gerenciada. Escolha Visualizar etapas para ver uma prévia das ações individuais que o runbook executaria.

b. Em Destinos, faça o seguinte:

   - Se você estiver gerenciando remediações para uma organização, em Contas, especifique se esse runbook teria como alvo todas as contas ou apenas um subconjunto de contas escolhido por você.
   - Em Regiões, especifique se esse runbook teria como alvo todas as Regiões da AWS em sua conta ou organização ou apenas a subconjunto de regiões escolhido por você.

c. Para ver uma Prévia do runbook, revise cuidadosamente as informações. Essas informações explicam qual seria o escopo e o impacto se você optasse por executar o runbook.

   > **Note**
   >
   > Executar de fato o runbook incorreria em cobranças. Revise as informações resultantes da prévia com cuidado antes de decidir se deseja continuar.

   O conteúdo da Prévia do runbook fornece as seguintes informações:

   - Em quantas regiões a operação do runbook ocorreria.
   - (Somente para organizações) Em quantas unidades organizacionais (OUs) a operação seria executada.
   - Os tipos de ações que seriam tomadas e quantas de cada uma.

     Os tipos de ações incluem os seguintes:

     - Mutante: a etapa de runbook faria alterações nos alvos por meio de ações que criam, modificam ou excluem recursos.
     - Não mutante: a etapa do runbook recuperaria dados sobre recursos, mas não faria alterações neles. Essa categoria geralmente inclui `Describe*`, `List*`, `Get*` e ações similares de API somente leitura.
     - Indeterminada: uma etapa indeterminada invoca execuçexecutadas por outro serviço de orquestração como AWS Lambda, AWS Step Functions ou AWS Systems Manager

Run Command. Uma etapa indeterminada também poderia chamar uma API de terceiros. O Systems Manager Automation não conhece o resultado dos processos de orquestração ou das execuções de API de terceiros. Portanto, os resultados das etapas são indeterminados.

d. Nesse ponto, você pode escolher uma das seguintes ações:

- Parar e não executar o runbook.
- Escolher Executar para executar o runbook com as opções que você já selecionou.

Se você decidir executar a operação, escolha Visualizar progresso ou Visualizar execuções para monitorar o estado atual da execução. Para ter mais informações, consulte Verificar o progresso da execução e o histórico de correções no Systems Manager.

Programar uma verificação recorrente para instâncias do EC2 não gerenciadas

Você pode executar uma verificação sob demanda para instâncias do Amazon EC2 em sua conta ou organização que o Systems Manager não consegue gerenciar devido a vários problemas de configuração. Também é possível programar essa verificação para que ela ocorra automaticamente de acordo com uma programação regular.

Para programar uma verificação recorrente para instâncias do EC2 não gerenciadas

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Diagnosticar e corrigir.
3. Escolha a guia Problema de instâncias do EC2 não gerenciadas.
4. Na seção Diagnosticar instâncias do EC2 não gerenciadas, ative Programar diagnóstico recorrente.
5. Em Frequência do diagnóstico, selecione se deseja executar o diagnóstico uma vez por dia ou uma vez por semana.
6. (Opcional) Em Hora de início, insira uma hora, no formato de 24 horas, para o início do diagnóstico. Por exemplo, para às 8h15 da noite, insira **`20:15`**.

   A hora inserida deve estar no fuso horário local atual.

   Se você não especificar uma hora, a verificação de diagnóstico será executada imediatamente. O Systems Manager também programa a verificação para ser executada no futuro no horário

atual. Se você especificar um horário, o Systems Manager aguardará para executar a verificação de diagnóstico no horário especificado.

7. Clique em Executar. O diagnóstico é executado imediatamente, mas também de acordo com a programação que você especificou.

Tipos de impacto de remediação de ações do runbook

O Systems Manager pode executar operações de diagnóstico que descobrem certos tipos de implantações com falha e configurações derivadas, bem como determinados tipos de problemas de configuração que estão impedindo o Systems Manager de gerenciar instâncias do EC2. Os resultados do diagnóstico podem incluir recomendações para runbooks do Automation que você pode executar para tentar solucionar um problema. Para obter mais informações sobre essas operações de diagnóstico, consulte os seguintes tópicos:

- Diagnosticar e corrigir implantações com falha
- Diagnosticar e revisar configurações com desvios
- Diagnosticar e corrigir instâncias não gerenciadas do Amazon EC2 no Systems Manager

Quando o Systems Manager identifica um problema que pode ser corrigido por meio da execução de um runbook do Automation nos recursos afetados, ele fornece uma prévia da execução. A prévia da execução fornece informações sobre os tipos de alterações que a execução do runbook faria em seus alvos. Essas informações incluem quanto de cada um dos três tipos de alterações o diagnóstico identificou.

Esses tipos de alteração são os seguintes:

- `Mutating`: uma etapa do runbook faria alterações nos alvos por meio de ações que criam, modificam ou excluem recursos.
- `Non-Mutating`: uma etapa do runbook recuperaria dados sobre recursos, mas não faria alterações neles. Essa categoria geralmente inclui `Describe*`, `List*`, `Get*` e ações similares de API somente leitura.
- `Undetermined`: uma etapa indeterminada invoca execuções executadas por outro serviço de orquestração como AWS Lambda, AWS Step Functions ou Run Command, uma ferramenta do AWS Systems Manager. Uma etapa indeterminada também pode chamar uma API de terceiros ou executar um script Python ou PowerShell. O Systems Manager Automation não consegue detectar qual seria o resultado dos processos de orquestração ou das execuções de API de

terceiros e, portanto, não pode avaliá-los. Os resultados dessas etapas precisariam ser revisados manualmente para determinar seus impactos.

Consulte a tabela a seguir para obter informações sobre o tipo de impacto das ações do Automation válidas.

Tipos de impacto das ações de correção válidas

A tabela apresenta o tipo de impacto (Mutante, Não mutante e Indeterminada) de várias ações que podem ser incluídas em um runbook de remediação.

Ação[1]	Tipo de impacto
aws:approve	Não mutante
aws:assertAwsResourceProperty	Não mutante
aws:branch	Não mutante
aws:changeInstanceState	Mutante
aws:copyImage	Mutante
aws:createImage	Mutante
aws:createStack	Mutante
aws:createTags	Mutante
aws:deleteImage	Mutante
aws:deleteStack	Mutante
aws:executeAutomation	Indeterminada
aws:executeAwsApi	Indeterminada
aws:executeScript	Indeterminada
aws:executeStateMachine	Indeterminada

Ação[1]	Tipo de impacto
aws:invokeLambdaFunction	Indeterminada
aws:invokeWebhook	Indeterminada
aws:loop	Varia. Depende das ações no loop.
aws:pause	Não mutante
aws:runCommand	Indeterminada
aws:runInstances	Mutante
aws:sleep	Não mutante
aws:updateVariable	Mutante
aws:waitForAwsResourceProperty	Não mutante

[1] Para obter informações sobre ações do Automation, consulte Referência de ações do Systems Manager Automation.

Verificar o progresso da execução e o histórico de correções no Systems Manager

É possível visualizar uma lista de todas operações de correção em andamento e concluídas feitas usando o recurso Diagnosticar e corrigir no Systems Manager.

Os dados na lista do histórico de execução relatam os seguintes tipos de informações:

- O tipo de execução, `Diagnosis` ou `Remediation`.
- O status da execução, como `Success` ou `Failed`.
- Os horários em que a execução começou e terminou.

Para verificar o progresso da execução e o histórico de correções

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Diagnosticar e corrigir.

3. Escolha Visualizar execuções.

> **Tip**
>
> Quando uma execução está em andamento, também é possível escolher Visualizar progresso para abrir a página Histórico de execuções.

4. (Opcional) Na caixa de pesquisa (), insira uma frase para ajudar a restringir a lista de execução, como **EC2** ou **VPC**.
5. (Opcional) Para visualizar detalhes adicionais sobre uma execução, na coluna Nome da execução escolha um nome de operação, como AWS-DiagnoseUnmanagedEC2NetworkIssues.

   No painel de detalhes, você pode revisar as informações sobre todas as etapas tentadas durante a operação e sobre todas as entradas e saídas da execução.

Ajustar as configurações do Systems Manager

As opções nas páginas Configurações habilitam e configuram recursos no console unificado do Systems Manager. As opções exibidas dependem da conta na qual você está conectado e se você já configurou ou não o Systems Manager.

> **Note**
>
> As opções na página Configurações não afetam as ferramentas do Systems Manager (anteriormente chamadas de recursos).

Configurações da conta

Se o Systems Manager estiver habilitado e você estiver conectado a uma conta que não é membro do Organizations ou se o administrador delegado não tiver adicionado sua conta do Organizations ao Systems Manager, a página Configuração da conta mostrará a opção Desabilitar o Systems Manager. Desabilitar o Systems Manager significa que o Systems Manager não exibirá o console unificado. Todas as ferramentas do Systems Manager ainda funcionam.

Configurações organizacionais

Na guia Configuração organizacional, a seção Região de origem exibe a região Região da AWS escolhida como a região de origem durante a configuração. Em ambientes com várias contas e várias regiões que usam o AWS Organizations, o Systems Manager agrega automaticamente os dados dos nós de todas as contas e regiões na região de origem. Agregar dados dessa forma permite que você visualize os dados dos nós em todas as contas e regiões em um único local.

> **Note**
>
> Caso deseje alterar a região de origem, será necessário desabilitar o Systems Manager e habilitá-lo novamente. Para desabilitar o Systems Manager, selecione Desabilitar.

A seção Configuração organizacional exibe as unidades organizacionais da AWS e as Regiões da AWS escolhidas durante a configuração. Para alterar quais unidades organizacionais e regiões exibem dados de nós no Systems Manager, escolha Editar. Para obter mais informações sobre a configuração do Systems Manager for Organizations, consulte Configurar o AWS Systems Manager.

Configurações de recursos

A seção Configurações de recursos permite a você habilitar e configurar os principais recursos do Systems Manager que aprimoram o gerenciamento de nós em sua organização. Esses recursos trabalham juntos para fornecer gerenciamento automatizado, monitoramento de conformidade e manutenção dos seus nós gerenciados.

Você pode configurar esses recursos durante a configuração inicial do Systems Manager ou modificá-los posteriormente por meio da página Configurações. Cada recurso pode ser ativado ou desativado de forma independente com base nos requisitos da sua organização.

Configuração de gerenciamento de host padrão

A Configuração de gerenciamento de hosts padrão (DHMC) configura automaticamente instâncias do Amazon Elastic Compute Cloud (Amazon EC2) em sua organização para ser gerenciadas pelo Systems Manager. Quando habilitada, a DHMC garante que as instâncias do EC2 novas e existentes tenham as permissões e configurações do AWS Identity and Access Management (IAM) necessárias para se comunicar com os serviços do Systems Manager.

O DHMC fornece os seguintes benefícios:

- Atribuição automática de perfis do IAM: garante que as instâncias do EC2 tenham os perfis e políticas do IAM necessários para funcionar como nós gerenciados
- Correção de desvios: corrige automaticamente os desvios de configuração quando as instâncias perdem o status de nó gerenciado
- Integração simplificada: reduz as etapas de configuração manual para novas instâncias
- Configuração consistente: mantém configurações uniformes em toda a sua frota do EC2

Configurar a frequência de correção de desvios

A correção de desvios detecta e corrige automaticamente quando as instâncias do EC2 perdem sua configuração de nó gerenciado. É possível configurar a frequência com que o Systems Manager verifica e corrige os desvios de configuração.

Para ajustar a Configuração de gerenciamento de hosts padrão

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, selecione Configurações.
3. Na seção Configurações de recursos, localize Configuração de gerenciamento de hosts padrão.
4. Para habilitar a DHMC, ative a chave seletora.
5. Para Frequência de correção de desvios, escolha com que frequência você deseja que o Systems Manager verifique e corrija o desvio de configuração:
   - Diariamente: verifica e corrige desvios uma vez por dia
   - Semanalmente: verifica e corrige desvios uma vez por semana
   - Mensalmente: verifica e corrige desvios uma vez por mês
6. Escolha Salvar.

> **Note**
>
> Quando você habilita a DHMC, o Systems Manager cria os perfis e políticas do IAM necessários em sua conta. Esses perfis permitem que as instâncias do EC2 se comuniquem com os serviços do Systems Manager. Para obter mais informações sobre os perfis do IAM criados pela DHMC, consulte Gerenciar instâncias do EC2 com o Systems Manager.

Coleta de metadados de inventário

A coleta de metadados de inventário reúne automaticamente informações detalhadas sobre seus nós gerenciados, incluindo aplicações instaladas, configurações de rede, atualizações do sistema e outros metadados do sistema. Essas informações ajudam você a manter a conformidade, realizar análises de segurança e entender a composição da sua infraestrutura.

A coleta de inventário oferece os seguintes benefícios:

- Monitoramento da conformidade: acompanhe o software instalado e as configurações para relatórios de conformidade
- Análise de segurança: identifique software desatualizado e possíveis vulnerabilidades de segurança
- Gerenciamento de ativos: mantenha um inventário atualizado da sua infraestrutura
- Recursos de consulta: use os dados coletados com o Amazon Q Developer para consultas em linguagem natural

Tipos de dados de inventário coletados

Quando a coleta de metadados de inventário está habilitada, o Systems Manager coleta os seguintes tipos de informações de seus nós gerenciados:

- Aplicações: pacotes de software e aplicações instalados
- Configurações de rede: interfaces de rede, endereços IP e configurações de rede
- Atualizações do sistema: patches instalados e atualizações disponíveis
- Propriedades do sistema: especificações de hardware, detalhes do sistema operacional e configurações do sistema
- Serviços: serviços em execução e suas configurações

Configurar a frequência da coleta de inventário

É possível configurar a frequência na qual o Systems Manager coleta metadados de inventário dos nós gerenciados. A coleta mais frequente fornece informações mais atualizadas, mas pode aumentar o uso do serviço da AWS.

Para configurar a coleta de metadados do inventário

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, selecione Configurações.
3. Na seção Configurações de recursos, localize Coleta de metadados de inventário.
4. Para habilitar a coleta de inventário, ative a chave seletora.
5. Em Frequência de coleta, escolha com que frequência você deseja que o Systems Manager colete dados de inventário:
   - Diariamente: coleta dados de inventário uma vez por dia
   - Semanalmente: coleta dados de inventário uma vez por semana
   - Mensalmente: coleta dados de inventário uma vez por mês
6. Escolha Salvar.

> **Important**
>
> A coleta de inventário exige que os nós gerenciados tenham as permissões necessárias para coletar informações do sistema. Certifique-se de que seus nós gerenciados tenham os perfis e políticas apropriados do IAM. Para obter mais informações sobre as permissões necessárias, consulte AWS Systems Manager Inventory.

Atualizações do SSM Agent

As atualizações automáticas do SSM Agent garantem que seus nós gerenciados estejam executando a versão mais recente do SSM Agent. Manter o agente atualizado fornece acesso aos recursos, melhorias de segurança e correções de erros mais recentes.

As atualizações automáticas do SSM Agent oferecem os seguintes benefícios:

- Recursos mais recentes: acesso aos novos recursos e melhorias do Systems Manager
- Atualizações de segurança: instalação automática de patches e correções de segurança
- Confiabilidade aprimorada: correções de erros e melhorias na estabilidade
- Manutenção reduzida: elimina a necessidade de atualizações manuais do agente

Configurar atualizações automáticas do agente

É possível configurar a frequência com que o Systems Manager verifica e instala atualizações do SSM Agent em seus nós gerenciados. Fazer atualizações regularmente ajuda a garantir níveis ideais de performance e segurança.

Para configurar atualizações do SSM Agent

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, selecione Configurações.
3. Na seção Configurações de recursos, localize Atualizações do SSM Agent.
4. Para habilitar as atualizações automáticas, ative a chave seletora.
5. Para Frequência de atualização, escolha com que frequência você deseja que o Systems Manager verifique e instale as atualizações do agente:
   - Diariamente: verifica se há atualizações uma vez por dia
   - Semanalmente: verifica se há atualizações uma vez por semana
   - Mensalmente: verifica se há atualizações uma vez por mês
6. Escolha Salvar.

Configurações de Diagnosticar e corrigir

As configurações de Diagnosticar e corrigir determinam se o Systems Manager verifica automaticamente seus nós para garantir que eles possam se comunicar com o Systems Manager. Se habilitado, o recurso é executado automaticamente de acordo com um cronograma definido por você. O recurso identifica quais nós não podem se conectar ao Systems Manager e por quê. Esse recurso também fornece runbooks recomendados para corrigir problemas de rede e outros problemas que impedem que os nós sejam configurados como nós gerenciados.

Programar uma verificação de diagnóstico recorrente

O Systems Manager pode diagnosticar e ajudar você a corrigir vários tipos de falhas de implantação, bem como configurações desviadas. O Systems Manager também pode identificar instâncias do Amazon Elastic Compute Cloud (Amazon EC2) em sua conta ou organização que o Systems Manager não pode tratar como um nó gerenciado. O processo de diagnóstico de instâncias do EC2 pode identificar problemas relacionados a configurações incorretas em uma nuvem privada virtual

(VPC), em uma configuração do Domain Name Service (DNS) ou em um grupo de segurança do Amazon Elastic Compute Cloud (Amazon EC2).

Para simplificar a tarefa de identificar nós que não conseguem se conectar ao Systems Manager, o recurso Programar diagnóstico recorrente permite automatizar uma verificação de diagnóstico recorrente. As verificações ajudam a identificar quais nós não podem se conectar ao Systems Manager e por quê. Use o procedimento a seguir para habilitar e configurar uma verificação de diagnóstico recorrente de seus nós.

Para programar uma verificação de diagnóstico recorrente

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, selecione Configurações e em seguida, a guia Diagnosticar e corrigir.
3. Ative a opção Programar diagnóstico recorrente.
4. Em Período de verificação, escolha a frequência na qual você deseja que a verificação seja executada.
5. (Opcional) Em Hora de início, insira uma hora, no formato de 24 horas, para o início do diagnóstico. Por exemplo, para às 8h15 da noite, insira **`20:15`**.

   A hora inserida deve estar no fuso horário local atual.

   Se você não especificar uma hora, a verificação de diagnóstico será executada imediatamente. O Systems Manager também programa a verificação para ser executada no futuro no horário atual. Se você especificar um horário, o Systems Manager aguardará para executar a verificação de diagnóstico no horário especificado.
6. Escolha Salvar.
7. Depois que a verificação for concluída, visualize os detalhes escolhendo Diagnosticar e corrigir no painel de navegação à esquerda.

Para obter mais informações sobre o recurso Diagnosticar e corrigir, consulte Diagnosticar e remediar.

Atualizar a criptografia do bucket do S3

Quando você integra o Systems Manager, a Configuração Rápida cria um bucket do Amazon Simple Storage Service (Amazon S3) na conta do administrador delegado para configurações do AWS Organizations. Para configurações de conta única, o bucket é armazenado na conta que está sendo

configurada. Esse bucket é usado para armazenar os metadados gerados durante as verificações de diagnóstico.

Para obter mais informações sobre a configuração do console unificado do Systems Manager, consulte Configurar o AWS Systems Manager.

Por padrão, seus dados no bucket são criptografados usando uma chave AWS Key Management Service (AWS KMS) que pertence a e é gerenciada pela AWS para você.

Você pode optar por usar uma chave do AWS KMS diferente para criptografar o bucket. Outra opção é usar a criptografia no lado do servidor com o AWS KMS keys (SSE-KMS) utilizando uma chave gerenciada pelo cliente (CMK). Para mais informações, consulte Como trabalhar com buckets do Amazon S3 e com políticas de bucket para o Systems Manager.

Para usar uma chave do AWS KMS diferente para criptografar o bucket do S3

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, selecione Configurações e em seguida, a guia Diagnosticar e corrigir.
3. Na área Atualizar criptografia do bucket do S3, escolha Editar.
4. Marque a caixa de seleção Personalizar configurações de criptografia (avançadas).
5. Em Escolher uma chave do AWS KMS, escolha ou insira o nome do recurso da Amazon (ARN) da chave.

   > **Tip**
   >
   > Para criar uma nova chave, escolha Criar uma chave do AWS KMS.

6. Escolha Salvar.

Como trabalhar com buckets do Amazon S3 e com políticas de bucket para o Systems Manager

Durante o processo de integração do AWS Systems Manager, o Quick Setup cria um bucket do Amazon Simple Storage Service (Amazon S3) na conta de administrador delegado para configurações da organização. Para configurações de conta única, o bucket é armazenado na conta que está sendo configurada.

É possível usar o Systems Manager para executar operações de diagnóstico em sua frota para identificar casos de falhas em implantações e configurações com desvios. O Systems Manager também pode detectar casos em que problemas de configuração estão impedindo o Systems Manager de gerenciar instâncias do EC2 em sua conta ou organização. Os resultados dessas operações de diagnóstico são armazenados nesse bucket do Amazon S3, o qual é protegido por um método de criptografia e por uma política de bucket do S3. Para obter informações sobre as operações de diagnóstico que geram dados para esse bucket, consulte Diagnosticar e remediar.

Alterar o método de criptografia do bucket

Por padrão, o bucket do S3 usa criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3).

Em vez disso, você pode usar criptografia do lado do servidor com AWS KMS keys (SSE-KMS) usando uma chave gerenciada pelo cliente (CMK) como alternativa às chaves gerenciadas pelo Amazon S3, conforme explicado em Mudar para uma chave gerenciada pelo cliente do AWS KMS para fornecer criptografia para recursos do S3.

Conteúdo da política de buckets

A política de buckets impede que as contas-membro de uma organização descubram umas às outras. As permissões de leitura e gravação no bucket são permitidas somente para os perfis de diagnóstico e remediação criados para o Systems Manager. O conteúdo dessas políticas geradas pelo sistema é apresentado em Políticas de bucket do S3 para o console unificado do Systems Manager.

> **Warning**
>
> A modificação da política de bucket padrão pode permitir que as contas-membro de uma organização descubram umas as outras ou leiam os resultados de diagnóstico de instâncias em outra conta. Recomendamos ter muito cuidado ao optar por modificar essa política.

Tópicos

- Mudar para uma chave gerenciada pelo cliente do AWS KMS para fornecer criptografia para recursos do S3
- Políticas de bucket do S3 para o console unificado do Systems Manager

Mudar para uma chave gerenciada pelo cliente do AWS KMS para fornecer criptografia para recursos do S3

Durante o processo de integração do console unificado do Systems Manager, a Quick Setup cria um bucket do Amazon Simple Storage Service (Amazon S3) na conta do administrador delegado. Esse bucket é usado para armazenar os dados de saída de diagnóstico gerados durante as execuções do runbook de correção. Por padrão, o bucket usa criptografia do lado do servidor com chaves de criptografia gerenciadas pelo Amazon S3 (SSE-S3).

Você pode revisar o conteúdo dessas políticas em Políticas de bucket do S3 para o console unificado do Systems Manager.

No entanto, você pode usar a criptografia no lado do servidor com o AWS KMS keys (SSE-KMS) usando uma chave gerenciada pelo cliente (CMK) como alternativa a uma AWS KMS key.

Conclua as tarefas a seguir para configurar o Systems Manager para usar sua CMK.

Tarefa 1: adicionar uma tag a uma CMK existente

O AWS Systems Manager usará sua CMK somente se ela estiver marcada com o seguinte par de chave-valor:

- Chave: `SystemsManagerManaged`
- Valor: `true`

Use o procedimento a seguir para fornecer acesso para criptografar o bucket do S3 com sua CMK.

Para adicionar uma tag à sua CMK existente

1. Abra o console do AWS KMS em https://console.aws.amazon.com/kms.
2. Na navegação esquerda, escolha Chaves gerenciadas pelo cliente.
3. Selecione o AWS KMS key para usar com o AWS Systems Manager.
4. Selecione a guia Etiquetas e escolha Editar.
5. Escolha Adicionar Tag.
6. Faça o seguinte:
   a. Em Tag Key (Chave de tags), digite **`SystemsManagerManaged`**.
   b. Em Valor da tag, insira **`true`**.

7. Escolha Salvar.

Tarefa 2: modificar uma política de chave de CMK existente

Use o procedimento a seguir para atualizar a política de chave do KMS da sua CMK para permitir que os perfis do AWS Systems Manager criptografem o bucket do S3 em seu nome.

Para modificar uma política de chave de CMK existente

1. Abra o console do AWS KMS em https://console.aws.amazon.com/kms.
2. Na navegação esquerda, escolha Chaves gerenciadas pelo cliente.
3. Selecione o AWS KMS key para usar com o AWS Systems Manager.
4. Na guia Política de chave, escolha Editar.
5. Adicione a instrução JSON a seguir ao campo `Statement` e substitua os *`valores do espaço reservado`* por suas próprias informações.

   Certifique-se de adicionar todas as IDs de Conta da AWS integradas em sua organização ao AWS Systems Manager no campo `Principal`.

   Para localizar o nome correto do bucket no console do Amazon S3, na conta do administrador delegado, localize o bucket no formato `do-not-delete-ssm-`*`operational-account-id`*`-`*`home-region`*`-`*`disambiguator`*.

```
{
     "Sid": "EncryptionForSystemsManagerS3Bucket",
     "Effect": "Allow",
     "Principal": {
         "AWS": [
             "account-id-1",
             "account-id-2",
             ...
         ]
     },
     "Action": ["kms:Decrypt", "kms:GenerateDataKey"],
     "Resource": "*",
     "Condition": {
         "StringEquals": {
             "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::amzn-s3-demo-bucket"
         },
         "StringLike": {
```

```
                "kms:ViaService": "s3.*.amazonaws.com"
            },
            "ArnLike": {
                "aws:PrincipalArn": "arn:aws:iam::*:role/AWS-SSM-*"
            }
        }
    }
```

> **Tip**
>
> Como alternativa, é possível atualizar a política de chave de CMK usando a chave de condição aws:PrincipalOrgID para conceder ao AWS Systems Manager acesso à sua CMK.

Tarefa 3: especificar a CMK nas configurações do Systems Manager

Após concluir as duas tarefas anteriores, use o procedimento a seguir para alterar a criptografia do bucket do S3. Essa alteração garante que o processo de configuração da Quick Setup associado possa adicionar permissões para que o Systems Manager aceite sua CMK.

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, selecione Configurações.
3. Na guia Diagnosticar e corrigir, na seção Atualizar criptografia do bucket S3, escolha Editar.
4. Marque a caixa de seleção Personalizar configurações de criptografia (avançadas).
5. Na caixa de pesquisa (), escolha o ID de uma chave existente ou cole o ARN de uma chave existente.
6. Escolha Salvar.

Políticas de bucket do S3 para o console unificado do Systems Manager

Este tópico inclui as políticas de bucket do Amazon S3 criadas pelo Systems Manager quando você integra uma organização ou conta única ao console unificado do Systems Manager.

> **Warning**
>
> A modificação da política de bucket padrão pode permitir que as contas-membro de uma organização descubram umas as outras ou leiam os resultados de diagnóstico de instâncias em outra conta. Recomendamos ter muito cuidado ao optar por modificar essa política.

Política de bucket do Amazon S3 para uma organização

O bucket de diagnóstico é criado com a seguinte política de bucket padrão ao integrar uma organização ao Systems Manager.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyHTTPRequests",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ],
            "Condition": {
                "Bool": {
                    "aws:SecureTransport": "false"
                }
            }
        },
        {
            "Sid": "DenyNonSigV4Requests",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ],
```

```
            "Condition": {
                "StringNotEquals": {
                    "s3:SignatureVersion": "AWS4-HMAC-SHA256"
                }
            }
        },
        {
            "Sid": "AllowAccessLog",
            "Effect": "Allow",
            "Principal": {
                "Service": "logging.s3.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/access-logs/*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "operational-account-id"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::amzn-s3-demo-bucket"
                }
            }
        },
        {
            "Sid": "AllowCrossAccountRead",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/actions/*/
${aws:PrincipalAccount}/*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": "organization-id"
                }
            }
        },
        {
            "Sid": "AllowCrossAccountWrite",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:PutObject",
                "s3:DeleteObject"
            ],
```

```
            "Resource": "arn:aws:s3:::bucket-name/actions/*/
${aws:PrincipalAccount}/*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": "organization-id"
                },
                "ArnLike": {
                    "aws:PrincipalArn": [
                        "arn:aws:iam::*:role/AWS-SSM-
DiagnosisExecutionRole-operational-account-id-home-region",
                        "arn:aws:iam::*:role/AWS-SSM-
DiagnosisAdminRole-operational-account-id-home-region",
                        "arn:aws:iam::*:role/AWS-SSM-
RemediationExecutionRole-operational-account-id-home-region",
                        "arn:aws:iam::*:role/AWS-SSM-
RemediationAdminRole-operational-account-id-home-region"
                    ]
                }
            }
        },
        {
            "Sid": "AllowCrossAccountListUnderAccountOwnPrefix",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": "organization-id"
                },
                "StringLike": {
                    "s3:prefix": "*/${aws:PrincipalAccount}/*"
                }
            }
        },
        {
            "Sid": "AllowCrossAccountGetConfigWithinOrganization",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetEncryptionConfiguration",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": "organization-id"
```

```
                }
            }
        }
    ]
}
```

Política de bucket do Amazon S3 para uma única conta

O bucket de diagnóstico é criado com a seguinte política de bucket padrão ao integrar uma conta única ao Systems Manager.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyHTTPRequests",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ],
      "Condition": {
        "Bool": {
          "aws:SecureTransport": "false"
        }
      }
    },
    {
      "Sid": "DenyNonSigV4Requests",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ],
      "Condition": {
        "StringNotEquals": {
```

```
          "s3:SignatureVersion": "AWS4-HMAC-SHA256"
        }
      }
    }
  ]
}
```

Como usar ferramentas do AWS Systems Manager

O Systems Manager agrupa as ferramentas em quatro categorias. A documentação a seguir descreve as várias ferramentas do AWS Systems Manager e como configurá-las e usá-las. Selecione as guias em cada categoria para saber mais sobre cada ferramenta.

Ferramentas de nós

Um nó gerenciado é qualquer máquina configurada para uso com o Systems Manager em ambientes híbridos e multinuvem.

Compliance

Use Conformidade para verificar a conformidade dos patches e as inconsistências de configuração em sua frota de nós gerenciados. Você pode coletar e agregar dados de várias Contas da AWS e Regiões da AWS e depois fazer buscas detalhadas em recursos específicos que não forem compatíveis. Por padrão, a Conformidade exibe dados de conformidade sobre a aplicação de patch do Patch Manager e as associações do State Manager. Você também pode personalizar o serviço e criar seus próprios tipos de conformidade com base nos seus requisitos de IT ou negócios.

Distribuidor

Use o Distributor para criar e implantar pacotes em nós gerenciados. Com o Distributor, é possível criar seu próprio pacote de software ou encontrar pacotes de software do agente fornecidos pela AWS, como o AmazonCloudWatchAgent, para instalar em nós gerenciados pelo Systems Manager. Depois de instalar um pacote pela primeira vez, você poderá usar o Distributor para desinstalar e reinstalar uma nova versão do pacote ou executar uma atualização local que adiciona arquivos novos ou alterados. O Distributor publica recursos, como pacotes de software, em nós gerenciados do Systems Manager.

Fleet Manager

O Fleet Manager é uma experiência de interface de usuário unificada (UI) que ajuda você a gerenciar remotamente os nós. Com o Fleet Manager, você pode visualizar a integridade e o status da performance de toda a sua frota em um único console. Você também pode coletar dados de instâncias e dispositivos individuais para executar tarefas comuns de solução de problemas e gerenciamento no console. Isso inclui a exibição de conteúdo de diretórios e

arquivos, gerenciamento de registros do Windows, gerenciamento de usuários do sistema operacional e muito mais.

Hybrid Activations

Para configurar máquinas que não são do EC2 em seu ambiente híbrido e multinuvem como nós gerenciados, crie uma ativação híbrida. Depois de concluir a ativação, você receberá um código de ativação e um ID. Essa combinação de código/ID funciona como um ID de acesso e uma chave secreta do Amazon Elastic Compute Cloud (Amazon EC2) para fornecer acesso seguro ao serviço do Systems Manager nas instâncias gerenciadas.

Você também pode criar uma ativação para dispositivos de borda se quiser gerenciá-los usando o Systems Manager.

Inventory

O Inventário automatiza o processo de coleta de inventário de software de seus nós gerenciados. Você pode usar o inventário para reunir metadados sobre aplicações, arquivos, componentes, patches e muito mais.

Patch Manager

Use o Patch Manager para automatizar o processo de aplicação de patches aos nós gerenciados com atualizações de segurança e outros tipos de atualizações. Você pode usar o Patch Manager para aplicar patches em sistemas operacionais e aplicações. (No Windows Server, o suporte a aplicações é limitado a atualizações de aplicações da Microsoft.)

Essa ferramenta permite verificar os nós gerenciados em busca de patches ausentes e aplicá-los individualmente ou em grandes grupos de nós gerenciados usando tags. O Patch Manager usa a lista de referência de patches, o que pode incluir regras para aprovação automática de patches dentro de dias após o lançamento e uma lista de patches aprovados e rejeitados. Você pode instalar patches de segurança regularmente programando a aplicação de patches para ser executada como uma tarefa da janela de manutenção do Systems Manager, ou pode corrigir os nós gerenciados sob demanda a qualquer momento.

Para sistemas operacionais Linux, você pode definir os repositórios que devem ser usados para operações de patch como parte de sua linha de base de patch. Isso permite que você verifique se as atualizações são instaladas apenas de repositórios confiáveis, independentemente de quais repositórios são configurados em seu nó gerenciado. Para o Linux, você também tem a capacidade de atualizar qualquer pacote em seu nó gerenciado, não apenas as que são classificadas como as atualizações de segurança do sistema operacional. Também é possível gerar relatórios de patches que são enviados para um bucket do S3 de sua preferência. Para um

único nó gerenciado, os relatórios incluem detalhes de todos os patches para a máquina. Para obter um relatório sobre todas os nós gerenciados, apenas um resumo de quantos patches estão ausentes é fornecido.

Executar comando

Use o Run Command para gerenciar remotamente e de forma segura a configuração em grande escala de seus nós gerenciados. Use o Run Command para realizar alterações sob demanda, como atualizar aplicações ou executar scripts de shell do Linux e comandos do Windows PowerShell em um conjunto de destino de dezenas ou centenas de nós gerenciados.

Session Manager

Use o Session Manager para gerenciar seus dispositivos de borda do Amazon Elastic Compute Cloud (Amazon EC2) por meio de um shell interativo baseado em navegador acionado por um clique ou por meio da AWS CLI. O Session Manager fornece gerenciamento seguro e auditável de instâncias sem a necessidade de abrir portas de entrada, manter bastion hosts ou gerenciar chaves de SSH. O Session Manager também permite cumprir as políticas corporativas que exigem acesso controlado às instâncias, práticas rígidas de segurança e logs totalmente auditáveis com detalhes do acesso à instância, fornecendo ao mesmo tempo aos usuários finais o acesso interplataformas com apenas um clique a suas instâncias do EC2. Para usar o Session Manager ative o nível de instâncias avançadas. Para obter mais informações, consulte Ativar o nível de instâncias avançadas.

State Manager

Use o State Manager para automatizar o processo de manutenção de seus nós gerenciados em um estado definido. Você pode usar o State Manager para garantir que os nós gerenciados passem por bootstrap com software específico no startup inicialização, ingressem em um domínio do Windows (apenas para nós gerenciados do Windows Server) ou recebam patches com atualizações específicas de software.

Ferramentas de gerenciamento de alterações

Automation

Use o Automation para automatizar tarefas comuns de manutenção e implantação. Você pode usar a automação para criar e atualizar as Amazon Machine Images (AMIs), aplicar atualizações de drivers e agentes, redefinir senhas na instância do Windows Server, redefinir chaves SSH em instâncias do Linux e aplicar patches do OS ou atualizações de aplicações.

Alterar calendário

O Change Calendar ajuda você a configurar intervalos de data e hora quando as ações especificadas (por exemplo, em runbooks do Automation do Systems Manager) puderem ou não ser executadas em sua Conta da AWS. No Change Calendar, esses intervalos são chamados de eventos. Ao criar uma entrada do Change Calendar, você está criando um documento do Systems Manager do tipo `ChangeCalendar`. No Change Calendar, o documento armazena dados do iCalendar 2.0 em formato de texto simples. Os eventos adicionados à entrada do Change Calendar tornam-se parte do documento. Você pode adicionar eventos manualmente na interface do Change Calendar importar eventos de um calendário de terceiros com suporte usando um arquivo do `.ics`.

Gerenciador de alterações

O Change Manager é um framework empresarial de gerenciamento de alterações para solicitar, aprovar, implementar e emitir relatórios sobre alterações operacionais em sua configuração e infraestrutura de aplicações. Em uma única Conta de administrador delegado, se você usar o AWS Organizations, poderá gerenciar alterações em várias Contas da AWS e Regiões da AWS. Como alternativa, usando um conta local, você pode gerenciar alterações para uma única Conta da AWS. Use o Change Manager para gerenciar alterações em recursos da AWS e recursos on-premises.

Documents

Um Documento do Systems Manager (documento SSM) define a ação que o Systems Manager realiza. Os tipos de documentos do SSM incluem os de Comando, que são usados pelo State Manager e Run Command, e runbooks do Automation, que são usados pelo Systems Manager Automation. O Systems Manager inclui dezenas de documentos pré-configurados, que você pode usar especificando parâmetros no runtime. Os documentos podem ser expresso em JSON ou YAML e incluem etapas e parâmetros especificados por você.

Janelas de manutenção

Use o Maintenance Windows para configurar programações recorrentes de instâncias gerenciadas e executar tarefas administrativas, como atualizações e instalação de patches, sem interromper operações comerciais essenciais.

Instalação rápida

Use o Quick Setup para configurar Serviços da AWS e recursos frequentemente utilizados com práticas recomendadas. Você pode usar o Quick Setup em uma Conta da AWS individual ou por

meio de várias Contas da AWS e Regiões da AWS pela integração com o AWS Organizations. O Quick Setupsimplifica a configuração de serviços, incluindo o Systems Manager, automatizando tarefas comuns ou recomendadas. Essas tarefas incluem, por exemplo, a criação de funções de perfil da instância AWS Identity and Access Management (IAM) e a configuração de práticas recomendadas operacionais, como verificações periódicas de patches e coleta de inventário.

Ferramentas de aplicações

AppConfig

O AppConfig ajuda a criar, gerenciar e implantar configurações de aplicações e sinalizadores de recursos. O AppConfig oferece suporte a implantações controladas em aplicações de qualquer tamanho. Você pode usar o AppConfig com aplicações hospedadas em instâncias do Amazon EC2, contêineres do AWS Lambda, aplicações móveis ou dispositivos de borda. Para evitar erros ao implantar configurações de aplicativos, o AppConfig inclui validadores. Um validador fornece uma verificação sintática ou semântica para verificar se a configuração que você quer implantar funciona conforme previsto. Durante uma implantação de configuração, o AppConfig monitora a aplicação para verificar se ela foi bem-sucedida. Se o sistema encontrar um erro ou se a implantação invocar um alarme, o AppConfig reverterá a alteração para minimizar o impacto para os usuários da aplicação.

Application Manager

O Application Manager ajuda os engenheiros de DevOps a investigar e corrigir problemas com seus recursos da AWS no contexto de suas aplicações e clusters. No Application Manager, uma aplicação é um grupo lógico de recursos da AWS que você deseja operar como uma unidade. Esse grupo lógico pode representar diferentes versões de uma aplicação, limites de propriedade para operadores ou ambientes de desenvolvedor, entre outros. O suporte do Application Manager a clusters de contêiner inclui clusters do Amazon Elastic Kubernetes Service (Amazon EKS) e do Amazon Elastic Container Service (Amazon ECS). O Application Manager agrega informações de operações de vários Serviços da AWS e ferramentas do Systems Manager em um único AWS Management Console.

Parameter Store

O Parameter Store oferece armazenamento hierárquico seguro para gerenciamento de dados de configuração e gerenciamento de segredos. É possível armazenar dados como senhas, strings de banco de dados, IDs de instância do Amazon Elastic Compute Cloud (Amazon EC2), IDs do Amazon Machine Image (AMI) e códigos de licença como valores de parâmetros. É

possível armazenar valores como texto sem formatação ou dados criptografados. Você pode fazer referência a valores usando o nome exclusivo que especificou ao criar o parâmetro.

Ferramentas de operações do

CloudWatch Dashboards

Os painéis do Amazon CloudWatch são páginas personalizáveis no console do CloudWatch que você pode usar para monitorar seus recursos em uma única visualização, mesmo os recursos distribuídos em regiões diferentes. Você pode usar os painéis do CloudWatch para criar visualizações personalizadas das métricas e dos alarmes para os recursos da AWS.

Explorer

O Explorer é um painel de operações personalizável que relata informações sobre seus recursos da AWS. O Explorer exibe uma visualização agregada dos dados de operações (OpsData) para suas Contas da AWS e em todas as Regiões da AWS. No Explorer, os OpsData incluem metadados sobre suas instâncias do Amazon EC2, detalhes de conformidade de patches e itens de trabalho operacionais (OpsItems). O Explorer fornece contexto sobre como os OpsItems são distribuídos em suas unidades de negócios ou aplicações, a tendência ao longo do tempo e como eles variam de acordo com a categoria. Você pode agrupar e filtrar informações no Explorer para se concentrar em itens que são relevantes para você e que exigem ação. Ao identificar problemas de alta prioridade, você pode usar o OpsCenter, uma ferramenta do Systems Manager, para executar runbooks do Automation e resolver rapidamente esses problemas.

Incident Manager

O Incident Manager é um console de gerenciamento de incidentes que ajuda os usuários a reduzir e se recuperar de incidentes que afetam as aplicações hospedadas na AWS.

O Incident Manager aumenta a resolução de incidentes notificando os respondentes sobre o impacto, destacando dados relevantes para a solução de problemas e fornecendo ferramentas de colaboração para colocar os serviços em funcionamento. O Incident Manager também automatiza os planos de resposta e permite o encaminhamento para a equipe de resposta.

OpsCenter

O OpsCenter fornece um local central onde engenheiros de operações e profissionais de IT podem visualizar, investigar e resolver itens de trabalho operacionais (OpsItems) relacionados a recursos da AWS. O OpsCenter foi projetado para reduzir o tempo médio de resolução de

problemas que afetam os recursos da AWS. Essa ferramenta do Systems Manager agrega e padroniza o OpsItems em todos os serviços enquanto fornece dados de investigação contextuais sobre cada OpsItem, OpsItems relacionados e recursos relacionados. O OpsCenter também fornece runbooks do Systems Manager Automation que você pode usar para resolver problemas. Você pode especificar dados personalizados e pesquisáveis para cada OpsItem. Você também pode visualizar relatórios de resumo gerados automaticamente sobre o OpsItems por status e origem.

Executar uma tarefa de nó com o Systems Manager

Use este tutorial para começar a usar o AWS Systems Manager. Você aprenderá como iniciar uma instância do Amazon Elastic Compute Cloud (Amazon EC2) gerenciada pelo Systems Manager e como se conectar à instância gerenciada.

Como o Systems Manager é um conjunto de várias ferramentas, não é possível apresentar todo o serviço com uma única demonstração ou tutorial. Este tutorial fornece uma introdução a algumas das ferramentas.

Pré-requisitos

Antes de começar, é necessário concluir as etapas em Gerenciar instâncias do EC2 com o Systems Manager.

Inicie uma instância usando uma AMI com o SSM Agent pré-instalado

É possível iniciar uma instância do Amazon EC2 usando o AWS Management Console, conforme descrito no procedimento a seguir. Este tutorial tem o objetivo de ajudar você a iniciar sua primeira instância gerenciada rapidamente, portanto, não abrange todas as opções possíveis.

Como iniciar uma instância

1. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. No painel do console do EC2, na caixa Launch instance (Iniciar instância), escolha Launch instance (Iniciar instância) e escolha Launch instance (Iniciar instância) entre as opções exibidas.
3. Em Nome e etiquetas, para Nome, insira um nome descritivo para a instância.
4. Em Imagens de aplicações e SO (imagem de máquina da Amazon), faça o seguinte:

a. Escolha a guia Início rápido e, em seguida, selecione Amazon Linux. Este é o sistema operacional (SO) de sua instância.

b. Em Imagem de máquina da Amazon (AMI), escolha uma versão HVM do Amazon Linux 2.

5. Em Tipo de instância, na lista Tipo de instância, escolha a configuração de hardware para a instância. Escolha o tipo de instância `t2.micro`, que é selecionado por padrão. O tipo de instância `t2.micro` é elegível para o nível gratuito da AWS. Nas Regiões da AWS em que `t2.micro` não está disponível, é possível usar uma instância `t3.micro` no nível gratuito. Para obter mais informações, consulte Nível gratuito da AWS.

6. Em Par de chaves (login), em Nome do par de chaves, escolha um par de chaves.

7. Em Configurações de rede, escolha Editar. Em Nome do grupo de segurança, observe que o assistente criou e selecionou um grupo de segurança para você. É possível usar esse grupo de segurança ou, como alternativa, selecionar um grupo de segurança criado anteriormente usando as etapas a seguir:

   a. Escolha Select existing security group (Selecionar um grupo de segurança existente).

   b. Em Common security groups (Grupos de segurança comuns), escolha o grupo de segurança na lista de grupos de segurança existentes.

8. Se você não estiver usando a configuração de gerenciamento de host padrão, expanda a seção Detalhes avançados e, em Perfil de instância do IAM, escolha o perfil de instância criado ao realizar a configuração na Configurar permissões de instância obrigatórias para o Systems Manager.

9. Mantenha as seleções padrão para outras configurações de sua instância.

10. Analise um resumo da configuração de sua instância no painel Resumo. Quando estiver tudo pronto, escolha Iniciar instância.

11. Uma página de confirmação informará que a instância está sendo iniciada. Escolha View all instances (Visualizar todas as instâncias) para fechar a página de confirmação e voltar ao console.

12. Na tela Instances, é possível visualizar o status da execução. Demora um pouco para executar uma instância.

13. Pode demorar alguns minutos para que a instância apareça como gerenciada e esteja pronta para você se conectar a ela. Para verificar se a instância foi aprovada nas verificações de status, visualize essas informações na coluna Verificação de status.

Conectar-se à sua instância gerenciada usando o Systems Manager

Como se conectar à sua instância gerenciada

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o botão ao lado da instância à qual você deseja se conectar.
4. No menu Ações do nó, escolha Iniciar sessão do terminal.
5. Selecione Connect (Conectar-se).

Limpe sua instância

Se você tiver terminado de trabalhar com a instância gerenciada criada para este tutorial, encerre-a. Encerrar uma instância efetivamente a exclui.

Encerrar sua instância

1. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, escolha Instances (Instâncias). Na lista de instâncias, selecione a instância.
3. Escolha Instance state (Estado da instância) e Terminate instance (Encerrar instância).
4. Quando a confirmação for solicitada, escolha Terminate (Encerrar).

   O Amazon EC2 desliga e encerra sua instância. Após a instância ser encerrada, ela permanecerá visível no console por um breve período e, em seguida, a entrada será excluída automaticamente. Não é possível remover a instância encerrada da exibição do console.

Ferramentas de nós do AWS Systems Manager

O AWS Systems Manager fornece as seguintes ferramentas para acessar, gerenciar e configurar os nós gerenciados. Um nó gerenciado é qualquer máquina configurada para uso com o Systems Manager em um ambiente híbrido e multinuvem.

Tópicos

- AWS Systems Manager Compliance

- AWS Systems Manager Distributor
- AWS Systems Manager Fleet Manager
- Ativações híbridas do AWS Systems Manager
- AWS Systems Manager Inventory
- AWS Systems Manager Patch Manager
- AWS Systems Manager Run Command
- AWS Systems Manager Session Manager
- AWS Systems Manager State Manager

AWS Systems Manager Compliance

Você pode usar o Compliance, uma ferramenta do AWS Systems Manager, para verificar a conformidade dos patches e as inconsistências de configuração em sua frota de nós gerenciados. Você pode coletar e agregar dados da Contas da AWS e regiões, e depois fazer buscas detalhadas em recursos específicos que não forem compatíveis. Por padrão, o Compliance exibe dados de conformidade atuais sobre aplicação de patches no Patch Manager e associações no State Manager. (Patch Manager e State Manager também são ferramentas do AWS Systems Manager). Para começar a usar o Compliance, abra o console do Systems Manager (Gerenciador de sistemas). No painel de navegação, selecione Compliance (Conformidade).

Dados de conformidade de patches do Patch Manager pode ser enviado para oAWS Security Hub. O Security Hub oferece uma visão abrangente dos alertas de segurança de alta prioridade e do status de conformidade. Também monitora o estado de aplicação de patches da sua frota. Para obter mais informações, consulte Integrar o Patch Manager ao AWS Security Hub.

O Compliance oferece os seguintes benefícios e recursos adicionais:

- Visualizar o histórico de conformidade e o controle de alterações para os dados da aplicação de patch do Patch Manager e as associações do State Manager usando o AWS Config.
- Personalizar a conformidade do para criar seus próprios tipos de conformidade com base em seus requisitos de TI ou de negócios.
- Corrija problemas usando o Run Command, outra ferramenta do AWS Systems Manager, o State Manager ou o Amazon EventBridge.
- Faça a portabilidade dos dados para o Amazon Athena e o Amazon QuickSight para gerar relatórios em toda a frota.

Suporte ao EventBridge

Essa ferramenta do Systems Manager pode ser usada como um tipo de evento nas regras do Amazon EventBridge. Para obter informações, consulte Monitorar eventos do Systems Manager com o Amazon EventBridge e Referência: padrões e tipos de eventos do Amazon EventBridge para o Systems Manager.

Integração do Chef InSpec

O Systems Manager integra-se ao Chef InSpec. InSpec é um framework de runtime de código aberto que permite criar perfis legíveis no GitHub ou no Amazon Simple Storage Service (Amazon S3). Você pode então usar o Systems Manager para executar verificações de compatibilidade e visualizar os nós gerenciados compatíveis e não compatíveis. Para obter mais informações, consulte Usar os perfis do Chef InSpec com o Systems Manager Compliance.

Preços

O Compliance é oferecido sem custo adicional. Você paga apenas pelo recursos da AWS que utilizar.

Conteúdo

- Conceitos básicos do Compliance
- Configuração de permissões para Compliance
- Criar uma sincronização de dados de recursos para o Compliance
- Conheça mais sobre a conformidade
- Excluir uma sincronização de dados de recursos para o Compliance
- Corrija problemas de conformidade usando o EventBridge
- Assign custom compliance metadata using the AWS CLI

Conceitos básicos do Compliance

Para começar a usar o Compliance, uma ferramenta do AWS Systems Manager, conclua as tarefas a seguir.

Tarefa	Para obter mais informações
O Compliance funciona com dados de patch no Patch Manager e associações no State	Configurar o console unificado do Systems Manager para uma organização

Tarefa	Para obter mais informações
Manager. (Patch Manager e State Manager também são ferramentas do AWS Systems Manager). O Compliance também funciona com tipos de conformidade personalizados em nós gerenciados que usam o Systems Manager. Verifique se você concluiu os requisitos de configuração para instâncias do Amazon Elastic Compute Cloud (Amazon EC2) e máquinas que não são do EC2 em um ambiente híbrido e multinuvem.	
Atualize o perfil (IAM) AWS Identity and Access Management usado pelos nós gerenciados para restringir as permissões de Compliance.	Configuração de permissões para Compliance
Se você planeja monitorar a conformidade de patches, verifique se configurou o Patch Manager. Você deve realizar operações de aplicação de patches usando o Patch Manager antes que o Compliance possa exibir dados de conformidade do patch.	AWS Systems Manager Patch Manager
Se você planeja monitorar a conformidade de associações, verifique se criou associações do State Manager. Você deve criar associações para que o Compliance possa exibir os dados de conformidade da associação.	AWS Systems Manager State Manager
(Opcional) Configure o sistema para visualizar o histórico de conformidade e o controle de alterações.	Visualizar o histórico de configuração da conformidade e do controle de alterações
(Opcional) Crie tipos de conformidade personalizados.	Assign custom compliance metadata using the AWS CLI

Tarefa	Para obter mais informações
(Opcional) Crie uma sincronização de dados de recursos para agregar todos os dados de conformidade em um bucket do Amazon Simple Storage Service (Amazon S3) de destino.	Criar uma sincronização de dados de recursos para o Compliance

Configuração de permissões para Compliance

Como prática recomendada de segurança, recomendamos que você atualize o perfil (IAM) AWS Identity and Access Management usado pelos nós gerenciados com as seguintes permissões para restringir a capacidade do nó de usar a ação da API PutComplianceItems. Essa ação de API registra um tipo de conformidade e outros detalhes de conformidade em um recurso designado, como uma instância do Amazon EC2 ou um nó gerenciado.

Se o seu nó for uma instância do Amazon EC2, você deverá atualizar o perfil de instância do IAM usado pela instância com as seguintes permissões. Para obter mais informações sobre perfis de instância para instâncias EC2 gerenciadas pelo Systems Manager, consulte Configurar permissões de instância obrigatórias para o Systems Manager. Para outros tipos de nós gerenciados, atualize o perfil do IAM usada pelo nó com as seguintes permissões. Para obter mais informações, consulte Atualizar permissões para um perfil no Guia do usuário do IAM.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutComplianceItems"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:SourceInstanceARN": "${ec2:SourceInstanceARN}"
                }
            }
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutComplianceItems"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ssm:SourceInstanceARN": "${ssm:SourceInstanceARN}"
                }
            }
        }
    ]
}
```

Criar uma sincronização de dados de recursos para o Compliance

Você pode usar o recurso de sincronização de dados de recursos no AWS Systems Manager para enviar dados de conformidade de todos os nós gerenciados para um bucket do Amazon Simple Storage Service (Amazon S3) de destino. Ao criar a sincronização, você pode especificar nós gerenciados de várias Contas da AWS, Regiões da AWS e do ambiente híbrido e multinuvem. Em seguida, a sincronização de dados de recursos atualiza automaticamente os dados centralizados quando novos dados de conformidade forem coletados. Com todos os dados de conformidade armazenados em um bucket do S3 de destino, você pode usar serviços como o Amazon Athena e Amazon QuickSight para consultar e analisar os dados agregados. Configurar a sincronização de dados de recursos para o Compliance é uma operação única.

Use o procedimento a seguir para criar uma sincronização de dados de recursos para o Compliance usando o AWS Management Console.

Como criar e configurar um bucket do S3 para sincronização de dados de recursos (console)

1. Abra o console do Amazon S3, em https://console.aws.amazon.com/s3/.
2. Crie um bucket para armazenar os dados de conformidade agregados. Para obter mais informações, consulte Criar um bucket no Guia do usuário do Amazon Simple Storage Service. Anote o nome do bucket e a Região da AWS em que você o criou.
3. Abra o bucket, escolha a guia Permissions (Permissões) e selecione Bucket Policy (Política de bucket).

4. Copie e cole a seguinte política de bucket no editor de políticas. Substitua amzn-s3-demo-bucket e *Account-ID* pelo nome do bucket do S3 que você criou e um ID de Conta da AWS válido. Opcionalmente, substitua *Bucket-Prefix* pelo nome de um prefixo do Amazon S3 (subdiretório). Se você não criou um prefixo, remova *Bucket-Prefix*/ do ARN na política.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SSMBucketPermissionsCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
        },
        {
            "Sid": " SSMBucketDelivery",
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket/Bucket-Prefix/*/
accountid=Account_ID_number/*"],
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control"
                }
            }
        }
    ]
}
```

Para criar uma sincronização de dados de recurso

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.

2. No painel de navegação, escolha Fleet Manager.
3. Escolha Account management (Gerenciamento de contas), Resource Data Syncs (Sincronizações de dados do recurso) e Create resource data sync (Criar sincronização de dados do recurso).
4. No campo Sync name (Nome da sincronização), digite um nome para a configuração de sincronização.
5. No campo Bucket name (Nome do bucket), digite o nome do bucket do Amazon S3 que você criou no início desse procedimento.
6. (Opcional) No campo Prefixo do bucket, insira o nome de um prefixo de bucket do S3 (subdiretório).
7. No campo Região do bucket, escolha Esta região se o bucket do S3 que você criou estiver localizado na Região da AWS atual. Se o bucket estiver localizado em uma Região da AWS diferente, escolha Another region (Outra região) e digite o nome da região.

> **Note**
>
> Se a sincronização e o bucket S3 de destino estiverem localizados em regiões diferentes, você poderá estar sujeito à precificação de transferência de dados. Para obter mais informações, consulte Preços do Amazon S3.

8. Escolha Criar.

Conheça mais sobre a conformidade

O Compliance, uma ferramenta do AWS Systems Manager, coleta e relata dados sobre o status da aplicação de patches no Patch Manager, patches e associações no State Manager (Patch Manager e State Manager também são ferramentas do AWS Systems Manager). O Compliance também relata os tipos de conformidade personalizados que você especificou para os nós gerenciados. Esta seção inclui detalhes sobre cada um desses tipos de conformidade e sobre como visualizar os dados de conformidade do Systems Manager. Esta seção também inclui informações sobre como visualizar o histórico de conformidade e o controle de alterações.

> **Note**
>
> O Systems Manager integra-se ao Chef InSpec. InSpec é um framework de runtime de código aberto que permite criar perfis legíveis no GitHub ou no Amazon Simple Storage

Service (Amazon S3). Em seguida, você pode usar o Systems Manager para executar verificações de compatibilidade e visualizar instâncias compatíveis e não compatíveis. Para obter mais informações, consulte Usar os perfis do Chef InSpec com o Systems Manager Compliance.

Sobre a conformidade de patches

Após usar o Patch Manager para instalar patches em suas instâncias, as informações do status de conformidade ficarão imediatamente disponíveis no console ou em resposta aos comandos da AWS Command Line Interface (AWS CLI) ou a ações de API correspondentes do Systems Manager.

Para obter informações sobre valores de status de conformidade de patches, consulte Valores de estados de conformidade de patches.

Sobre a conformidade de associações do State Manager

Após criar uma ou mais associações do State Manager, as informações de status de conformidade ficam imediatamente disponíveis no console ou em resposta a comandos da AWS CLI ou a ações de API correspondentes do Systems Manager. Para associações, o Compliance mostrará os status de `Compliant` ou `Non-compliant` e o nível de gravidade atribuído à associação, como `Critical` ou `Medium`.

Quando o State Manager executa uma associação em um nó gerenciado, ele aciona um processo de agregação de conformidade que atualiza o status de conformidade de todas as associações nesse nó. O valor de `ExecutionTime` nos relatórios de conformidade representa quando o status de conformidade foi capturado pelo Systems Manager, não quando a associação foi executada no nó gerenciado. Isso significa que várias associações podem exibir valores de `ExecutionTime` idênticos, mesmo que tenham sido executadas em momentos diferentes. Para determinar os tempos reais de execução da associação, consulte o histórico de execução da associação usando o comando describe-association-execution-targets da AWS CLI ou visualizando os detalhes da execução no console.

Sobre a conformidade personalizada

Você pode atribuir metadados de conformidade a um nó gerenciado. Esses metadados podem ser agregados com outros dados de conformidade para fins de relatórios de conformidade. Por exemplo, digamos que sua empresa execute as versões 2.0, 3.0 e 4.0 do software X em seus nós gerenciados. A empresa deseja padronizar na versão 4.0, o que significa que as instâncias

que executam as versões 2.0 e 3.0 não são compatíveis. É possível usar a operação da API PutComplianceItems para observar explicitamente quais nós gerenciados estão executando versões antigas do software X. Só é possível atribuir metadados de conformidade usando a AWS CLI, o AWS Tools for Windows PowerShell ou os SDKs. O seguinte comando de exemplo da CLI atribui metadados de conformidade a uma instância gerenciada e especifica o tipo de conformidade no formato necessário `Custom:`. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

Linux & macOS

```
aws ssm put-compliance-items \
    --resource-id i-1234567890abcdef0 \
    --resource-type ManagedInstance \
    --compliance-type Custom:SoftwareXCheck \
    --execution-summary ExecutionTime=AnyStringToDenoteTimeOrDate \
    --items
 Id=Version2.0,Title=SoftwareXVersion,Severity=CRITICAL,Status=NON_COMPLIANT
```

Windows

```
aws ssm put-compliance-items ^
    --resource-id i-1234567890abcdef0 ^
    --resource-type ManagedInstance ^
    --compliance-type Custom:SoftwareXCheck ^
    --execution-summary ExecutionTime=AnyStringToDenoteTimeOrDate ^
    --items
 Id=Version2.0,Title=SoftwareXVersion,Severity=CRITICAL,Status=NON_COMPLIANT
```

> **Note**
>
> O parâmetro `ResourceType` só suporta o `ManagedInstance`. Se você adicionar conformidade personalizada a um dispositivo principal do AWS IoT Greengrass gerenciado, você deverá especificar um `ResourceType` do `ManagedInstance`.

Os gerentes do Compliance podem então visualizar resumos ou criar relatórios sobre quais instâncias são ou não são compatíveis. Você pode atribuir um máximo de 10 tipos de conformidade personalizados diferentes a um nó gerenciado.

Para obter um exemplo de como criar um tipo de conformidade personalizado e visualizar dados de conformidade, consulte Assign custom compliance metadata using the AWS CLI.

Visualizar dados da conformidade atual

Esta seção descreve como visualizar os dados de conformidade no console do Systems Manager usando a AWS CLI. Para obter informações sobre como visualizar o histórico de conformidade de patches e de associações e o controle de alterações, consulte Visualizar o histórico de configuração da conformidade e do controle de alterações.

Tópicos

- Visualizar dados da conformidade atual (console)
- Visualizar dados da conformidade atual (AWS CLI)

Visualizar dados da conformidade atual (console)

Use o procedimento a seguir para visualizar dados de conformidade no console do Systems Manager.

Para visualizar relatórios de conformidade atuais no console do Systems Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, selecione Compliance
3. Na seção Compliance dashboard filtering (Filtragem do painel de conformidade), escolha uma opção para filtrar dados de conformidade. A seção Compliance resources summary (Resumo dos recursos de conformidade) exibe a contagem de dados de conformidade com base no filtro que você escolheu.
4. Para analisar detalhadamente um recurso a fim de obter mais informações, role para baixo até a área Details overview for resources (Visão geral dos detalhes dos recursos) e escolha o ID de um nó gerenciado.
5. Na página Instance ID (ID da instância) ou Name (Nome), selecione a guia Configuration compliance (Conformidade de configuração) para exibir o relatório detalhado de conformidade da configuração para o nó gerenciado.

> **Note**
>
> Para obter mais informações sobre soluções de problemas com conformidade, consulte Corrija problemas de conformidade usando o EventBridge.

Visualizar dados da conformidade atual (AWS CLI)

Você pode visualizar resumos de dados de conformidade de aplicação de patches, associações e tipos de conformidade personalizados na AWS CLI usando os seguintes comandos da AWS CLI.

list-compliance-summaries

retorna uma contagem de resumo de status de associação compatíveis e não compatíveis, de acordo com o filtro que você especificou. (API: ListComplianceSummaries)

list-resource-compliance-summaries

Retorna uma contagem de resumo no nível de recursos. O resumo inclui informações sobre status compatíveis e não compatíveis, bem como contagens detalhadas de severidade de itens de conformidade, de acordo com os critérios de filtro que você especifica. (API: ListResourceComplianceSummaries)

Você pode visualizar dados de conformidade adicionais de aplicação de patches executando os seguintes comandos na AWS CLI.

describe-patch-group-state

retorna o estado de conformidade de patches agregados de alto nível para um grupo de patches. (API: DescribePatchGroupState)

describe-instance-patch-states-for-patch-group

retorna o estado do patch de alto nível para as instâncias no grupo de patches especificado. (API: DescribeInstancePatchStatesForPatchGroup)

> **Note**
>
> Para ver uma ilustração de como configurar a aplicação de patches e visualizar os detalhes de conformidade de patches usando a AWS CLI, consulte Tutorial: Aplicar patches a um ambiente de servidor usando a AWS CLI.

Visualizar o histórico de configuração da conformidade e do controle de alterações

O Systems Manager Compliance exibe a aplicação de patches e associações atuais dos nós gerenciados. Você pode visualizar o histórico de conformidade da aplicação de patches e de associações e o controle de alterações usando o AWS Config. O AWS Config fornece uma visão detalhada da configuração dos recursos da AWS em sua conta da Conta da AWS. Isso inclui como os recursos estão relacionados um com o outro e como eles foram configurados no passado, de modo que você possa ver como os relacionamentos e as configurações foram alterados ao longo do tempo. Para visualizar o histórico de conformidade da aplicação de patches e das associações e controle de alterações, você deve ativar os seguintes recursos no AWS Config:

- `SSM:PatchCompliance`
- `SSM:AssociationCompliance`

Para obter mais informações sobre como escolher e configurar esses recursos específicos no AWS Config, consulte Selecionar que recursos o AWS Config registra no Manual do desenvolvedor do AWS Config.

> **Note**
>
> Para obter mais informações sobre a definição de preço do AWS Config, consulte Definição de preço do .

Excluir uma sincronização de dados de recursos para o Compliance

Caso você não queira mais usar o AWS Systems Manager Compliance para exibir dados de conformidade, recomendamos também excluir as sincronizações de dados de recursos usados para a coleta de dados do Compliance.

Para excluir uma sincronização de dados de recursos do Compliance

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione Account management (Gerenciamento de contas) e Resource data syncs (Sincronizações de dados do recurso).
4. Na lista, escolha um sincronização.

> **Important**
>
> Escolha a sincronização usada para o Compliance. O Systems Manager oferece suporte à sincronização de dados de recursos para várias ferramentas. Se você escolher a sincronização errada, a agregação de dados para o Systems Manager ou para o Systems Manager Inventory poderá ser interrompida.

5. Escolha Excluir.
6. Exclua o bucket do Amazon Simple Storage Service (Amazon S3) no qual os dados foram armazenados. Para obter informações sobre como excluir um bucket do S3, consulte Excluir um bucket.

Corrija problemas de conformidade usando o EventBridge

É possível corrigir problemas de conformidade de patches e associações rapidamente usando o Run Command, uma ferramenta do AWS Systems Manager. Você pode segmentar a instância ou os IDs ou etiquetas do dispositivo principal do AWS IoT Greengrass e executar o documento `AWS-RunPatchBaseline` ou o `AWS-RefreshAssociation`. Se a atualização da associação ou a nova execução da lista de referência de patches não resolver o problema de conformidade, você precisará investigar suas associações, listas de referência de patches ou configurações da instância para entender por que as execuções do Run Command não resolveram o problema.

Para obter mais informações sobre aplicação de patches, consulte AWS Systems Manager Patch Manager e Documento de comando do SSM para a aplicação de patches: `AWS-RunPatchBaseline`.

Para obter mais informações sobre associações, consulte Para obter informações, consulte Trabalhar com associações no Systems Manager..

Para obter mais informações sobre como executar um comando, consulte AWS Systems Manager Run Command.

Especifique o Compliance como o destino de um evento do EventBridge

Você também pode configurar o Amazon EventBridge para executar uma ação em resposta a eventos do Systems Manager Compliance. Por exemplo, se um ou mais nós gerenciados não conseguirem instalar atualizações de patch críticas ou executar uma associação que instale um software antivírus, você poderá configurar o EventBridge para executar o documento `AWS-RunPatchBaseline` ou `AWS-RefreshAssocation` quando o evento do Compliance ocorrer.

Use o procedimento a seguir para configurar o Compliance como o destino de um evento do EventBridge.

Para configurar o Compliance como o destino de um evento do EventBridge (console)

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Regras.
3. Escolha Create rule.
4. Insira um nome e uma descrição para a regra.

   Uma regra não pode ter o mesmo nome que outra regra na mesma Região da AWS e no mesmo barramento de eventos.
5. Em Barramento de Eventos, escolha o barramento de eventos que você deseja associar a essa regra. Se você quiser que essa regra responda a eventos correspondentes provenientes da sua Conta da AWS, selecione default (padrão). Quando um AWS service (Serviço da AWS) na sua conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta.
6. Em Tipo de Regra, escolha Regra com Padrão de Evento.
7. Escolha Próximo.
8. Em Event source, escolha Eventos da AWS ou eventos de parceiro do EventBridge.
9. Na seção Event patter (Padrão de evento), selecione Event pattern form (Formulário de padrão de evento).
10. Em Fonte do evento, selecione Serviços da AWS.
11. Em Serviço da AWS, escolha Systems Manager.
12. Em Event Type (Tipo de evento), escolha Configuration Compliance (Conformidade das configurações).

13. Em Specific detail type(s) (Tipos de detalhes específicos), escolha Configuration Compliance State Change (Alteração de estado de conformidade da configuração).
14. Escolha Next (Avançar).
15. Em Tipos de destino, escolha Serviço da AWS.
16. Em Select a target (Selecionar um destino), escolha Systems Manager Run Command.
17. Na lista Document (Documento), escolha um documento do Systems Manager (Documento SSM) para executar quando seu destino for invocado. Por exemplo, escolha `AWS-RunPatchBaseline` para um evento de patch fora de conformidade ou escolha `AWS-RefreshAssociation` para um evento de associação fora de conformidade.
18. Especifique informações para os campos e parâmetros restantes.

> **Note**
>
> Campos e parâmetros obrigatórios têm um asterisco (*) ao lado do nome. Para criar um destino, você deve especificar um valor para cada parâmetro ou campo obrigatório. Se não fizer isso, o sistema criará a regra, mas a regra não será executada.

19. Escolha Próximo.
20. (Opcional) Insira uma ou mais tags para a regra. Para obter mais informações, consulte Marcar recursos do Amazon EventBridge no Guia do usuário do Amazon EventBridge.
21. Escolha Próximo.
22. Analise os detalhes da regra e selecione Criar regra.

Assign custom compliance metadata using the AWS CLI

O procedimento a seguir envolve o processo de usar o AWS Command Line Interface (AWS CLI) para chamar a operação de API PutComplianceItems do AWS Systems Manager para atribuir metadados de conformidade personalizados a um recurso. Você também pode usar essa operação de API para atribuir manualmente os metadados de conformidade de associações ou patches a um nó gerenciado, conforme mostrado na demonstração a seguir. Para obter mais informações sobre conformidade personalizada, consulte Sobre a conformidade personalizada.

Para atribuir metadados de conformidade personalizados a uma instância gerenciada (AWS CLI)

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Execute o seguinte comando para atribuir metadados de conformidade personalizados a um nó gerenciado. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações. O parâmetro `ResourceType` só suporta um valor de `ManagedInstance`. Especifique esse valor mesmo se você estiver atribuindo metadados de conformidade personalizados a um dispositivo principal do AWS IoT Greengrass gerenciado.

   Linux & macOS

   ```
   aws ssm put-compliance-items \
       --resource-id instance_ID \
       --resource-type ManagedInstance \
       --compliance-type Custom:user-defined_string \
       --execution-summary ExecutionTime=user-defined_time_and/or_date_value \
       --items Id=user-defined_ID,Title=user-
   defined_title,Severity=one_or_more_comma-separated_severities:CRITICAL, MAJOR,
    MINOR,INFORMATIONAL, or UNSPECIFIED,Status=COMPLIANT or NON_COMPLIANT
   ```

   Windows

   ```
   aws ssm put-compliance-items ^
       --resource-id instance_ID ^
       --resource-type ManagedInstance ^
       --compliance-type Custom:user-defined_string ^
       --execution-summary ExecutionTime=user-defined_time_and/or_date_value ^
       --items Id=user-defined_ID,Title=user-
   defined_title,Severity=one_or_more_comma-separated_severities:CRITICAL, MAJOR,
    MINOR,INFORMATIONAL, or UNSPECIFIED,Status=COMPLIANT or NON_COMPLIANT
   ```

3. Repita a etapa anterior para atribuir metadados de conformidade personalizados adicionais a um ou mais nós. Você também pode atribuir manualmente metadados de conformidade de patches ou de associação aos nós gerenciados usando os seguintes comandos:

   Metadados de conformidade de associações

   Linux & macOS

   ```
   aws ssm put-compliance-items \
       --resource-id instance_ID \
       --resource-type ManagedInstance \
   ```

```
    --compliance-type Association \
    --execution-summary ExecutionTime=user-defined_time_and/or_date_value \
    --items Id=user-defined_ID,Title=user-
defined_title,Severity=one_or_more_comma-separated_severities:CRITICAL, MAJOR,
 MINOR,INFORMATIONAL, or UNSPECIFIED,Status=COMPLIANT or NON_COMPLIANT
```

Windows

```
aws ssm put-compliance-items ^
    --resource-id instance_ID ^
    --resource-type ManagedInstance ^
    --compliance-type Association ^
    --execution-summary ExecutionTime=user-defined_time_and/or_date_value ^
    --items Id=user-defined_ID,Title=user-
defined_title,Severity=one_or_more_comma-separated_severities:CRITICAL, MAJOR,
 MINOR,INFORMATIONAL, or UNSPECIFIED,Status=COMPLIANT or NON_COMPLIANT
```

Metadados de conformidade de patches

Linux & macOS

```
aws ssm put-compliance-items \
    --resource-id instance_ID \
    --resource-type ManagedInstance \
    --compliance-type Patch \
    --execution-summary ExecutionTime=user-defined_time_and/
or_date_value,ExecutionId=user-defined_ID,ExecutionType=Command  \
    --items Id=for_example, KB12345,Title=user-
defined_title,Severity=one_or_more_comma-separated_severities:CRITICAL,
 MAJOR, MINOR,INFORMATIONAL, or UNSPECIFIED,Status=COMPLIANT or
 NON_COMPLIANT,Details="{PatchGroup=name_of_group,PatchSeverity=the_patch_severity,
 for example, CRITICAL}"
```

Windows

```
aws ssm put-compliance-items ^
    --resource-id instance_ID ^
    --resource-type ManagedInstance ^
    --compliance-type Patch ^
    --execution-summary ExecutionTime=user-defined_time_and/
or_date_value,ExecutionId=user-defined_ID,ExecutionType=Command  ^
```

```
    --items Id=for_example, KB12345,Title=user-
defined_title,Severity=one_or_more_comma-separated_severities:CRITICAL,
 MAJOR, MINOR,INFORMATIONAL, or UNSPECIFIED,Status=COMPLIANT or
 NON_COMPLIANT,Details="{PatchGroup=name_of_group,PatchSeverity=the_patch_severity,
 for example, CRITICAL}"
```

4. Execute o seguinte comando para visualizar uma lista de itens de conformidade para uma instância gerenciada específica. Use filtros para busca detalhada em dados de conformidade específicos.

   Linux & macOS

   ```
   aws ssm list-compliance-items \
       --resource-ids instance_ID \
       --resource-types ManagedInstance \
       --filters one_or_more_filters
   ```

   Windows

   ```
   aws ssm list-compliance-items ^
       --resource-ids instance_ID ^
       --resource-types ManagedInstance ^
       --filters one_or_more_filters
   ```

   Os exemplos a seguir mostram como usar esse comando com filtros.

   Linux & macOS

   ```
   aws ssm list-compliance-items \
       --resource-ids i-02573cafcfEXAMPLE \
       --resource-type ManagedInstance \
       --filters Key=DocumentName,Values=AWS-RunPowerShellScript
    Key=Status,Values=NON_COMPLIANT,Type=NotEqual
    Key=Id,Values=cee20ae7-6388-488e-8be1-a88ccEXAMPLE
    Key=Severity,Values=UNSPECIFIED
   ```

   Windows

   ```
   aws ssm list-compliance-items ^
       --resource-ids i-02573cafcfEXAMPLE ^
   ```

```
    --resource-type ManagedInstance ^
    --filters Key=DocumentName,Values=AWS-RunPowerShellScript
 Key=Status,Values=NON_COMPLIANT,Type=NotEqual
 Key=Id,Values=cee20ae7-6388-488e-8be1-a88ccEXAMPLE
 Key=Severity,Values=UNSPECIFIED
```

Linux & macOS

```
aws ssm list-resource-compliance-summaries \
    --filters Key=OverallSeverity,Values=UNSPECIFIED
```

Windows

```
aws ssm list-resource-compliance-summaries ^
    --filters Key=OverallSeverity,Values=UNSPECIFIED
```

Linux & macOS

```
aws ssm list-resource-compliance-summaries \
    --filters Key=OverallSeverity,Values=UNSPECIFIED
 Key=ComplianceType,Values=Association Key=InstanceId,Values=i-02573cafcfEXAMPLE
```

Windows

```
aws ssm list-resource-compliance-summaries ^
    --filters Key=OverallSeverity,Values=UNSPECIFIED
 Key=ComplianceType,Values=Association Key=InstanceId,Values=i-02573cafcfEXAMPLE
```

5. Execute o seguinte comando para visualizar um resumo dos status de conformidade. Use filtros para busca detalhada em dados de conformidade específicos.

```
aws ssm list-resource-compliance-summaries --filters One or more filters.
```

Os exemplos a seguir mostram como usar esse comando com filtros.

Linux & macOS

```
aws ssm list-resource-compliance-summaries \
```

```
    --filters Key=ExecutionType,Values=Command
```

Windows

```
aws ssm list-resource-compliance-summaries ^
    --filters Key=ExecutionType,Values=Command
```

Linux & macOS

```
aws ssm list-resource-compliance-summaries \
    --filters Key=AWS:InstanceInformation.PlatformType,Values=Windows
 Key=OverallSeverity,Values=CRITICAL
```

Windows

```
aws ssm list-resource-compliance-summaries ^
    --filters Key=AWS:InstanceInformation.PlatformType,Values=Windows
 Key=OverallSeverity,Values=CRITICAL
```

6. Execute o seguinte comando para visualizar uma contagem de resumo de recursos compatíveis e não compatíveis para um tipo de conformidade. Use filtros para busca detalhada em dados de conformidade específicos.

```
aws ssm list-compliance-summaries --filters One or more filters.
```

Os exemplos a seguir mostram como usar esse comando com filtros.

Linux & macOS

```
aws ssm list-compliance-summaries \
    --filters Key=AWS:InstanceInformation.PlatformType,Values=Windows
 Key=PatchGroup,Values=TestGroup
```

Windows

```
aws ssm list-compliance-summaries ^
    --filters Key=AWS:InstanceInformation.PlatformType,Values=Windows
 Key=PatchGroup,Values=TestGroup
```

Linux & macOS

```
aws ssm list-compliance-summaries \
    --filters Key=AWS:InstanceInformation.PlatformType,Values=Windows
 Key=ExecutionId,Values=4adf0526-6aed-4694-97a5-14522EXAMPLE
```

Windows

```
aws ssm list-compliance-summaries ^
    --filters Key=AWS:InstanceInformation.PlatformType,Values=Windows
 Key=ExecutionId,Values=4adf0526-6aed-4694-97a5-14522EXAMPLE
```

AWS Systems Manager Distributor

O Distributor, uma ferramenta do AWS Systems Manager, ajuda você a empacotar e publicar software em nós gerenciados do AWS Systems Manager. Você pode empacotar e publicar seu próprio software ou usar o Distributor para encontrar e publicar pacotes de software de agente fornecidos pela AWS, como o AmazonCloudWatchAgent, ou pacotes de terceiros, como Trend Micro.Publicar um pacote anuncia versões específicas do documento do pacote para nós gerenciados que você identifica usando IDs de nó, IDs da Conta da AWS, etiquetas ou uma Região da AWS. Para começar a usar o Distributor, abra o Systems Manager console (Console do gerenciador de sistemas). No painel de navegação, escolha Distributor.

Depois de criar um pacote no Distributor, você poderá instalar o pacote de uma das seguintes formas:

- Uma vez, usando AWS Systems Manager Run Command
- Em uma programação, usando AWS Systems Manager State Manager

> **Important**
>
> Os pacotes distribuídos por vendedores terceirizados não são gerenciados pela AWS. Eles são publicados pelo fornecedor do pacote. Recomendamos realizar diligências adicionais para garantir a conformidade com seus controles de segurança internos. A segurança é uma responsabilidade compartilhada entre a AWS e você. Isso é descrito como o modelo de

responsabilidade compartilhada da. Para saber mais, consulte o modelo de responsabilidade compartilhada.

Como o Distributor beneficia minha organização?

Distributor oferece estes benefícios:

- Um pacote, várias plataformas

  Quando você cria um pacote no Distributor, o sistema cria um documento AWS Systems Manager (documento do SSM). Você pode anexar arquivos .zip a este documento. Quando você executa o Distributor, o sistema processa as instruções no documento SSM e instala o pacote de software no arquivo.zip nos destinos especificados. O Distributor é oferece suporte a vários sistemas operacionais, incluindo Windows, Ubuntu Server, Debian Server e Red Hat Enterprise Linux. Para obter mais informações sobre as plataformas compatíveis, consulte Plataformas de pacotes e arquiteturas compatíveis.

- Controlar acesso do pacote entre grupos de instâncias gerenciadas

  É possível usar o Run Command ou o State Manager para controlar quais dos nós gerenciados obtêm um pacote e qual é a versão desse pacote. Run Command e State Manager são ferramentas do AWS Systems Manager. Os nós gerenciados podem ser agrupados por IDs de instância, números de Conta da AWS, etiquetas ou Regiões da AWS. Você pode usar associações State Manager para fornecer diferentes versões de um pacote para diferentes grupos de instâncias.

- Muitos pacotes do agente AWS inclusos e prontos para uso

  O Distributor inclui vários pacotes do agente AWS prontos para a implantação em nós gerenciados. Procure por pacotes na página de listas do Distributor e `Packages` que são publicadas pela `Amazon`. Exemplos incluem `AmazonCloudWatchAgent` e `AWSPVDriver`.

- Automatizar implantação

  Para manter seu ambiente atualizado, use o State Manager para programar a implantação automática de pacotes em nós gerenciados de destino quando eles forem executados pela primeira vez.

Quem deve usar o Distributor?

- Todos os clientes da AWS que quiserem criar novos pacotes de software ou implantar pacotes existentes, incluindo pacotes publicados pela AWS, em vários nós gerenciados do Systems Manager de uma só vez.
- Desenvolvedores de software que criam pacotes de software.
- Administradores responsáveis por manter os nós gerenciados do Systems Manager atualizados com os mais recentes pacotes de software.

Quais são os recursos do Distributor?

- Implantação de pacotes para instâncias Windows e Linux

  Com o Distributor, você pode implantar pacotes de software em instâncias do Amazon Elastic Compute Cloud (Amazon EC2) e dispositivos principais do AWS IoT Greengrass para Linux e Windows Server. Para obter uma lista dos tipos de sistema operacional de instâncias compatíveis, consulte the section called “Plataformas de pacotes e arquiteturas compatíveis”.

  > **Note**
  >
  > O Distributor não tem suporte no sistema operacional macOS.

- Implante pacotes uma vez ou em uma programação automatizada

  Você pode optar por implantar pacotes uma vez, em uma programação normal, ou sempre que a versão do pacote padrão for alterada para uma versão diferente.

- Reinstale completamente os pacotes ou execute atualizações no local

  Para instalar uma nova versão de pacote, você pode desinstalar completamente a versão atual e instalar uma nova em seu lugar, ou apenas atualizar a versão atual com componentes novos e atualizados, de acordo com um script de atualização fornecido. Seu aplicativo de pacote não está disponível durante uma reinstalação, mas pode permanecer disponível durante uma atualização no local. As atualizações no local são especialmente úteis para aplicativos de monitoramento de segurança ou outros cenários em que você precisa evitar o tempo de inatividade do aplicativo.

- Acesso por Console, CLI, PowerShell e SDK aos recursos do Distributor

Você pode trabalhar com o Distributor usando o console do Systems Manager, o AWS Command Line Interface (AWS CLI), o Ferramentas da AWS para PowerShell ou o SDK da AWS de sua escolha.

- Controle de acesso do IAM

  Ao usar as políticas do AWS Identity and Access Management (IAM), você poderá controlar quais membros da organização poderão criar, atualizar, implantar ou excluir pacotes ou versões de pacote. Por exemplo, você pode querer dar a um administrador permissões para implantar pacotes, mas não para alterar pacotes ou criar novas versões de pacote.

- Registro de auditoria de suporte a recursos

  Você pode auditar e registrar ações do usuário do Distributor em sua Conta da AWS por meio da integração com outros Serviços da AWS. Para obter mais informações, consulte Auditar e registrar atividades do Distributor em log.

O que é um pacote no Distributor?

Um pacote é uma coleção de softwares instaláveis ou ativos que inclui o seguinte.

- Um arquivo .zip do software por plataforma de sistema operacional de destino. Cada arquivo .zip deve incluir o seguinte.
  - Um script install e uninstall. Os nós gerenciados com base no Windows Server exigem scripts do PowerShell (scripts chamados `install.ps1` e `uninstall.ps1`). Os nós gerenciados baseados em Linux exigem scripts de shell (scripts denominados `install.sh` e `uninstall.sh`). AWS Systems Manager SSM Agent leem e executam as instruções nos scripts install e uninstall.
  - Um arquivo executável. O SSM Agent deve encontrar este executável para instalar o pacote em nós gerenciados de destino.
- Um arquivo manifesto em formato JSON que descreve o conteúdo do pacote. O manifesto não está incluído no arquivo .zip, mas está armazenado no mesmo bucket do Amazon Simple Storage Service (Amazon S3) que os arquivos .zip que compõem o pacote. O manifesto identifica a versão do pacote e mapeia os arquivos .zip no pacote para direcionar os atributos do nó gerenciado, como a versão do sistema operacional ou arquitetura. Para obter informações sobre como criar o manifesto, consulte Etapa 2: Criar o manifesto do pacote JSON.

Ao escolher um pacote Simple (Simples) no console do Distributor, o Distributor gera a os scripts de instalação e desinstalação, hashes de arquivo e o manifesto do pacote JSON para você, com base no nome de arquivo executável do software e as plataformas e arquiteturas de destino.

Plataformas de pacotes e arquiteturas compatíveis

Você pode usar o Distributor para publicar pacotes nas seguintes plataformas de nós gerenciados do Systems Manager. Um valor de versão deve corresponder à versão exata do sistema operacional da Amazon Machine Image (AMI) de destino. Para obter mais informações sobre como determinar esta versão, consulte o passo 4 do Etapa 2: Criar o manifesto do pacote JSON.

> **Note**
>
> O Systems Manager não oferece suporte a todos os sistemas operacionais a seguir para dispositivos principais do AWS IoT Greengrass. Para obter mais informações, consulte Configurar dispositivos principais do AWS IoT Greengrass no Guia do desenvolvedor do AWS IoT Greengrass Version 2.

Plataforma	Valor de código no arquivo manifesto	Arquiteturas compatíveis
AlmaLinux	`almalinux`	x86_64 ARM64
Amazon Linux 1, Amazon Linux 2 e Amazon Linux 2023	`amazon`	x86_64 ou x86 ARM64 (Amazon Linux 2 e AL2023, tipos de instância A1)
CentOS	`centos`	x86_64 ou x86
Debian Server	`debian`	x86_64 ou x86
openSUSE	`opensuse`	x86_64 ou x86
openSUSE Leap	`opensuseleap`	x86_64 ou x86
Oracle Linux	`oracle`	x86_64

Plataforma	Valor de código no arquivo manifesto	Arquiteturas compatíveis
Red Hat Enterprise Linux (RHEL)	`redhat`	x86_64 ou x86 ARM64 (RHEL 7.6 e posterior, tipos de instância A1)
Rocky Linux	`rocky`	x86_64 ARM64
SUSE Linux Enterprise Server (SLES)	`suse`	x86_64 ou x86
Ubuntu Server	`ubuntu`	x86_64 ou x86 ARM64 (Ubuntu Server 16 e posterior, tipos de instância A1)
Windows Server	`windows`	x86_64 ou x86

Tópicos

- Configurar o Distributor
- Trabalhando com pacotes do Distributor
- Auditar e registrar atividades do Distributor em log
- Solução de problemas do AWS Systems ManagerDistributor

Configurar o Distributor

Antes de usar o Distributor, uma ferramenta do AWS Systems Manager, para criar, gerenciar e implantar pacotes de software, siga estas etapas.

Concluir os pré-requisitos do Distributor

Antes de usar o Distributor, uma ferramenta do AWS Systems Manager, verifique se seu ambiente atende aos requisitos a seguir.

Pré-requisitos da Distributor

Requisito	Descrição
SSM Agent	O AWS Systems Manager SSM Agent versão 2.3.274.0 ou posterior deve ser instalado em nós gerenciados nos quais você deseja implantar ou dos quais deseja remover pacotes. Para instalar ou atualizar o SSM Agent, consulte Trabalhar com o SSM Agent.
AWS CLI	(Opcional) Para usar a AWS Command Line Interface (AWS CLI) em vez do console do Systems Manager para criar e gerenciar pacotes, instale a versão mais recente da AWS CLI em seu computador local. Para obter mais informações sobre como instalar ou atualizar a CLI, consulte Instalar a AWS Command Line Interface no Manual do usuário da AWS Command Line Interface.
Ferramentas da AWS para PowerShell	(Opcional) Para usar o Tools for PowerShell em vez do console do Systems Manager para criar e gerenciar pacotes, instale a versão mais recente do Tools for PowerShell em seu computador local. Para obter mais informações sobre como instalar ou atualizar as ferramentas para PowerShell, consulte Instalar o AWS Tools for Windows PowerShell ou o AWS Tools for PowerShell Core, no Guia do usuário do Ferramentas da AWS para PowerShell.

> **Note**
>
> O Systems Manager não oferece suporte à distribuição de pacotes para nós gerenciados do Oracle Linux usando o Distributor.

Verificar ou criar um perfil de instância do IAM com permissões do Distributor

Por padrão, o AWS Systems Manager não tem permissão para executar ações em suas instâncias. Você deve conceder acesso usando um perfil de instância do AWS Identity and Access Management (IAM). Um perfil de instância é um contêiner que transmite as informações da função do IAM para uma instância do Amazon Elastic Compute Cloud (Amazon EC2) na inicialização. Esse requisito se aplica a permissões para todas as ferramentas do Systems Manager, e não apenas ao Distributor.

> **Note**
>
> Quando você configura seus dispositivos de borda para executar o software principal doAWS IoT Greengrass e SSM Agent, você especifica uma função de serviço do IAM que permite que o Systems Manager realize ações nela. Não é necessário configurar dispositivos de borda gerenciados com um perfil da instância.

Se você já usa outras ferramentas do Systems Manager, como Run Command e State Manager, um perfil de instância com as permissões necessárias para Distributor já está anexado às suas instâncias. A maneira mais simples de garantir que você tenha permissões para executar tarefas do Distributor é anexar a política AmazonSSMManagedInstanceCore ao seu perfil de instância. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager.

Controlar acesso do usuário aos pacotes

Usando políticas do AWS Identity and Access Management (IAM), você controla quem pode criar, implantar e gerenciar pacotes. Você também pode controlar quais operações de API do Run Command e do State Manager podem ser executadas em seus nós gerenciados. Assim como o Distributor, tanto o Run Command quanto o State Manager são ferramentas disponíveis no AWS Systems Manager.

Formato do ARN

Os pacotes definidos pelo usuário são associados aos nomes do recurso da Amazon (ARNs) do documento e têm o seguinte formato:

```
arn:aws:ssm:region:account-id:document/document-name
```

Veja um exemplo a seguir.

```
arn:aws:ssm:us-west-1:123456789012:document/ExampleDocumentName
```

Você pode usar um par de políticas do IAM padrão fornecidas pela AWS, uma para os usuários finais e outra para os administradores, para conceder permissões para atividades do Distributor. Ou você pode criar políticas IAM personalizadas e apropriadas para suas necessidades de permissões.

Para obter mais informações sobre usar variáveis em políticas IAM, consulte Elementos da política do IAM: variáveis.

Para obter informações sobre como criar políticas e anexá-las a usuários ou grupos, consulte Criação de políticas do IAM e Adição e remoção de políticas do IAM no Guia do usuário do IAM.

Criar ou escolher um bucket do Amazon S3 para armazenar pacotes do Distributor

Ao criar um pacote usando o fluxo de trabalho Simples no console do AWS Systems Manager, você escolhe um bucket do Amazon Simple Storage Service (Amazon S3) existente para o qual o Distributor fará upload do seu software. O Distributor é uma ferramenta do AWS Systems Manager. No fluxo de trabalho Advanced (Avançado), é necessário fazer upload de arquivos .zip do seu software ou ativos em um bucket do Amazon S3 antes de começar. Se você criar um pacote usando os fluxos de trabalho Simple (Simples) ou Advanced (Avançado) no console, ou usando a API, você precisará ter um bucket do Amazon S3 antes de começar a criar o pacote. Como parte do processo de criação do pacote, o Distributor copia o software e ativos instaláveis desse bucket em um armazenamento interno do Systems Manager. Como os ativos são copiados em um armazenamento interno, você poderá excluir ou redefinir seu bucket do Amazon S3 quando a criação do pacote for concluída.

Para obter mais informações sobre como criar um bucket, consulte Criar um bucket no Manual de conceitos básicos do Amazon Simple Storage Service. Para obter mais informações sobre como executar um comando AWS CLI para criar um bucket, consulte mb na AWS CLI Referência de comando.

Trabalhando com pacotes do Distributor

Você pode usar o console do AWS Systems Manager, as ferramentas de linha de comando da AWS (AWS CLI e Ferramentas da AWS para PowerShell) e os AWS SDKs para adicionar, gerenciar ou implantar pacotes no Distributor. O Distributor é uma ferramenta do AWS Systems Manager. Antes de adicionar um pacote a Distributor:

- Crie e faça um zip dos ativos instaláveis.
- (Opcional) Crie um arquivo manifesto JSON para o pacote. Não é necessário fazer isso no processo de criação do pacote Simple (Simples) no console do Distributor. Uma criação de pacote simples gera um arquivo manifesto JSON para você.

  Você pode usar o console do AWS Systems Manager ou um editor de texto ou JSON para criar o arquivo manifesto.
- Tenha um bucket do Amazon Simple Storage Service (Amazon S3) pronto para armazenar ativos ou software instaláveis. Se você estiver usando o processo de criação de pacote Advanced (Avançado), faça upload dos ativos no bucket do Amazon S3, antes de começar.

> **Note**
>
> Você pode excluir ou redefinir esse bucket depois de terminar de criar o pacote, porque o Distributor transfere o conteúdo do pacote para um bucket interno do Systems Manager como parte do processo de criação dele.

Os pacotes publicados pela AWS já estão empacotados e prontos para implantação. Para implantar um pacote publicado pela AWS para nós gerenciados, consulte Instalar ou atualizar pacotes do Distributor.

Você pode compartilhar pacotes do Distributor entre Contas da AWS. Ao usar um pacote compartilhado de outra conta nos comandos da AWS CLI, use o nome do recurso da Amazon (ARN) do pacote em vez do nome do pacote.

Tópicos

- Visualizar pacotes no Distributor
- Criar um pacote no Distributor
- Editar permissões do pacote do Distributor no console

- Editar tags de um pacote do Distributor no console
- Adicionar uma versão a um pacote do Distributor
- Instalar ou atualizar pacotes do Distributor
- Desinstalar um pacote do Distributor
- Excluir um pacote do Distributor

Visualizar pacotes no Distributor

Para visualizar os pacotes que estão disponíveis para instalação, você pode usar o console do AWS Systems Manager ou a ferramenta de linha de comando da AWS de sua preferência. O Distributor é uma ferramenta do AWS Systems Manager. Para acessar o Distributor, abra o console do AWS Systems Manager e escolha Distributor no painel de navegação à esquerda. Você verá todos os pacotes disponíveis para você.

A seção a seguir descreve como você pode exibir os pacotes do Distributor usando sua ferramenta de linha de comando preferida.

Visualizar pacotes usando a linha de comando

Esta seção contém informações sobre como usar a ferramenta de linha de comando de sua preferência para visualizar os pacotes do Distributor usando os comandos fornecidos.

Linux & macOS

Para visualizar pacotes usando a AWS CLI no Linux

- Para visualizar todos os pacotes, excluindo pacotes compartilhados, execute o comando a seguir.

```
aws ssm list-documents \
    --filters Key=DocumentType,Values=Package
```

- Para visualizar todos os pacotes da Amazon, execute o comando a seguir.

```
aws ssm list-documents \
    --filters Key=DocumentType,Values=Package Key=Owner,Values=Amazon
```

- Para visualizar todos os pacotes de propriedade de terceiros, execute o comando a seguir.

```
aws ssm list-documents \
```

```
    --filters Key=DocumentType,Values=Package Key=Owner,Values=ThirdParty
```

Windows

Para visualizar pacotes usando a AWS CLI no Windows

- Para visualizar todos os pacotes, excluindo pacotes compartilhados, execute o comando a seguir.

```
aws ssm list-documents ^
    --filters Key=DocumentType,Values=Package
```

- Para visualizar todos os pacotes da Amazon, execute o comando a seguir.

```
aws ssm list-documents ^
    --filters Key=DocumentType,Values=Package Key=Owner,Values=Amazon
```

- Para visualizar todos os pacotes de propriedade de terceiros, execute o comando a seguir.

```
aws ssm list-documents ^
    --filters Key=DocumentType,Values=Package Key=Owner,Values=ThirdParty
```

PowerShell

Para visualizar pacotes usando o Tools for PowerShell

- Para visualizar todos os pacotes, excluindo pacotes compartilhados, execute o comando a seguir.

```
$filter = New-Object Amazon.SimpleSystemsManagement.Model.DocumentKeyValuesFilter
$filter.Key = "DocumentType"
$filter.Values = "Package"

Get-SSMDocumentList `
    -Filters @($filter)
```

- Para visualizar todos os pacotes da Amazon, execute o comando a seguir.

```
$typeFilter = New-Object
 Amazon.SimpleSystemsManagement.Model.DocumentKeyValuesFilter
```

```
$typeFilter.Key = "DocumentType"
$typeFilter.Values = "Package"

$ownerFilter = New-Object
 Amazon.SimpleSystemsManagement.Model.DocumentKeyValuesFilter
$ownerFilter.Key = "Owner"
$ownerFilter.Values = "Amazon"

Get-SSMDocumentList `
    -Filters @($typeFilter,$ownerFilter)
```

- Para visualizar todos os pacotes de propriedade de terceiros, execute o comando a seguir.

```
$typeFilter = New-Object
 Amazon.SimpleSystemsManagement.Model.DocumentKeyValuesFilter
$typeFilter.Key = "DocumentType"
$typeFilter.Values = "Package"

$ownerFilter = New-Object
 Amazon.SimpleSystemsManagement.Model.DocumentKeyValuesFilter
$ownerFilter.Key = "Owner"
$ownerFilter.Values = "ThirdParty"

Get-SSMDocumentList `
    -Filters @($typeFilter,$ownerFilter)
```

Criar um pacote no Distributor

Para criar um pacote, prepare o software ou ativos instaláveis, um arquivo por plataforma do sistema operacional. É necessário pelo menos um arquivo para criar um pacote.

Plataformas diferentes às vezes podem usar o mesmo arquivo, mas todos os arquivos que você anexa ao seu pacote devem estar listados na seção `Files` do manifesto. Se você estiver criando um pacote usando o fluxo de trabalho simples no console, o manifesto será gerado para você. Você pode anexar um máximo de 20 arquivos a um único documento. O tamanho máximo de cada arquivo é 1 GB. Para obter mais informações sobre as plataformas compatíveis, consulte Plataformas de pacotes e arquiteturas compatíveis.

Quando você cria um pacote, o sistema cria um documento SSM. O documento permite que você implante o pacote em nós gerenciados.

Apenas para fins de demonstração, um pacote de exemplo, ExamplePackage.zip, está disponível para download em nosso site. O pacote de exemplo inclui um manifesto JSON concluído e três arquivos .zip contendo instaladores do PowerShell v7.0.0. Os scripts de instalação e desinstalação não contêm comandos válidos. É preciso compactar cada software instalável e scripts em um arquivo .zip para criar um pacote no fluxo de trabalho Advanced (Avançado), mas não compacte os ativos instaláveis no fluxo de trabalho Simple (Simples).

Tópicos

- Criar um pacote usando o fluxo de trabalho Simples
- Criar um pacote usando o fluxo de trabalho Avançado

Criar um pacote usando o fluxo de trabalho Simples

Esta seção descreve como criar um pacote no Distributor escolhendo o fluxo de trabalho de criação Simples no console do Distributor. O Distributor é uma ferramenta do AWS Systems Manager. Para criar um pacote, prepare seus ativos instaláveis, um arquivo por plataforma do sistema operacional. É necessário pelo menos um arquivo para criar um pacote. O processo de criação de pacote Simple (Simples) gera os scripts de instalação e instalação, os hashes de arquivo e um manifesto em formato JSON para você. O fluxo de trabalho Simple (Simples) cuida do processo de upload e compactação dos seus arquivos de instalação e cria um pacote e um documento SSM associado. Para obter mais informações sobre as plataformas compatíveis, consulte Plataformas de pacotes e arquiteturas compatíveis.

Quando você usa o método Simple para criar um pacote, o Distributor cria scripts `install` e `uninstall` para você. No entanto, ao criar um pacote para uma atualização no local, é necessário fornecer seu próprio conteúdo de script `update` na guia Update script (Script de atualização). Quando você adiciona comandos de entrada para um script `update`, o Distributor inclui esse script no pacote .zip criado para você, junto com os scripts `install` e `uninstall`.

> **Note**
>
> Use a opção de atualização `In-place` para adicionar arquivos novos ou atualizados a uma instalação de pacote existente sem deixar a aplicação associada offline.

Para criar um pacote usando o fluxo de trabalho Simples

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.

2. No painel de navegação, escolha Distributor.
3. Na página inicial do Distributor, selecione Create package (Criar pacote) e, então, Simple (Simples).
4. Na página Create package (Criar pacote), digite um nome para seu pacote. Nomes de pacotes podem conter letras, números, pontos, traços e sublinhados. O nome deve ser genérico o suficiente para se aplicar a todas as versões dos anexos do pacote, mas específicos o suficiente para identificar a finalidade do pacote.
5. (Opcional) Em Version name (Nome da versão), insira um nome para a versão. Os nomes de versões podem ter no máximo 512 caracteres e não devem conter caracteres especiais.
6. Em Location (Localização), escolha um bucket usando o nome e o prefixo do bucket ou usando o URL do bucket.
7. Em Upload software (Fazer upload de software), escolha Add software (Adicionar software) e procure os arquivos de software instaláveis com as extensões `.rpm`, `.msi` ou `.deb`. Se o nome do arquivo contiver espaços, o upload falhará. Você pode fazer upload de mais de um arquivo de software com uma única ação.
8. Em Target platform (Plataforma de destino), verifique se a plataforma do sistema operacional de destino exibida para cada arquivo de instalação está correta. Se o sistema operacional correto não for o exibido, escolha o correto na lista suspensa.

   Para o fluxo de trabalho de criação do pacote Simple (Simples), como você faz upload de cada arquivo de instalação somente uma vez, são necessárias etapas adicionais para que o Distributor direcione um único arquivo para mais de um sistema operacional. Por exemplo, se você carregar um arquivo de software instalável chamado `Logtool_v1.1.1.rpm`, você precisará alterar alguns valores padrão no fluxo de trabalho Simple (Simples) para direcionar o mesmo software aos dois sistemas operacionais, Amazon Linux e Ubuntu. Ao segmentar várias plataformas, siga um destes procedimentos.

   - Use o fluxo de trabalho Advanced (Avançado), compacte cada arquivo de instalação em um arquivo .zip antes de começar e crie o manifesto manualmente de modo que aquele arquivo de instalação possa ser direcionado a mais de uma plataforma ou versão de sistema operacional. Para obter mais informações, consulte Criar um pacote usando o fluxo de trabalho Avançado.
   - Edite manualmente o arquivo manifesto no fluxo de trabalho Simple (Simples) para que seu arquivo .zip seja direcionado a mais de uma plataforma ou versão de sistema operacional. Para obter mais informações sobre como fazer isso, consulte o final da etapa 4 em Etapa 2: Criar o manifesto do pacote JSON.

9. Em Platform version (Versão da plataforma), verifique se a versão exibida da plataforma do sistema operacional é **_any**, uma versão importante seguida pelo curinga (7.*) ou a versão específica do sistema operacional ao qual você deseja que o software se aplique. Para obter mais informações sobre como especificar uma versão de plataforma do sistema operacional, consulte a etapa 4 em Etapa 2: Criar o manifesto do pacote JSON.
10. Em Architecture (Arquitetura) escolha, na lista suspensa, a arquitetura do processador correta para cada arquivo de instalação. Para obter mais informações sobre as arquiteturas de processador compatíveis, consulte Plataformas de pacotes e arquiteturas compatíveis.
11. (Opcional) Expanda Scripts e revise os scripts que o Distributor gerou para seu software instalável.
12. (Opcional) Para fornecer um script de atualização para uso com atualizações no local, expanda Scripts, escolha a guia Update script (Script de atualização) e insira os comandos de script de atualização.

    O Systems Manager não gera scripts de atualização em seu nome.
13. Para adicionar mais arquivos de software instalável, escolha Add software (Adicionar software). Senão, siga para a próxima etapa.
14. (Opcional) Expanda Manifest (Manifesto) e revise o manifesto do pacote JSON que o Distributor gerou para o software instalável. Se você tiver alterado qualquer informação sobre seu software desde que começou esse procedimento, como a versão da plataforma ou plataforma de destino, escolha Generate manifest (Gerar manifesto) para exibir o manifesto do pacote atualizado.

    Você pode editar o manifesto manualmente se quiser direcionar o software instalável para mais de um sistema operacional, conforme descrito na etapa 8. Para obter mais informações sobre a edição de manifesto, consulte Etapa 2: Criar o manifesto do pacote JSON.
15. Escolha a opção Criar pacote.

Aguarde até que o Distributor termine de carregar o software e criar seu pacote. O Distributor mostra o status do carregamento para cada arquivo de instalação. Dependendo do número e do tamanho dos pacotes que você estiver adicionando, isso pode levar alguns minutos. O Distributor automaticamente redirecionará você para a página Package details (Detalhes do pacote) do novo pacote, mas você poderá optar por abri-lo depois que o software for carregado. A página Package details (Detalhes do pacote) só exibirá todas as informações sobre ele quando o Distributor finalizar o processo de criação. Para interromper o processo de criação e upload do pacote, escolha Cancel (Cancelar).

Se o Distributor não conseguir fazer upload de qualquer um dos arquivos de instalação do software, ele exibirá a mensagem Upload failed (Falha no upload). Para tentar fazer upload novamente, escolha Retry upload (Tentar novamente). Para obter mais informações sobre como solucionar as falhas de criação do pacote, consulte Solução de problemas do AWS Systems ManagerDistributor.

Criar um pacote usando o fluxo de trabalho Avançado

Nesta seção, saiba como usuários avançados podem criar um pacote no Distributor após fazer upload de ativos instaláveis compactados com scripts de instalação e desinstalação e um arquivo manifesto JSON em um bucket do Amazon S3.

Para criar um pacote, prepare seus arquivos .zip de ativos instaláveis, um arquivo .zip por plataforma do sistema operacional. Pelo menos um arquivo .zip é necessário para criar um pacote. Em seguida, crie um manifesto JSON. O manifesto inclui ponteiros para arquivos de código do seu pacote. Quando você tiver os arquivos de código necessários adicionados a uma pasta ou diretório, e o manifesto tiver sido preenchido com os valores corretos, faça upload do pacote em um bucket do S3.

Um exemplo de pacote, ExamplePackage.zip, está disponível para download no nosso site. O exemplo inclui um manifesto JSON concluído e três arquivos .zip.

Tópicos

- Etapa 1: Criar os arquivos ZIP
- Etapa 2: Criar o manifesto do pacote JSON
- Etapa 3: Fazer upload do pacote e do manifesto em um bucket do Amazon S3
- Etapa 4: Adicionar um pacote ao Distributor

Etapa 1: Criar os arquivos ZIP

A base do seu pacote é pelo menos um arquivo .zip de software ou ativos instaláveis. Um pacote inclui um arquivo .zip por sistema operacional ao qual você deseja oferecer suporte, a menos que um arquivo .zip possa ser instalado em vários sistemas operacionais. Por exemplo, as instâncias do Red Hat Enterprise Linux e do Amazon Linux normalmente podem executar os mesmos arquivos .RPM executáveis, então você precisa anexar apenas um arquivo .zip ao seu pacote para oferecer suporte aos dois sistemas operacionais.

Arquivos necessários

Os seguintes itens são necessários em cada arquivo .zip:

- Um script install e uninstall. Os nós gerenciados com base no Windows Server exigem scripts do PowerShell (scripts chamados `install.ps1` e `uninstall.ps1`). Os nós gerenciados baseados em Linux exigem scripts de shell (scripts denominados `install.sh` e `uninstall.sh`). O SSM Agent executa as instruções nos scripts install e uninstall.

  Por exemplo, os scripts de instalação podem executar um instalador (como .rpm ou .msi), podem copiar arquivos ou podem definir configurações.
- Um arquivo executável, pacotes do instalador (.rpm, .deb, .msi etc.), outros scripts ou arquivos de configuração.

Arquivos opcionais

O seguinte item é opcional em cada arquivo .zip:

- Um script update. Fornecer um script de atualização possibilita o uso da opção `In-place update` para instalar um pacote. Para adicionar arquivos novos ou atualizados a uma instalação de pacote existente, a opção `In-place update` não coloca a aplicação de pacote offline enquanto a atualização é executada. Os nós gerenciados com base no Windows Server requerem um script do PowerShell (chamado `update.ps1`). Os nós gerenciados baseados em Linux exigem um script de shell (script chamado `update.sh`). O SSM Agent executa as instruções no script update.

Para obter mais informações sobre como instalar ou atualizar pacotes, consulte Instalar ou atualizar pacotes do Distributor.

Para obter exemplos de arquivos .zip, incluindo scripts de amostra install e uninstall, faça download do pacote de exemplos, ExamplePackage.zip.

Etapa 2: Criar o manifesto do pacote JSON

Depois de preparar e compactar os arquivos de instalação, crie um manifesto JSON. A seguir está um modelo. As partes do modelo manifesto estão descritas no procedimento nesta seção. Você pode usar um editor JSON para criar esse manifesto em um arquivo separado. Como alternativa, você pode criar o manifesto no console do AWS Systems Manager ao criar um pacote.

```
{
  "schemaVersion": "2.0",
  "version": "your-version",
  "publisher": "optional-publisher-name",
```

```
  "packages": {
    "platform": {
      "platform-version": {
        "architecture": {
          "file": ".zip-file-name-1.zip"
        }
      }
    },
    "another-platform": {
      "platform-version": {
        "architecture": {
          "file": ".zip-file-name-2.zip"
        }
      }
    },
    "another-platform": {
      "platform-version": {
        "architecture": {
          "file": ".zip-file-name-3.zip"
        }
      }
    }
  },
  "files": {
    ".zip-file-name-1.zip": {
      "checksums": {
        "sha256": "checksum"
      }
    },
    ".zip-file-name-2.zip": {
      "checksums": {
        "sha256": "checksum"
      }
    }
  }
}
```

Para criar um manifesto de pacote JSON

1. Adicione a versão do esquema ao seu manifesto. Nesta versão, a versão de esquema é sempre `2.0`.

```
{ "schemaVersion": "2.0",
```

2. Adicione uma versão do pacote definido pelo usuário ao manifesto. Isso também é o valor de Version name (Nome da versão) que você especifica ao adicionar o pacote ao Distributor. Ele se torna parte do documento do AWS Systems Manager que o Distributor cria quando você adiciona seu pacote. Você também pode fornecer esse valor como uma entrada no documento `AWS-ConfigureAWSPackage` para instalar uma versão do pacote que não seja a última. Um valor `version` pode conter letras, números, sublinhados, hifens e pontos, e ter no máximo 128 caracteres. Recomendamos que você use uma versão legível do pacote para facilitar que você e outros administradores especifiquem versões de pacote exatas ao implantar. Veja um exemplo a seguir.

   ```
   "version": "1.0.1",
   ```

3. (Opcional) Adicione um nome de editor. Veja um exemplo a seguir.

   ```
   "publisher": "MyOrganization",
   ```

4. Adicionar pacotes. A seção `"packages"` descreve as plataformas, versões e arquiteturas compatíveis com os arquivos .zip em seu pacote. Para obter mais informações, consulte Plataformas de pacotes e arquiteturas compatíveis.

   O *`platform-version`* pode ser o valor curinga `_any`. Use-o para indicar que um arquivo .zip é compatível com qualquer versão da plataforma. Você também pode especificar uma versão principal seguida de um curinga para que todas as versões secundárias sejam suportadas, por exemplo 7.*. Se você optar por especificar um valor da *`platform-version`* para uma versão específica do sistema operacional, certifique-se de que ele corresponde à versão exata da AMI do sistema operacional que você definir como destino. Veja a seguir os recursos sugeridos para obter o valor correto do sistema operacional.

   - Em nós gerenciados baseados no Windows Server, a versão está disponível como dados do Windows Management Instrumentation (WMI). Você pode executar o seguinte comando em um prompt de comando para obter informações sobre a versão e depois analisar os resultados da `version`.

     ```
     wmic OS get /format:list
     ```

- Em um nó gerenciado baseado em Linux, obtenha a versão primeiramente verificando a versão do sistema operacional (o comando a seguir). Procure o valor de `VERSION_ID`.

  ```
  cat /etc/os-release
  ```

  Se isso não retornar os resultados necessários, execute o seguinte comando para obter informações de versão do LSB do arquivo `/etc/lsb-release` e procure o valor de `DISTRIB_RELEASE`.

  ```
  lsb_release -a
  ```

  Se esses métodos falharem, você pode geralmente encontrar a versão com base na distribuição. Por exemplo, no Debian Server, você pode analisar o arquivo `/etc/debian_version`, ou no Red Hat Enterprise Linux, o arquivo `/etc/redhat-release`.

  ```
  hostnamectl
  ```

```
"packages": {
    "platform": {
      "platform-version": {
        "architecture": {
          "file": ".zip-file-name-1.zip"
        }
      }
    },
    "another-platform": {
      "platform-version": {
        "architecture": {
          "file": ".zip-file-name-2.zip"
        }
      }
    },
    "another-platform": {
      "platform-version": {
        "architecture": {
          "file": ".zip-file-name-3.zip"
        }
      }
    }
```

```
  }
```

Veja um exemplo a seguir. Neste exemplo, a plataforma do sistema operacional é `amazon`, a versão compatível é `2016.09`, a arquitetura é `x86_64`, e o arquivo .zip que oferece suporte a essa plataforma é `test.zip`.

```
{
    "amazon": {
        "2016.09": {
            "x86_64": {
                "file": "test.zip"
            }
        }
    }
},
```

Você pode adicionar o valor curinga `_any` para indicar que o pacote é compatível com todas as versões do elemento pai. Por exemplo, para indicar que o pacote é compatível com qualquer versão do Amazon Linux, sua declaração de pacote deve ser semelhante ao seguinte: Você pode usar o curinga `_any` nos níveis de versão ou arquitetura para oferecer suporte a todas as versões de uma plataforma, ou todas as arquiteturas em uma versão, ou todas as versões e todas as arquiteturas de uma plataforma.

```
{
    "amazon": {
        "_any": {
            "x86_64": {
                "file": "test.zip"
            }
        }
    }
},
```

O exemplo a seguir adiciona `_any` para mostrar que o primeiro pacote, `data1.zip`, é compatível com todas as arquiteturas do Amazon Linux 2016.09. O segundo pacote, `data2.zip`, tem suporte para todas as versões do Amazon Linux, mas apenas para nós gerenciados com a arquitetura `x86_64`. Tanto a versão `2016.09` quanto a versão `_any` são entradas em `amazon`. Há uma plataforma (Amazon Linux), mas diferentes versões, arquiteturas e arquivos .zip associados compatíveis.

```
{
    "amazon": {
        "2016.09": {
            "_any": {
                "file": "data1.zip"
            }
        },
        "_any": {
            "x86_64": {
                "file": "data2.zip"
            }
        }
    }
}
```

Você pode fazer referência a um arquivo .zip mais de uma vez na seção "packages" do manifesto, se o arquivo .zip oferece suporte a mais de uma plataforma. Por exemplo, se você tiver um arquivo .zip que oferece suporte ao Red Hat Enterprise Linux versões 7.x e ao Amazon Linux, você terá duas entradas na seção "packages" apontando para o mesmo arquivo .zip, conforme mostrado no exemplo a seguir.

```
{
    "amazon": {
        "2018.03": {
            "x86_64": {
                "file": "test.zip"
            }
        }
    },
    "redhat": {
        "7.*": {
            "x86_64": {
                "file": "test.zip"
            }
        }
    }
},
```

5. Adicione a lista de arquivos .zip que fazem parte deste pacote na etapa 4. Cada entrada de arquivo exige o nome do arquivo e a soma de verificação do valor de hash do sha256. Os

valores da soma de verificação no manifesto devem corresponder ao valor de hash `sha256` nos ativos compactados para impedir que a instalação do pacote falhe.

Para obter a soma de verificação exata dos instaláveis, você pode executar os comandos a seguir. No Linux, execute `shasum -a 256 file-name.zip` ou `openssl dgst -sha256 file-name.zip`. No Windows, execute o cmdlet `Get-FileHash -Path path-to-.zip-file` no PowerShell.

A seção `"files"` do manifesto inclui uma referência para cada um dos arquivos .zip em seu pacote.

```
"files": {
        "test-agent-x86.deb.zip": {
            "checksums": {
                "sha256":
 "EXAMPLE2706223c7616ca9fb28863a233b38e5a23a8c326bb4ae241dcEXAMPLE"
            }
        },
        "test-agent-x86_64.deb.zip": {
            "checksums": {
                "sha256":
 "EXAMPLE572a745844618c491045f25ee6aae8a66307ea9bff0e9d1052EXAMPLE"
            }
        },
        "test-agent-x86_64.nano.zip": {
            "checksums": {
                "sha256":
 "EXAMPLE63ccb86e830b63dfef46995af6b32b3c52ce72241b5e80c995EXAMPLE"
            }
        },
        "test-agent-rhel5-x86.nano.zip": {
            "checksums": {
                "sha256":
 "EXAMPLE13df60aa3219bf117638167e5bae0a55467e947a363fff0a51EXAMPLE"
            }
        },
        "test-agent-x86.msi.zip": {
            "checksums": {
                "sha256":
 "EXAMPLE12a4abb10315aa6b8a7384cc9b5ca8ad8e9ced8ef1bf0e5478EXAMPLE"
            }
        },
```

```
        "test-agent-x86_64.msi.zip": {
            "checksums": {
                "sha256":
 "EXAMPLE63ccb86e830b63dfef46995af6b32b3c52ce72241b5e80c995EXAMPLE"
            }
        },
        "test-agent-rhel5-x86.rpm.zip": {
            "checksums": {
                "sha256":
 "EXAMPLE13df60aa3219bf117638167e5bae0a55467e947a363fff0a51EXAMPLE"
            }
        },
        "test-agent-rhel5-x86_64.rpm.zip": {
            "checksums": {
                "sha256":
 "EXAMPLE7ce8a2c471a23b5c90761a180fd157ec0469e12ed38a7094d1EXAMPLE"
            }
        }
    }
```

6. Depois de adicionar as informações do seu pacote, salve e feche o arquivo manifesto.

Veja a seguir um exemplo de um manifesto concluído. Neste exemplo, você tem um arquivo .zip, `NewPackage_LINUX.zip`, que oferece suporte a mais de uma plataforma, mas é referenciada na seção `"files"` somente uma vez.

```
{
    "schemaVersion": "2.0",
    "version": "1.7.1",
    "publisher": "Amazon Web Services",
    "packages": {
        "windows": {
            "_any": {
                "x86_64": {
                    "file": "NewPackage_WINDOWS.zip"
                }
            }
        },
        "amazon": {
            "_any": {
                "x86_64": {
                    "file": "NewPackage_LINUX.zip"
                }
```

```
            }
        },
        "ubuntu": {
            "_any": {
                "x86_64": {
                    "file": "NewPackage_LINUX.zip"
                }
            }
        }
    },
    "files": {
        "NewPackage_WINDOWS.zip": {
            "checksums": {
                "sha256":
 "EXAMPLEc2c706013cf8c68163459678f7f6daa9489cd3f91d52799331EXAMPLE"
            }
        },
        "NewPackage_LINUX.zip": {
            "checksums": {
                "sha256":
 "EXAMPLE2b8b9ed71e86f39f5946e837df0d38aacdd38955b4b18ffa6fEXAMPLE"
            }
        }
    }
}
```

Exemplo de pacote

Um exemplo de pacote, ExamplePackage.zip, está disponível para download no nosso site. O exemplo inclui um manifesto JSON concluído e três arquivos .zip.

Etapa 3: Fazer upload do pacote e do manifesto em um bucket do Amazon S3

Prepare seu pacote, copiando ou movendo todos os arquivos .zip para uma pasta ou diretório. Um pacote válido requer o manifesto criado em Etapa 2: Criar o manifesto do pacote JSON e todos os arquivos .zip identificados na lista do arquivo manifesto.

Para carregar o pacote e o manifesto no Amazon S3

1. Copiar ou mover todos os arquivos de arquivamento .zip especificados no manifesto para uma pasta ou diretório. Não compacte a pasta ou o diretório para o qual você move seus arquivos .zip e o arquivo de manifesto.

2. Crie um bucket ou escolha um bucket existente. Para obter mais informações, consulte Criar um bucket no Manual de conceitos básicos do Amazon Simple Storage Service. Para obter mais informações sobre como executar um comando AWS CLI para criar um bucket, consulte mb na AWS CLI Referência de comando.
3. Carregue a pasta ou diretório no bucket. Para obter mais informações, consulte Adicionar um objeto a um bucket no Guia de conceitos básicos do Amazon Simple Storage Service. Se você planeja colar o manifesto JSON no console do AWS Systems Manager, não faça upload do manifesto. Para obter mais informações sobre como executar um comando AWS CLI para fazer upload de arquivos a um bucket, consulte mv na AWS CLI Referência de comando.
4. Na página inicial do bucket, escolha a pasta ou diretório que você carregou. Se você tiver feito upload dos seus arquivos para uma subpasta em um bucket, anote a subpasta (também conhecida como um prefixo). Você precisará do prefixo para adicionar seu pacote ao Distributor.

Etapa 4: Adicionar um pacote ao Distributor

Você pode usar o console do AWS Systems Manager, as ferramentas de linha de comando da AWS (AWS CLI e Ferramentas da AWS para PowerShell) ou os AWS SDKs para adicionar um novo pacote do Distributor. Ao adicionar um pacote, você está adicionando um novo documento do SSM. O documento permite que você implante o pacote em nós gerenciados.

Tópicos

- Adicionar um pacote usando o console
- Adicionar um pacote usando a AWS CLI

Adicionar um pacote usando o console

Você pode usar o console AWS Systems Manager para criar um pacote. Tenha o nome do bucket para o qual você fez upload do pacote no Etapa 3: Fazer upload do pacote e do manifesto em um bucket do Amazon S3.

Para adicionar um pacote ao Distributor (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Distributor.
3. Na página inicial do Distributor, selecione Create package (Criar pacote) e escolha Advanced (Avançado).

4. Na página Create package (Criar pacote), digite um nome para seu pacote. Nomes de pacotes podem conter letras, números, pontos, traços e sublinhados. O nome deve ser genérico o suficiente para se aplicar a todas as versões dos anexos do pacote, mas específicos o suficiente para identificar a finalidade do pacote.
5. Em Version name (Nome da versão), insira o valor exato da entrada `version` no arquivo manifesto.
6. Em S3 bucket name (Nome do bucket do S3), escolha o nome do bucket para o qual você fez upload do manifesto e dos arquivos .zip no the section called “Etapa 3: Fazer upload do pacote e do manifesto em um bucket do Amazon S3”.
7. Em S3 key prefix (Prefixo de chave do S3), insira a subpasta do bucket em que os arquivos .zip e o manifesto estão armazenados.
8. Em Manifest (Manifesto), escolha Extract from package (Extrair do pacote) para usar um manifesto carregado no bucket do Amazon S3 com os arquivos .zip.

   (Opcional) Se você não fez upload do manifesto JSON no bucket do S3 em que armazenou os arquivos .zip, escolha New manifest (Novo manifesto). Você pode criar ou colar o manifesto inteiro no campo de edição JSON. Para obter mais informações sobre como criar o manifesto JSON, consulte Etapa 2: Criar o manifesto do pacote JSON.
9. Quando terminar o manifesto, escolha Create package (Criar pacote).
10. Aguarde até o Distributor criar seu pacote com os arquivos .zip e o manifesto. Dependendo do número e do tamanho dos pacotes que você está adicionando, isso pode levar alguns minutos. O Distributor automaticamente redirecionará você para a página Package details (Detalhes do pacote) do novo pacote, mas você pode optar por abri-la após o software ser carregado. A página Package details (Detalhes do pacote) só exibirá todas as informações sobre ele quando o Distributor finalizar o processo de criação. Para interromper o processo de criação e upload do pacote, escolha Cancel (Cancelar).

Adicionar um pacote usando a AWS CLI

Você pode usar o AWS CLI para criar um pacote. Deixe o URL pronto do bucket para o qual você fez upload do pacote no Etapa 3: Fazer upload do pacote e do manifesto em um bucket do Amazon S3.

Para adicionar um pacote ao Amazon S3 usando a AWS CLI

1. Para usar a AWS CLI para criar um pacote, execute o seguinte comando, substituindo *`package-name`* pelo nome do pacote e *`path-to-manifest-file`* pelo caminho do arquivo

de manifesto JSON. amzn-s3-demo-bucket é o URL do bucket do Amazon S3 onde todo o pacote está armazenado. Ao executar o comando create-document no Distributor, você especifica o valor `Package` para `--document-type`.

Se você não adicionou o arquivo de manifesto ao bucket do Amazon S3, o valor do parâmetro `--content` será o caminho do arquivo de manifesto JSON.

```
aws ssm create-document \
    --name "package-name" \
    --content file://path-to-manifest-file \
    --attachments Key="SourceUrl",Values="amzn-s3-demo-bucket" \
    --version-name version-value-from-manifest \
    --document-type Package
```

Veja um exemplo a seguir.

```
aws ssm create-document \
    --name "ExamplePackage" \
    --content file://path-to-manifest-file \
    --attachments Key="SourceUrl",Values="https://s3.amazonaws.com/amzn-s3-demo-
bucket/ExamplePackage" \
    --version-name 1.0.1 \
    --document-type Package
```

2. Verifique se o pacote foi adicionado e exiba o manifesto do pacote executando o seguinte comando, substituindo *package-name* pelo nome do seu pacote. Para obter uma versão específica do documento (não o mesmo que a versão de um pacote), você pode adicionar o parâmetro `--document-version`.

```
aws ssm get-document \
    --name "package-name"
```

Para obter informações sobre outras opções que podem ser usadas com o comando create-document, consulte create-document na seção AWS Systems Manager da Referência de comando da AWS CLI. Para obter informações sobre outras opções que podem ser usadas com o comando get-document, consulte get-document.

Editar permissões do pacote do Distributor no console

Após adicionar um pacote ao Distributor, uma ferramenta do AWS Systems Manager, você poderá editar as permissões do pacote no console do Systems Manager. Você pode adicionar outras Contas da AWS às permissões de um pacote. Os pacotes podem ser compartilhados somente com outras contas na mesma Região da AWS. O compartilhamento entre regiões não é compatível. Por padrão, os pacotes são definidos como Private (Privado), o que significa que apenas aqueles que tiverem acesso à Conta da AWS do criador do pacote poderá visualizar informações e atualizar ou excluir o pacote. Se as permissões Private são aceitáveis, você pode ignorar este procedimento.

> **Note**
>
> Você pode atualizar as permissões de pacotes que são compartilhados com 20 ou menos contas.

Para editar permissões do pacote no console

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Distributor.
3. Na página Packages, escolha o pacote para o qual você deseja editar permissões.
4. Na guia Package details (Detalhes do pacote), escolha Edit permissions (Editar permissões) para alterar as permissões.
5. Em Edit permissions (Editar permissões), escolha Shared with specific accounts (Compartilhadas com contas específicas).
6. Em Shared with specific accounts (Compartilhadas com contas específicas), adicione os números da Conta da AWS, um de cada vez. Ao terminar, escolha Salvar.

Editar tags de um pacote do Distributor no console

Após adicionar um pacote ao Distributor, uma ferramenta do AWS Systems Manager, você poderá editar as tags do pacote no console do Systems Manager. Essas etiquetas são aplicadas ao pacote e não estão conectadas a etiquetas em nós gerenciados nos quais você quer implantar o pacote. As tags são valores e chaves que diferenciam maiúsculas de minúsculas que podem ajudar a agrupar e filtrar seus pacotes por critérios que são relevantes para a sua organização. Se você não quiser adicionar tags, já estará pronto para instalar o pacote ou adicionar uma nova versão.

Para editar tags de um pacote no console

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Distributor.
3. Na página Packages, escolha o pacote para o qual você deseja editar tags.
4. Na guia Package details (Detalhes do pacote), em Tags, escolha Edit (Editar).
5. Em Add tags (Adicionar tags), insira uma chave de tag ou um par de chave e valor de tag e escolha Add (Adicionar). Repita se você deseja adicionar mais tags. Para excluir tags, escolha X na tag na parte inferior da janela.
6. Ao concluir a adição de tags ao seu pacote, selecione Save (Salvar).

Adicionar uma versão a um pacote do Distributor

Para adicionar uma versão do pacote, crie um pacote e use o Distributor para adicionar uma versão do pacote, acrescentando uma entrada ao documento AWS Systems Manager (SSM) que já existe para versões mais antigas. O Distributor é uma ferramenta do AWS Systems Manager. Para economizar tempo, atualize o manifesto para uma versão mais antiga do pacote, altere o valor da entrada `version` no manifesto (por exemplo, de `Test_1.0` para `Test_2.0`) e salve-o como manifesto para a nova versão. O fluxo de trabalho simples Add version (Adicionar versão) no console do Distributor faz a atualização do arquivo de manifesto para você.

Uma nova versão do pacote pode:

- Substituir pelo menos um dos arquivos de instalação anexados à versão atual.
- Adicionar novos arquivos de instalação para oferecer suporte a mais plataformas.
- Excluir arquivos para interromper o suporte para plataformas específicas.

Uma versão mais recente pode usar o mesmo bucket do Amazon Simple Storage Service (Amazon S3), mas deve ter um URL com outro nome de arquivo mostrado no final. É possível usar o console do Systems Manager ou a AWS Command Line Interface (AWS CLI) para adicionar a nova versão. Carregar um arquivo de instalação com o nome exato de um arquivo de instalação existente no bucket do Amazon S3 substitui o arquivo existente. Nenhum arquivo de instalação será copiado da versão mais antiga para a nova. Você deve fazer upload dos arquivos de instalação da versão mais antiga para que eles façam parte de uma nova versão. Depois que o Distributor terminar de criar a nova versão do pacote, você poderá excluir ou redefinir o bucket do Amazon S3, porque o

Distributor copia seu software em um bucket interno do Systems Manager como parte do processo de versionamento.

> **Note**
>
> Cada pacote é mantido em um máximo de 25 versões. Você pode excluir versões que não são mais necessárias.

Tópicos

- Adicionar uma versão de pacote usando o console
- Adicionar uma versão de pacote usando a AWS CLI

Adicionar uma versão de pacote usando o console

Antes de realizar estas etapas, siga as instruções em Criar um pacote no Distributor a fim de criar um pacote para a versão. Então, use o console do Systems Manager para adicionar uma nova versão de pacote ao Distributor.

Adicionar uma versão de pacote usando o fluxo de trabalho Simples

Para adicionar uma versão do pacote usando o fluxo de trabalho Simple (Simples), prepare os arquivos de instalação atualizados ou adicione arquivos instaláveis com suporte a mais plataformas e arquiteturas. Depois, use o Distributor para fazer upload de arquivos de instalação novos e atualizados e adicionar uma versão do pacote. O fluxo de trabalho simplificado Add version (Adicionar versão) no console do Distributor faz a atualização do arquivo de manifesto e o documento SSM associado para você.

Para adicionar uma versão de pacote usando o fluxo de trabalho Simples

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Distributor.
3. Na página inicial do Distributor, selecione o pacote ao qual você deseja adicionar outra versão.
4. Na página Add version (Adicionar versão), escolha Simple (Simples).
5. Em Version name (Nome da versão), digite um nome para a versão. O nome da versão nova precisa ser diferente do antigo. Os nomes de versões podem ter no máximo 512 caracteres e não devem conter caracteres especiais.

6. Em S3 bucket name (Nome do bucket do S3), escolha um bucket do S3 existente na lista. Ele pode ser o mesmo bucket que você usou para armazenar arquivos de instalação de versões mais antigas, mas os nomes dos arquivos de instalação precisam ser diferentes para evitar substituições de arquivos existentes no bucket.
7. Em S3 key prefix (Prefixo de chave do S3), insira a subpasta do bucket em que os ativos instaláveis estão armazenados.
8. Em Upload software (Fazer upload de software), procure os arquivos de software instaláveis que você quiser anexar à nova versão. Os arquivos de instalação de versões existentes não são automaticamente copiados para uma nova versão. Você precisa fazer upload dos arquivos de instalação de versões mais antigas do pacote se quiser que qualquer um dos mesmos arquivos de instalação façam parte da nova versão. Você pode fazer upload de mais de um arquivo de software com uma única ação.
9. Em Target platform (Plataforma de destino), verifique se a plataforma do sistema operacional de destino exibida para cada arquivo de instalação está correta. Se o sistema operacional correto não for o exibido, escolha o correto na lista suspensa.

   No fluxo de trabalho de versionamento Simple (Simples), como você carrega cada arquivo de instalação somente uma vez, etapas adicionais são necessárias para direcionar um único arquivo a mais de um sistema operacional. Por exemplo, se você carregar um arquivo de software instalável chamado `Logtool_v1.1.1.rpm`, você precisará alterar alguns valores padrão no fluxo de trabalho Simple (Simples) para instruir o Distributor a direcionar o mesmo software aos dois sistemas operacionais, Amazon Linux e Ubuntu. Você pode seguir um dos seguintes procedimentos para contornar essa limitação.

   - Use o fluxo de trabalho de versionamento Advanced (Avançado), compacte cada arquivo de instalação em um arquivo .zip antes de começar e crie o manifesto manualmente de modo que aquele arquivo de instalação possa ser direcionado a mais de uma plataforma ou versão de sistema operacional. Para ter mais informações, consulte Adicionar uma versão de pacote usando o fluxo de trabalho Avançado.
   - Edite manualmente o arquivo manifesto no fluxo de trabalho Simple (Simples) para que seu arquivo .zip seja direcionado a mais de uma plataforma ou versão de sistema operacional. Para obter mais informações sobre como fazer isso, consulte o final da etapa 4 em Etapa 2: Criar o manifesto do pacote JSON.
10. Em Platform version (Versão da plataforma), verifique se a versão exibida da plataforma do sistema operacional é **_any**, uma versão importante seguida pelo curinga (7.*) ou a versão específica do sistema operacional ao qual você deseja que o software se aplique. Para obter

mais informações sobre como especificar uma versão de plataforma, consulte a etapa 4 em Etapa 2: Criar o manifesto do pacote JSON.

11. Em Architecture (Arquitetura) escolha, na lista suspensa, a arquitetura do processador correta para cada arquivo de instalação. Para obter mais informações sobre as arquiteturas compatíveis, consulte Plataformas de pacotes e arquiteturas compatíveis.
12. (Opcional) Expanda Scripts e revise os scripts de instalação e desinstalação que o Distributor gerou para o software instalável.
13. Para adicionar mais arquivos de software instaláveis à nova versão, escolha Add software (Adicionar software). Senão, siga para a próxima etapa.
14. (Opcional) Expanda Manifest (Manifesto) e revise o manifesto do pacote JSON que o Distributor gerou para o software instalável. Se você tiver alterado qualquer informação sobre seu software instalável desde que começou esse procedimento, como a versão da plataforma ou plataforma de destino, escolha Generate manifest (Gerar manifesto) para exibir o manifesto do pacote atualizado.

    Você pode editar o manifesto manualmente se quiser direcionar o software instalável para mais de um sistema operacional, conforme descrito na etapa 9. Para obter mais informações sobre a edição de manifesto, consulte Etapa 2: Criar o manifesto do pacote JSON.
15. Quando terminar de adicionar o software e revisar os dados da plataforma de destino, da versão e da arquitetura, escolha Add version (Adicionar versão).
16. Aguarde até que o Distributor termine de carregar o software e criar sua nova versão do pacote. O Distributor mostra o status do upload de cada arquivo de instalação. Dependendo do número e do tamanho dos pacotes que você está adicionando, isso pode levar alguns minutos. O Distributor automaticamente redirecionará você para a página Package details (Detalhes do pacote) do pacote, mas você pode optar por abri-la após o software ser carregado. A página Package details (Detalhes do pacote) não mostra todas as informações sobre ele quando o Distributor finalizar o processo de criação do novo pacote. Para interromper o upload e criação da versão do pacote, escolha Stop upload (Parar upload).
17. Se o Distributor não conseguir fazer upload de qualquer um dos arquivos de instalação do software, ele exibirá a mensagem Upload failed (Falha no upload). Para tentar fazer upload novamente, escolha Retry upload (Tentar novamente). Para obter mais informações sobre como solucionar as falhas de criação da versão do pacote, consulte Solução de problemas do AWS Systems ManagerDistributor.
18. Quando o Distributor terminar de criar a nova versão do pacote, na página de Details (Detalhes) do pacote, na guia Versions (Versões), visualize a nova versão na lista versões disponíveis

do pacote. Defina uma versão padrão do pacote escolhendo uma versão e, em seguida, escolhendo Set default version (Definir versão padrão).

Se você não definir uma versão padrão, a versão mais recente do pacote será a versão padrão.

Adicionar uma versão de pacote usando o fluxo de trabalho Avançado

Para adicionar uma versão do pacote, crie um pacote e use o Distributor para adicionar uma versão do pacote, adicionando uma entrada ao documento que existe para versões mais antigas. Para economizar tempo, atualize o manifesto para uma versão mais antiga do pacote, altere o valor da entrada `version` no manifesto (por exemplo, de `Test_1.0` para `Test_2.0`) e salve-o como manifesto para a nova versão. Você precisa ter um manifesto atualizado para adicionar uma nova versão do pacote usando o fluxo de trabalho Advanced (Avançado).

Para adicionar uma versão de pacote usando o fluxo de trabalho Avançado

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Distributor.
3. Na página inicial do Distributor, selecione o pacote ao qual você quer adicionar outra versão e escolha Add version (Adicionar versão).
4. Em Version name (Nome da versão), insira o valor exato que está na entrada `version` do arquivo manifesto.
5. Em S3 bucket name (Nome do bucket do S3), escolha um bucket do S3 existente na lista. Ele pode ser o mesmo bucket que você usou para armazenar arquivos de instalação de versões mais antigas, mas os nomes dos arquivos de instalação precisam ser diferentes para evitar substituições de arquivos existentes no bucket.
6. Em S3 key prefix (Prefixo de chave do S3), insira a subpasta do bucket em que os ativos instaláveis estão armazenados.
7. Em Manifest (Manifesto), escolha Extract from package (Extrair do pacote) para usar um manifesto enviado ao bucket do S3 com os arquivos .zip.

   (Opcional) Se você não carregou o manifesto JSON revisado no bucket do Amazon S3 em que armazenou os arquivos .zip, escolha New manifest (Novo manifesto). Você pode criar ou colar o manifesto inteiro no campo de edição JSON. Para obter mais informações sobre como criar o manifesto JSON, consulte Etapa 2: Criar o manifesto do pacote JSON.
8. Quando terminar o manifesto, escolha Add package version (Adicionar versão do pacote).

9. Na página de Detalhes do pacote, na guia Versions, visualize a nova versão na lista de versões de pacote disponíveis. Defina uma versão padrão do pacote escolhendo uma versão e, em seguida, escolhendo Set default version (Definir versão padrão).

   Se você não definir uma versão padrão, a versão mais recente do pacote será a versão padrão.

Adicionar uma versão de pacote usando a AWS CLI

Você pode usar a AWS CLI para adicionar uma nova versão de pacote ao Distributor. Antes de executar esses comandos, você deve criar uma nova versão do pacote e fazer upload dela no S3, como descrito no início deste tópico.

Para adicionar uma versão de pacote usando a AWS CLI

1. Execute o comando a seguir para editar o documento do AWS Systems Manager com uma entrada para uma nova versão do pacote. Substitua *document-name* pelo nome do seu documento. Substitua *amzn-s3-demo-bucket* pela URL do manifesto JSON que você copiou em Etapa 3: Fazer upload do pacote e do manifesto em um bucket do Amazon S3. *S3-bucket-URL-of-package* é a URL do bucket do Amazon S3 no qual o pacote inteiro está armazenado. Substitua *version-name-from-updated-manifest* pelo valor de `version` no manifesto. Defina o parâmetro `--document-version` para `$LATEST` para fazer o documento associado a essa versão do pacote a versão mais recente do documento.

   ```
   aws ssm update-document \
       --name "document-name" \
       --content "S3-bucket-URL-to-manifest-file" \
       --attachments Key="SourceUrl",Values="amzn-s3-demo-bucket" \
       --version-name version-name-from-updated-manifest \
       --document-version $LATEST
   ```

   Veja um exemplo a seguir.

   ```
   aws ssm update-document \
       --name ExamplePackage \
       --content "https://s3.amazonaws.com/amzn-s3-demo-bucket/ExamplePackage/
   manifest.json" \
       --attachments Key="SourceUrl",Values="https://s3.amazonaws.com/amzn-s3-demo-
   bucket/ExamplePackage" \
       --version-name 1.1.1 \
   ```

```
    --document-version $LATEST
```

2. Execute o seguinte comando para verificar se o pacote foi atualizado e mostrar o manifesto do pacote. Substitua *package-name* pelo nome de seu pacote e, opcionalmente, *document-version* pelo número da versão do documento (não o mesmo que a versão do pacote) que você atualizou. Se essa versão do pacote estiver associada à versão mais recente do documento, você pode especificar `$LATEST` para o valor do parâmetro opcional `--document-version`.

```
aws ssm get-document \
    --name "package-name" \
    --document-version "document-version"
```

Para obter informações sobre outras opções que podem ser usadas com o comando update-document, consulte update-document na seção AWS Systems Manager da Referência de comando da AWS CLI.

Instalar ou atualizar pacotes do Distributor

Você pode implantar pacotes em seus nós gerenciados do AWS Systems Manager usando o Distributor, uma ferramenta do AWS Systems Manager. Para implantar os pacotes, use o AWS Management Console ou AWS Command Line Interface (AWS CLI). Você pode implantar uma versão de um pacote por comando. Você pode instalar novos pacotes ou atualizar instalações existentes no local. Você pode optar por implantar uma versão específica ou escolha sempre implantar a versão mais recente de um pacote de implantação. Recomendamos usar o State Manager, uma ferramenta do AWS Systems Manager, para instalar pacotes. O uso do State Manager ajuda a garantir que os nós gerenciados estejam sempre executando a versão mais atualizada do pacote.

> **Important**
>
> Os pacotes instalados com o Distributor devem ser desinstalados somente com o Distributor. Caso contrário, o Systems Manager ainda poderá registrar a aplicação como `INSTALLED` e gerar outros resultados indesejados.

Preferência	Ação do AWS Systems Manager	Mais informações
Instalar ou atualizar um pacote imediatamente.	Run Command	• Instalar ou atualizar um pacote uma única vez usando o console • Instalar um pacote uma única vez usando a AWS CLI • Atualizar um pacote uma única vez usando a AWS CLI
Instalar ou atualizar um pacote em uma programação, para que a instalação sempre inclua a versão padrão.	State Manager	• Programar uma instalação ou atualização de pacote usando o console • Programar uma instalação de pacote usando a AWS CLI • Programar uma atualização de pacote usando a AWS CLI
Instalar automaticamente um pacote em novos nós gerenciados que tenham uma etiqueta ou um conjunto específico de etiquetas. Por exemplo, a instalação do agente Amazon CloudWatch em novas instâncias.	State Manager	Uma maneira de fazer isso é aplicar etiquetas aos novos nós gerenciados e, em seguida, especificar as etiquetas como destinos em sua associação State Manager. O State Manager instala automaticamente o pacote em uma associação em nós gerenciados que têm etiquetas correspondentes. Consulte Sobre destinos e controles de taxa

Preferência	Ação do AWS Systems Manager	Mais informações
		em associações do State Manager.

Tópicos

- Instalar ou atualizar um pacote uma única vez usando o console
- Programar uma instalação ou atualização de pacote usando o console
- Instalar um pacote uma única vez usando a AWS CLI
- Atualizar um pacote uma única vez usando a AWS CLI
- Programar uma instalação de pacote usando a AWS CLI
- Programar uma atualização de pacote usando a AWS CLI

Instalar ou atualizar um pacote uma única vez usando o console

É possível usar o console do AWS Systems Manager para instalar ou atualizar um pacote uma vez. Quando você configura uma instalação única, o Distributor usa AWS Systems Manager Run Command, uma ferramenta do AWS Systems Manager, para executar a instalação.

Para instalar ou atualizar um pacote uma única vez usando o console

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Distributor.
3. Na página inicial do Distributor, selecione o pacote que deseja instalar.
4. Escolha Install one time (Instalar uma vez).

   Esse comando abre o Run Command com o documento de comando `AWS-ConfigureAWSPackage` e seu pacote Distributor já selecionado.
5. Em Document version (Versão do documento), selecione a versão documento `AWS-ConfigureAWSPackage` que deseja executar.
6. Em Action (Ação), selecione Install (Instalar).
7. Em Installation type (Tipo de instalação), escolha uma das seguintes opções:

- Uninstall and reinstall (Desinstalar e reinstalar): o pacote é totalmente desinstalado e, depois, reinstalado. A aplicação estará indisponível até que a reinstalação seja concluída.
- In-place update (Atualização no local): somente arquivos novos ou alterados são adicionados à instalação existente de acordo com as instruções fornecidas em um script `update`. O aplicativo permanece disponível durante todo o processo de atualização. Esta opção não tem suporte para os pacotes publicados da AWS, exceto os pacotes `AWSEC2Launch-Agent`.

8. Em Name (Nome), verifique se o nome do pacote selecionado foi inserido.
9. (Opcional) Em Version (Versão), insira o valor do nome da versão do pacote. Se você deixar esse campo em branco, o Run Command instala a versão padrão que você selecionou em Distributor.
10. Na seção Targets (Destinos), escolha os nós gerenciados nos quais você quer executar essa operação, especificando as etiquetas, selecionando as instâncias ou dispositivos manualmente ou especificando um grupo de recursos.

> **Note**
>
> Se você não encontrar um nó gerenciado na lista, consulte Solução de problemas de disponibilidade do nó gerenciado.

11. Para Other parameters (Outros parâmetros):
    - Em Comment (Comentário), digite as informações sobre esse comando.
    - Em Timeout (seconds) (Tempo limite [segundos]), especifique o número de segundos para o sistema aguardar até a falha de execução do comando total.
12. Para Rate control (Controle de taxa):
    - Em Concurrency (Concorrência), especifique um número ou uma porcentagem de destinos nos quais executar o comando ao mesmo tempo.

> **Note**
>
> Se você selecionou destinos especificando etiquetas ou grupos de recursos, e não tiver certeza de quantos nós gerenciados são selecionados como destino, restrinja o número de destinos que poderão executar o documento ao mesmo tempo, especificando uma porcentagem.

- Em Error threshold (Limite de erro), especifique quando parar de executar o comando em outros destinos depois de falhar em alguns ou em uma porcentagem de nós gerenciados. Por exemplo, se você especificar três erros, o Systems Manager deixará de enviar o comando quando o 4° erro for recebido. Os nós gerenciados que continuam processando o comando também podem enviar erros.

13. (Opcional) Em Output options (Opções de saída), para salvar a saída do comando em um arquivo, selecione a caixa Write command output to an S3 bucket (Gravar saída do comando em um bucket do S3). Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

    > **Note**
    >
    > As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil de instância (para instâncias do EC2) ou perfil de serviço do IAM (máquinas ativadas para ambientes híbridos) atribuído à instância, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço do IAM associado ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

14. Na seção SNS notifications (Notificações do SNS), se quiser enviar notificações sobre o status da execução do comando, marque a caixa de seleção Enable SNS notifications (Habilitar notificações do SNS).

    Para obter mais informações sobre a configuração de notificações do Amazon SNS para o Run Command, consulte Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS.

15. Quando estiver pronto para instalar o pacote, escolha Run (Executar).
16. A área Command status (Status do comando) informa o andamento da execução. Se o comando ainda estiver em andamento, escolha o ícone de atualização no canto superior esquerdo do console até a coluna Overall status (Status geral) ou Detailed status (Status detalhado) mostrar Success (Êxito) ou Failed (Falha).
17. Na área Targets and outputs (Destinos e saídas), escolha o botão ao lado de um nome de nó gerenciado e escolha View output (Visualizar saída).

    A página de saída do comando mostra os resultados da execução do comando.

18. (Opcional) Se você optar por gravar a saída do comando em um bucket do Amazon S3, escolha Amazon S3 para visualizar os dados do log de resultados.

Programar uma instalação ou atualização de pacote usando o console

Você pode usar o console do AWS Systems Manager para agendar a instalação ou atualização de um pacote. Quando você agenda a instalação ou atualização do pacote, o Distributor usa AWS Systems Manager State Manager para instalar ou atualizar.

Para programar uma instalação de pacote usando o console

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Distributor.
3. Na página inicial do Distributor, selecione o pacote que deseja instalar ou atualizar.
4. Em Package (Pacote), escolha Install on a schedule (Instalação programada).

   Esse comando abre o State Manager para uma nova associação que será criada para você.
5. Em Name (Nome), insira um nome (por exemplo, **`Deploy-test-agent-package`**). Isso é opcional, mas recomendado. Espaços não são permitidos no nome.
6. Na lista Document (Documento), o nome do documento `AWS-ConfigureAWSPackage` já está selecionado.
7. Em Action (Ação), verifique se Install (Instalar) está selecionada.
8. Em Installation type (Tipo de instalação), escolha uma das seguintes opções:

   - Uninstall and reinstall (Desinstalar e reinstalar): o pacote é totalmente desinstalado e, depois, reinstalado. A aplicação estará indisponível até que a reinstalação seja concluída.
   - In-place update (Atualização no local): somente arquivos novos ou alterados são adicionados à instalação existente de acordo com as instruções fornecidas em um script `update`. O aplicativo permanece disponível durante todo o processo de atualização.
9. Em Name (Nome), verifique se o nome do seu pacote foi inserido.
10. Em Version (Versão), se você quiser instalar uma versão de pacote diferente da versão publicada mais recente, insira o identificador de versão.
11. Em Targets (Destinos), escolha Selecting all managed instances in this account (Selecionar todas as instâncias gerenciadas nesta conta), Specifying tags (Especificar tags) ou Manually Selecting Instance (Selecionar instância manualmente). Se você tiver recursos de destino usando tags, insira uma chave de tag e um valor de tag nos campos fornecidos.

> **Note**
>
> Você pode escolher dispositivos principais do AWS IoT Greengrass gerenciado, escolhendo as opções Selecting all managed instances in this account (Selecionar todas as instâncias gerenciadas nesta conta) ou Manually Selecting Instance (Selecionar a instância manualmente).

12. Em Specify schedule (Especificar programação), escolha On Schedule (Na programação) para executar a associação em uma programação normal, ou No Schedule (Sem programação) para executar a associação uma vez. Para obter mais informações sobre essas opções, consulte Para obter informações, consulte Trabalhar com associações no Systems Manager.. Use os controles para criar uma programação `cron` ou rate para a associação.
13. Escolha Create Association (Criar associação).
14. Na página Association (Associação) escolha o botão ao lado da associação criada e escolha Apply association now (Aplicar associação agora).

    O State Manager cria e executa imediatamente a associação nos destinos especificados. Para obter mais informações sobre os resultados da execução de associações, consulte Para obter informações, consulte Trabalhar com associações no Systems Manager. neste guia.

Para obter mais informações sobre como trabalhar com as opções em Advanced options (Opções avançadas), Rate control (Controle de taxa) e Output options (Opções de saída), consulte Para obter informações, consulte Trabalhar com associações no Systems Manager..

Instalar um pacote uma única vez usando a AWS CLI

É possível executar send-command na AWS CLI para instalar um pacote de Distributor uma vez. Se o pacote já estiver instalado, o aplicativo ficará offline enquanto o pacote for desinstalado e a nova versão for instalada em seu lugar.

Para instalar um pacote uma única vez usando a AWS CLI

- Execute o comando a seguir na AWS CLI.

```
aws ssm send-command \
    --document-name "AWS-ConfigureAWSPackage" \
    --instance-ids "instance-IDs" \
```

```
    --parameters '{"action":["Install"],"installationType":["Uninstall and
 reinstall"],"name":["package-name (in same account) or package-ARN (shared from
 different account)"]}'
```

> **Note**
>
> O comportamento padrão para `installationType` é `Uninstall and reinstall`. Você pode omitir `"installationType":["Uninstall and reinstall"]` desse comando quando estiver instalando um pacote completo.

Veja um exemplo a seguir.

```
aws ssm send-command \
    --document-name "AWS-ConfigureAWSPackage" \
    --instance-ids "i-00000000000000" \
    --parameters '{"action":["Install"],"installationType":["Uninstall and
 reinstall"],"name":["ExamplePackage"]}'
```

Para obter informações sobre outras opções que podem ser usadas com o comando send-command, consulte send-command na seção AWS Systems Manager da Referência de comando da AWS CLI.

Atualizar um pacote uma única vez usando a AWS CLI

Você pode executar send-command na AWS CLI para atualizar um pacote do Distributor sem deixar o aplicativo associado offline. Somente arquivos novos ou atualizados no pacote são substituídos.

Para atualizar um pacote uma única vez usando a AWS CLI

- Execute o comando a seguir na AWS CLI.

```
aws ssm send-command \
    --document-name "AWS-ConfigureAWSPackage" \
    --instance-ids "instance-IDs" \
    --parameters '{"action":["Install"],"installationType":["In-place
 update"],"name":["package-name (in same account) or package-ARN (shared from
 different account)"]}'
```

> **Note**
>
> Ao adicionar arquivos novos ou alterados, é necessário incluir `"installationType": ["In-place update"]` no comando.

Veja um exemplo a seguir.

```
aws ssm send-command \
    --document-name "AWS-ConfigureAWSPackage" \
    --instance-ids "i-02573cafcfEXAMPLE" \
    --parameters '{"action":["Install"],"installationType":["In-place
 update"],"name":["ExamplePackage"]}'
```

Para obter informações sobre outras opções que podem ser usadas com o comando send-command, consulte send-command na seção AWS Systems Manager da Referência de comando da AWS CLI.

Programar uma instalação de pacote usando a AWS CLI

É possível executar create-association na AWS CLI para instalar um pacote de Distributor em uma programação. O valor de `--name`, o nome do documento, é sempre `AWS-ConfigureAWSPackage`. O comando a seguir usa a chave `InstanceIds` para especificar os nós gerenciados de destino. Se o pacote já estiver instalado, o aplicativo ficará offline enquanto o pacote for desinstalado e a nova versão for instalada em seu lugar.

```
aws ssm create-association \
    --name "AWS-ConfigureAWSPackage" \
    --parameters '{"action":["Install"],"installationType":["Uninstall and
 reinstall"],"name":["package-name (in same account) or package-ARN (shared from
 different account)"]}' \
    --targets [{\"Key\":\"InstanceIds\",\"Values\":[\"instance-ID1\",\"instance-
ID2\"]}]
```

> **Note**
>
> O comportamento padrão para `installationType` é `Uninstall and reinstall`. Você pode omitir `"installationType":["Uninstall and reinstall"]` desse comando quando estiver instalando um pacote completo.

Veja um exemplo a seguir.

```
aws ssm create-association \
    --name "AWS-ConfigureAWSPackage" \
    --parameters '{"action":["Install"],"installationType":["Uninstall and
 reinstall"],"name":["Test-ConfigureAWSPackage"]}' \
    --targets [{\"Key\":\"InstanceIds\",\"Values\":[\"i-02573cafcfEXAMPLE\",
\"i-0471e04240EXAMPLE\"]}]
```

Para obter informações sobre outras opções que podem ser usadas com o comando create-association, consulte create-association na seção AWS Systems Manager da Referência de comando da AWS CLI.

Programar uma atualização de pacote usando a AWS CLI

Você pode executar create-association na AWS CLI para atualizar um pacote do Distributor em uma programação sem deixar o aplicativo associado offline. Somente arquivos novos ou atualizados no pacote são substituídos. O valor de `--name`, o nome do documento, é sempre `AWS-ConfigureAWSPackage`. O comando a seguir usa a chave `InstanceIds` para especificar as instâncias de destino.

```
aws ssm create-association \
    --name "AWS-ConfigureAWSPackage" \
    --parameters '{"action":["Install"],"installationType":["In-place update"],"name":
["package-name (in same account) or package-ARN (shared from different account)"]}' \
    --targets [{\"Key\":\"InstanceIds\",\"Values\":[\"instance-ID1\",\"instance-
ID2\"]}]
```

> **Note**
>
> Ao adicionar arquivos novos ou alterados, é necessário incluir `"installationType": ["In-place update"]` no comando.

Veja um exemplo a seguir.

```
aws ssm create-association \
    --name "AWS-ConfigureAWSPackage" \
    --parameters '{"action":["Install"],"installationType":["In-place update"],"name":
["Test-ConfigureAWSPackage"]}' \
    --targets [{\"Key\":\"InstanceIds\",\"Values\":[\"i-02573cafcfEXAMPLE\",
\"i-0471e04240EXAMPLE\"]}]
```

Para obter informações sobre outras opções que podem ser usadas com o comando create-association, consulte create-association na seção AWS Systems Manager da Referência de comando da AWS CLI.

Desinstalar um pacote do Distributor

Você pode usar o AWS Management Console ou o AWS Command Line Interface (AWS CLI) para desinstalar os pacotes do Distributor dos nós gerenciados pelo AWS Systems Manager, usando o Run Command. O Distributor e o Run Command são ferramentas do AWS Systems Manager. Nesta versão, você pode desinstalar uma versão de um pacote por comando. Você pode desinstalar uma versão específica ou a versão padrão.

> **Important**
>
> Os pacotes instalados com o Distributor devem ser desinstalados somente com o Distributor. Caso contrário, o Systems Manager ainda poderá registrar a aplicação como `INSTALLED` e gerar outros resultados indesejados.

Tópicos

- Desinstalar um pacote usando o console
- Desinstalar um pacote usando a AWS CLI

Desinstalar um pacote usando o console

Você pode usar o Run Command no console do Systems Manager para desinstalar um pacote uma vez. O Distributor usa AWS Systems Manager Run Command para desinstalar pacotes.

Para desinstalar um pacote usando o console

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Run Command.
3. Na página inicial do Run Command, selecione Run command (Executar comando).
4. Escolha o documento de comando `AWS-ConfigureAWSPackage`.
5. Em Action (Ação), escolha Desinstalar
6. Para Name, insira o nome do pacote que você deseja desinstalar.
7. Para Targets (Destinos), escolha como você deseja segmentar seus nós gerenciados. Você pode especificar uma chave e valores de etiqueta que são compartilhados pelos destinos. Você também pode especificar destinos escolhendo atributos, como ID, plataforma e versão do SSM Agent.
8. Você pode usar as opções avançadas para adicionar comentários sobre a operação, alterar os valores Concurrency (Simultaneidade) e Error threshold (Limite de erros) em Rate control (Controle de taxas), especificar as opções de saída ou configurar notificações do Amazon Simple Notification Service (Amazon SNS). Para obter mais informações, consulte Executar comandos a partir do console neste guia.
9. Quando você estiver pronto para desinstalar o pacote, escolha Run (Executar), e depois View results (Visualizar resultados).
10. Na lista de comandos, escolha o comando `AWS-ConfigureAWSPackage` que você executou. Se o comando ainda estiver em andamento, escolha o ícone de atualização no canto superior direito do console.
11. Quando a coluna Status indicar Success (Êxito) ou Failed (Falha), escolha a guia Output (Saída).
12. Escolha View output. A página de saída do comando mostra os resultados da execução do comando.

Desinstalar um pacote usando a AWS CLI

Você pode usar o AWS CLI para desinstalar um pacote do Distributor dos nós gerenciados usando o Run Command.

Para desinstalar um pacote usando a AWS CLI

- Execute o comando a seguir na AWS CLI.

```
aws ssm send-command \
    --document-name "AWS-ConfigureAWSPackage" \
    --instance-ids "instance-IDs" \
    --parameters '{"action":["Uninstall"],"name":["package-name (in same account)
 or package-ARN (shared from different account)"]}'
```

Veja um exemplo a seguir.

```
aws ssm send-command \
    --document-name "AWS-ConfigureAWSPackage" \
    --instance-ids "i-02573cafcfEXAMPLE" \
    --parameters '{"action":["Uninstall"],"name":["Test-ConfigureAWSPackage"]}'
```

Para obter informações sobre outras opções que podem ser usadas com o comando send-command, consulte send-command na seção AWS Systems Manager da Referência de comando da AWS CLI.

Excluir um pacote do Distributor

Esta seção descreve como excluir um pacote. Você não pode excluir uma versão de um pacote, somente o pacote inteiro.

Excluir um pacote usando o console

Você pode usar o console do AWS Systems Manager para excluir um pacote ou a versão de um pacote do Distributor, uma ferramenta do AWS Systems Manager. Excluir um pacote excluirá todas as versões de um pacote do Distributor.

Para excluir um pacote usando o console

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Distributor.
3. Na página inicial do Distributor, selecione o pacote que deseja excluir.
4. Na página de detalhes do pacote, escolha Excluir pacote.
5. Quando for solicitado a confirmar a exclusão, escolha Delete package (Excluir pacote).

Excluir uma versão de pacote usando o console

É possível usar o console do Systems Manager para excluir uma versão pacote do Distributor.

Para excluir uma versão de pacote usando o console

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Distributor.
3. Na página inicial do Distributor, selecione o pacote a ser excluído.
4. Na página de versões do pacote, escolha a versão a ser excluída e escolha Delete version (Excluir versão).
5. Quando for solicitado a confirmar a exclusão, escolha Delete package version (Excluir versão do pacote).

Excluir um pacote usando a linha de comando

Você pode usar a ferramenta de linha de comando de sua preferência para excluir um pacote do Distributor.

Linux & macOS

Para excluir um pacote usando a AWS CLI

1. Execute o comando a seguir para listar documentos para pacotes específicos. Nos resultados deste comando, procure o pacote que você deseja excluir.

```
aws ssm list-documents \
    --filters Key=Name,Values=package-name
```

2. Execute o seguinte comando para excluir um pacote. Substitua *package-name* pelo nome do pacote.

```
aws ssm delete-document \
    --name "package-name"
```

3. Execute o comando list-documents novamente para verificar se o pacote foi excluído. O pacote que você excluiu não deve ser incluído na lista.

```
aws ssm list-documents \
```

```
    --filters Key=Name,Values=package-name
```

Windows

Para excluir um pacote usando a AWS CLI

1. Execute o comando a seguir para listar documentos para pacotes específicos. Nos resultados deste comando, procure o pacote que você deseja excluir.

```
aws ssm list-documents ^
    --filters Key=Name,Values=package-name
```

2. Execute o seguinte comando para excluir um pacote. Substitua *package-name* pelo nome do pacote.

```
aws ssm delete-document ^
    --name "package-name"
```

3. Execute o comando list-documents novamente para verificar se o pacote foi excluído. O pacote que você excluiu não deve ser incluído na lista.

```
aws ssm list-documents ^
    --filters Key=Name,Values=package-name
```

PowerShell

Para excluir um pacote usando as Ferramentas para PowerShell

1. Execute o comando a seguir para listar documentos para pacotes específicos. Nos resultados deste comando, procure o pacote que você deseja excluir.

```
$filter = New-Object
 Amazon.SimpleSystemsManagement.Model.DocumentKeyValuesFilter
$filter.Key = "Name"
$filter.Values = "package-name"

Get-SSMDocumentList `
    -Filters @($filter)
```

2. Execute o seguinte comando para excluir um pacote. Substitua *package-name* pelo nome do pacote.

```
Remove-SSMDocument `
    -Name "package-name"
```

3. Execute o comando Get-SSMDocumentList novamente para verificar se o pacote foi excluído. O pacote que você excluiu não deve ser incluído na lista.

```
$filter = New-Object
 Amazon.SimpleSystemsManagement.Model.DocumentKeyValuesFilter
$filter.Key = "Name"
$filter.Values = "package-name"

Get-SSMDocumentList `
    -Filters @($filter)
```

Excluir uma versão de pacote usando a linha de comando

Você pode usar a ferramenta de linha de comando de sua preferência para excluir uma versão de pacote do Distributor.

Linux & macOS

Para excluir uma versão de pacote usando a AWS CLI

1. Execute o seguinte comando para listar as versões do seu pacote. Nos resultados desse comando, procure a versão do pacote que você deseja excluir.

```
aws ssm list-document-versions \
    --name "package-name"
```

2. Execute o seguinte comando para excluir uma versão do pacote. Substitua *package-name* pelo nome do pacote e *version* pelo número da versão.

```
aws ssm delete-document \
    --name "package-name" \
    --document-version version
```

3. Execute o comando list-document-versions para verificar se a versão do pacote foi excluída. A versão do pacote que você excluiu não deve mais ser encontrada.

```
aws ssm list-document-versions \
    --name "package-name"
```

Windows

Para excluir uma versão de pacote usando a AWS CLI

1. Execute o seguinte comando para listar as versões do seu pacote. Nos resultados desse comando, procure a versão do pacote que você deseja excluir.

```
aws ssm list-document-versions ^
    --name "package-name"
```

2. Execute o seguinte comando para excluir uma versão do pacote. Substitua *package-name* pelo nome do pacote e *version* pelo número da versão.

```
aws ssm delete-document ^
    --name "package-name" ^
    --document-version version
```

3. Execute o comando list-document-versions para verificar se a versão do pacote foi excluída. A versão do pacote que você excluiu não deve mais ser encontrada.

```
aws ssm list-document-versions ^
    --name "package-name"
```

PowerShell

Para excluir uma versão do pacote usando as Ferramentas para PowerShell

1. Execute o seguinte comando para listar as versões do seu pacote. Nos resultados desse comando, procure a versão do pacote que você deseja excluir.

```
Get-SSMDocumentVersionList `
    -Name "package-name"
```

2. Execute o seguinte comando para excluir uma versão do pacote. Substitua *package-name* pelo nome do pacote e *version* pelo número da versão.

```
Remove-SSMDocument `
    -Name "package-name" `
    -DocumentVersion version
```

3. Execute o comando Get-SSMDocumentVersionList para verificar se a versão do pacote foi excluída. A versão do pacote que você excluiu não deve mais ser encontrada.

```
Get-SSMDocumentVersionList `
    -Name "package-name"
```

Para obter informações sobre outras opções que podem ser usadas com o comando list-documents, consulte list-documents na seção AWS Systems Manager da Referência de comando da AWS CLI. Para obter informações sobre outras opções que podem ser usadas com o comando delete-document, consulte delete-document.

Auditar e registrar atividades do Distributor em log

É possível usar o AWS CloudTrail para auditar atividades relacionadas ao Distributor, uma ferramenta do AWS Systems Manager. Para obter mais informações sobre opções de auditoria e registro para o Systems Manager, consulte Registrar em log e monitorar no AWS Systems Manager.

Auditore a atividade do Distributor usando o CloudTrail

O CloudTrail captura chamadas à API feitas no console do AWS Systems Manager, na AWS Command Line Interface (AWS CLI) e no SDK do Systems Manager. As informações podem ser visualizadas no console do CloudTrail ou armazenadas em um bucket do Amazon Simple Storage Service (Amazon S3). Um bucket é usado em todos os logs do CloudTrail para sua conta.

Os logs de ações do Run Command e State Manager mostram a criação de documentos, a instalação de pacotes e a atividade de desinstalação do pacote. Run Command e State Manager são ferramentas do AWS Systems Manager. Para obter mais informações sobre como visualizar e utilizar os logs do CloudTrail de atividades do Systems Manager, consulte Registrar em log chamadas de API do AWS Systems Manager com o AWS CloudTrail.

Solução de problemas do AWS Systems ManagerDistributor

As informações a seguir podem ajudá-lo a solucionar problemas que podem ocorrer ao usar o Distributor, uma ferramenta do AWS Systems Manager.

Tópicos

- Pacote incorreto com o mesmo nome está instalado
- Erro: falha ao recuperar manifesto: não foi possível encontrar a versão mais recente do pacote
- Erro: falha ao recuperar manifesto: exceção de validação
- O pacote não é compatível (a ação de instalação está faltando no pacote)
- Erro: Falha ao baixar o manifesto: Documento com nome não existe
- O upload falhou.
- Erro: falha ao localizar a plataforma: nenhum manifesto encontrado para a plataforma: oracle, versão 8.9, arquitetura x86_64

Pacote incorreto com o mesmo nome está instalado

Problema: você instalou um pacote, mas o Distributor instalou um pacote diferente.

Causa: durante a instalação, o Systems Manager localiza pacotes publicados pela AWS como resultados, antes dos pacotes externos definidos pelo usuário. Se o nome do pacote definido pelo usuário for o mesmo que um nome de pacote publicado pela AWS, o pacote da AWS será instalado em vez do seu pacote.

Solução: para evitar esse problema, dê um nome ao seu pacote que seja diferente do nome de um pacote publicado pela AWS.

Erro: falha ao recuperar manifesto: não foi possível encontrar a versão mais recente do pacote

Problema: você recebeu um erro semelhante ao seguinte:

```
Failed to retrieve manifest: ResourceNotFoundException: Could not find the latest
 version of package
arn:aws:ssm:::package/package-name status code: 400, request id: guid
```

Causa: você está usando uma versão do SSM Agent cujo com o Distributor, que é anterior à versão 2.3.274.0.

Solução: atualize a versão do SSM Agent para a versão 2.3.274.0 ou posterior. Para ter mais informações, consulte Atualização do SSM Agent por meio de Run Command ou Walkthrough: Automatically update SSM Agent with the AWS CLI.

Erro: falha ao recuperar manifesto: exceção de validação

Problema: você recebeu um erro semelhante ao seguinte:

```
Failed to retrieve manifest: ValidationException: 1 validation error detected: Value
 'documentArn'
at 'packageName' failed to satisfy constraint: Member must satisfy regular expression
 pattern:
arn:aws:ssm:region-id:account-id:package/package-name
```

Causa: você está usando uma versão do SSM Agent cujo com o Distributor, que é anterior à versão 2.3.274.0.

Solução: atualize a versão do SSM Agent para a versão 2.3.274.0 ou posterior. Para ter mais informações, consulte Atualização do SSM Agent por meio de Run Command ou Walkthrough: Automatically update SSM Agent with the AWS CLI.

O pacote não é compatível (a ação de instalação está faltando no pacote)

Problema: você recebeu um erro semelhante ao seguinte:

```
Package is not supported (package is missing install action)
```

Causa: a estrutura do diretório do pacote está incorreta.

Solução: não compacte um diretório pai contendo o software e os scripts necessários. Em vez disso, crie um arquivo `.zip` de todos os conteúdos necessários diretamente no caminho absoluto. Para verificar se o `.zip` foi criado corretamente, descompacte o diretório da plataforma de destino e revise a estrutura do diretório. Por exemplo, o caminho absoluto do script de instalação deve ser */ExamplePackage_targetPlatform*`/install.sh`.

Erro: Falha ao baixar o manifesto: Documento com nome não existe

Problema: você recebeu um erro semelhante ao seguinte:

```
Failed to download manifest - failed to retrieve package document description:
 InvalidDocument: Document with name filename does not exist.
```

Causa 1: o Distributor não consegue encontrar o pacote pelo nome ao compartilhar um pacote do Distributor de outra conta.

Solução 1: ao compartilhar um pacote de outra conta, use o nome do recurso da Amazon (ARN) completo do pacote, não apenas o nome.

Causa 2: ao usar uma VPC, você não forneceu ao seu perfil de instância do IAM acesso ao bucket do S3 gerenciado pela AWS que contém o documento `AWS-ConfigureAWSPackage` para a Região da AWS que você está direcionando.

Solução 2: certifique-se de que seu perfil de instância do IAM forneça ao SSM Agent acesso ao bucket do S3 gerenciado pela AWS que contém o documento `AWS-ConfigureAWSPackage` para a Região da AWS que você está direcionando, conforme explicado em Comunicações do SSM Agent com os buckets do S3 gerenciados pela AWS.

O upload falhou.

Problema: você recebeu um erro semelhante a este.

```
Upload failed. At least one of your files was not successfully uploaded to your S3
 bucket.
```

Causa: o nome do pacote de software inclui um espaço. Por exemplo, `Hello World.msi` falharia no upload.

Erro: falha ao localizar a plataforma: nenhum manifesto encontrado para a plataforma: oracle, versão 8.9, arquitetura x86_64

Problema: você recebeu um erro semelhante a este.

```
Failed to find platform: no manifest found for platform: oracle, version 8.9,
 architecture x86_64
```

Causa: o erro significa que o manifesto do pacote JSON não tem uma definição para a plataforma, neste caso, o Oracle Linux.

Solução: no site do Trend Micro Deep Security Software, baixe o pacote que você deseja distribuir. Crie um pacote de software `.rpm` utilizando o Processo de fluxo de trabalho simples do Distributor. Defina os valores a seguir para o pacote e finalize o upload do pacote de software utilizando o Distributor:

```
Platform version: _any
Target platform: oracle
Architecture: x86_64
```

AWS Systems Manager Fleet Manager

O Fleet Manager, uma ferramenta do AWS Systems Manager, é uma experiência de interface de usuário unificada (UI) que ajuda você a gerenciar remotamente os nós em execução na AWS ou on-premises. com o Fleet Manager, é possível visualizar o status de integridade e a performance de toda a sua frota de servidores em um console. Você também pode coletar dados de nós individuais para executar tarefas comuns de solução de problemas e gerenciamento no console. Isso inclui a conexão com instâncias do Windows usando o Remote Desktop Protocol (RDP), visualização de conteúdo de pastas e arquivos, gerenciamento de registro do Windows, gerenciamento de usuários do sistema operacional e muito mais.

Para começar a usar o Fleet Manager, abra o Systems Manager console (Console do gerenciador de sistemas). No painel de navegação, escolha Fleet Manager.

Quem deve usar o Fleet Manager?

Qualquer cliente da AWS que quiser ter uma maneira centralizada de gerenciar sua frota de nós deverá usar o Fleet Manager.

Como o Fleet Manager beneficia minha organização?

Fleet Manager oferece estes benefícios:

- Execute uma variedade de tarefas comuns de administração de sistemas sem precisar se conectar manualmente aos nós gerenciados.
- Gerencie nós executados em várias plataformas em um único console unificado.
- Gerencie nós que executam diferentes sistemas operacionais em um único console unificado.
- Melhore a eficiência da administração de seus sistemas.

Quais são os recursos do Fleet Manager?

Os principais recursos do Fleet Manager incluem o seguinte:

- Acesse o Red Hat Knowledgebase Portal

Acesse binários, compartilhamentos de conhecimento e fóruns de discussão no Red Hat Knowledgebase Portal através das instâncias do Red Hat Enterprise Linux (RHEL).

- Status do nó gerenciado

  Visualize quais instâncias estão `running` e quais estão `stopped`. Para obter mais informações sobre instâncias interrompidas, consulte Interromper e iniciar a instância no Guia do usuário do Amazon EC2. Para os dispositivos principais do AWS IoT Greengrass, você pode ver quais estão `online`, `offline` ou quais mostram um status de `Connection lost`.

  > **Note**
  >
  > Se você interrompeu a instância gerenciada antes de 12 de julho de 2021, ela não exibirá o marcador `stopped`. Para mostrar o marcador, inicie e interrompa a instância.

- Visualizar informações da instância

  Visualizar informações sobre a pasta e os dados de arquivo armazenados nos volumes anexados às instâncias gerenciadas, dados de performance sobre as instâncias em tempo real e dados de log armazenados em suas instâncias.

- Exiba informações do dispositivo de borda

  Visualize o nome da coisa do AWS IoT Greengrass para o dispositivo, o status e a versão do ping do SSM Agent e muito mais.

- Gerencie contas e registro

  Gerencie contas de usuários do sistema operacional (SO) em instâncias e registro nas instâncias do Windows.

- Controlar o acesso aos recursos

  Controle o acesso aos recursos do Fleet Manager usando as políticas do AWS Identity and Access Management (IAM). Com essas políticas, você pode controlar quais usuários individuais ou grupos na sua organização podem usar vários recursos do Fleet Manager e quais nós gerenciados eles podem gerenciar.

Tópicos

- Configurar o Fleet Manager

- Trabalhar com nós gerenciados
- Gerenciar instâncias do EC2 automaticamente com a Configuração de gerenciamento de hosts padrão
- Conectar a uma instância gerenciada pelo Windows Server usando o Remote Desktop
- Gerenciar volumes do Amazon EBS em instâncias gerenciadas
- Como acessar o portal da base de conhecimento do Red Hat
- Solução de problemas de disponibilidade do nó gerenciado

Configurar o Fleet Manager

Antes que os usuários na Conta da AWS possam usar o Fleet Manager, uma ferramenta do AWS Systems Manager, para monitorar e gerenciar nós gerenciados, é necessário que recebam as permissões necessárias. Além disso, para que instâncias do Amazon Elastic Compute Cloud (Amazon EC2), dispositivos principais do AWS IoT Greengrass, servidores on-premises, dispositivos de borda e máquinas virtuais (VMs) sejam monitoradas e gerenciadas usando o Fleet Manager, são necessários nós gerenciados do Systems Manager. Um nó gerenciado é qualquer máquina configurada para uso com o Systems Manager em ambientes híbridos e multinuvem.

Isso significa que seus nós devem atender a determinados pré-requisitos e ser configurados com o Agente do AWS Systems Manager (SSM Agent).

Dependendo do tipo de máquina, consulte um dos tópicos a seguir para garantir que suas máquinas atendam aos requisitos de nós gerenciados.

- Instâncias do Amazon EC2: Gerenciar instâncias do EC2 com o Systems Manager

> **Tip**
>
> Você também pode usar o Quick Setup, uma ferramenta do AWS Systems Manager, para ajudar a configurar rapidamente suas instâncias do Amazon EC2 como instâncias gerenciadas em uma conta individual. Se sua empresa ou organização usa o AWS Organizations, você também poderá configurar instâncias em várias unidades organizacionais (UOs) e Regiões da AWS. Para obter mais informações sobre como usar o Quick Setup para configurar instâncias gerenciadas, consulte o Configurar o gerenciamento de host do Amazon EC2 usando a Quick Setup.

- On-premises e outros tipos de servidor na nuvem: Gerenciar nós em ambientes híbridos e multinuvem com o Systems Manager
- Dispositivos (de borda) do AWS IoT Greengrass: Gerenciar dispositivos de borda com o Systems Manager

Políticas de exemplo

- Controlar o acesso ao Fleet Manager

Controlar o acesso ao Fleet Manager

Para usar o Fleet Manager, uma ferramenta do AWS Systems Manager, o usuário ou perfil do AWS Identity and Access Management (IAM) deve ter as permissões necessárias. Você pode criar uma política do IAM que forneça acesso a todos os recursos do Fleet Manager ou modificar sua política para conceder acesso aos recursos que você escolher. Em seguida, você concederá essas permissões a usuários ou identidades em sua conta.

Tarefa 1: crie políticas do IAM para definir permissões de acesso

Siga um dos métodos fornecidos no tópico a seguir no Guia do usuário do IAM para criar um IAM para fornecer identidades (usuários, perfis ou grupos de usuários) com acesso ao Fleet Manager:

- Defina permissões personalizadas do IAM com políticas gerenciadas pelo cliente

Você pode usar um dos exemplos de políticas que fornecemos abaixo ou modificá-los de acordo com as permissões que deseja conceder. Nós oferecemos amostras de políticas disponibilizar um acesso total ao Fleet Manager, bem como o acesso somente leitura.

Tarefa 2: anexe as políticas do IAM aos usuários para conceder permissões

Depois de criar a política ou políticas do IAM que definem as permissões de acesso ao Fleet Manager, use um dos procedimentos presentes no Guia do usuário do IAM para conceder essas permissões às identidades em sua conta:

- Adicionar permissões de identidade do IAM (console)
- Adicionar permissões de identidade do IAM (AWS CLI)
- Adicionar permissões de identidade do IAM (API da AWS)

Tópicos

- Exemplo de política para acesso de administrador do Fleet Manager

- Exemplo de política para acesso somente leitura do Fleet Manager

Exemplo de política para acesso de administrador do Fleet Manager

A política a seguir fornece permissões para todos os recursos do Fleet Manager Isso significa que um usuário pode criar e excluir usuários e grupos locais, modificar a associação de grupo para qualquer grupo local e modificar chaves ou valores do registro do Windows Server. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EC2",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags",
                "ec2:DeleteTags",
                "ec2:DescribeInstances",
                "ec2:DescribeTags"
            ],
            "Resource": "*"
        },
        {
            "Sid": "General",
            "Effect": "Allow",
            "Action": [
                "ssm:AddTagsToResource",
                "ssm:DescribeInstanceAssociationsStatus",
                "ssm:DescribeInstancePatches",
                "ssm:DescribeInstancePatchStates",
                "ssm:DescribeInstanceProperties",
                "ssm:GetCommandInvocation",
                "ssm:GetServiceSetting",
                "ssm:GetInventorySchema",
                "ssm:ListComplianceItems",
                "ssm:ListInventoryEntries",
                "ssm:ListTagsForResource",
                "ssm:ListCommandInvocations",
                "ssm:ListAssociations",
```

```
                "ssm:RemoveTagsFromResource"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DefaultHostManagement",
            "Effect": "Allow",
            "Action": [
                "ssm:ResetServiceSetting",
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:servicesetting/ssm/
managed-instance/default-ec2-instance-management-role"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/service-role/
AWSSystemsManagerDefaultEC2InstanceManagementRole",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ssm.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "SendCommand",
            "Effect": "Allow",
            "Action": [
                "ssm:GetDocument",
                "ssm:SendCommand",
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ec2:*:111122223333:instance/*",
                "arn:aws:ssm:*:111122223333:managed-instance/*",
                "arn:aws:ssm:*:111122223333:document/SSM-SessionManagerRunShell",
                "arn:aws:ssm:*:*:document/AWS-PasswordReset",
                "arn:aws:ssm:*:*:document/AWSFleetManager-AddUsersToGroups",
                "arn:aws:ssm:*:*:document/AWSFleetManager-CopyFileSystemItem",
```

```
                "arn:aws:ssm:*:*:document/AWSFleetManager-CreateDirectory",
                "arn:aws:ssm:*:*:document/AWSFleetManager-CreateGroup",
                "arn:aws:ssm:*:*:document/AWSFleetManager-CreateUser",
                "arn:aws:ssm:*:*:document/AWSFleetManager-CreateUserInteractive",
                "arn:aws:ssm:*:*:document/AWSFleetManager-
CreateWindowsRegistryKey",
                "arn:aws:ssm:*:*:document/AWSFleetManager-DeleteFileSystemItem",
                "arn:aws:ssm:*:*:document/AWSFleetManager-DeleteGroup",
                "arn:aws:ssm:*:*:document/AWSFleetManager-DeleteUser",
                "arn:aws:ssm:*:*:document/AWSFleetManager-
DeleteWindowsRegistryKey",
                "arn:aws:ssm:*:*:document/AWSFleetManager-
DeleteWindowsRegistryValue",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetDiskInformation",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetFileContent",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetFileSystemContent",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetGroups",
                "arn:aws:ssm:*:*:document/AWSFleetManager-
GetPerformanceCounters",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetProcessDetails",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetUsers",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetWindowsEvents",
                "arn:aws:ssm:*:*:document/AWSFleetManager-
GetWindowsRegistryContent",
                "arn:aws:ssm:*:*:document/AWSFleetManager-MountVolume",
                "arn:aws:ssm:*:*:document/AWSFleetManager-MoveFileSystemItem",
                "arn:aws:ssm:*:*:document/AWSFleetManager-RemoveUsersFromGroups",
                "arn:aws:ssm:*:*:document/AWSFleetManager-RenameFileSystemItem",
                "arn:aws:ssm:*:*:document/AWSFleetManager-
SetWindowsRegistryValue",
                "arn:aws:ssm:*:*:document/AWSFleetManager-StartProcess",
                "arn:aws:ssm:*:*:document/AWSFleetManager-TerminateProcess"
            ]
        },
        {
            "Sid": "TerminateSession",
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/aws:ssmmessages:session-id": [
```

```
                        "${aws:userid}"
                    ]
                }
            }
        }
    ]
}
```

Exemplo de política para acesso somente leitura do Fleet Manager

A política a seguir fornece permissões para os recursos somente leitura do Fleet Manager. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EC2",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeTags"
            ],
            "Resource": "*"
        },
        {
            "Sid": "General",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeInstanceAssociationsStatus",
                "ssm:DescribeInstancePatches",
                "ssm:DescribeInstancePatchStates",
                "ssm:DescribeInstanceProperties",
                "ssm:GetCommandInvocation",
                "ssm:GetServiceSetting",
                "ssm:GetInventorySchema",
                "ssm:ListComplianceItems",
                "ssm:ListInventoryEntries",
                "ssm:ListTagsForResource",
                "ssm:ListCommandInvocations",
```

```
                "ssm:ListAssociations"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SendCommand",
            "Effect": "Allow",
            "Action": [
                "ssm:GetDocument",
                "ssm:SendCommand",
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ec2:*:111122223333:instance/*",
                "arn:aws:ssm:*:111122223333:managed-instance/*",
                "arn:aws:ssm:*:111122223333:document/SSM-SessionManagerRunShell",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetDiskInformation",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetFileContent",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetFileSystemContent",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetGroups",
                "arn:aws:ssm:*:*:document/AWSFleetManager-
GetPerformanceCounters",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetProcessDetails",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetUsers",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetWindowsEvents",
                "arn:aws:ssm:*:*:document/AWSFleetManager-
GetWindowsRegistryContent"
            ]
        },
        {
            "Sid": "TerminateSession",
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/aws:ssmmessages:session-id": [
                        "${aws:userid}"
                    ]
                }
            }
        }
```

```
    ]
}
```

Trabalhar com nós gerenciados

Um nó gerenciado é qualquer máquina configurada para o AWS Systems Manager. É possível configurar os seguintes tipos de máquina como nós gerenciados:

- Instâncias do Amazon Elastic Compute Cloud (Amazon EC2)
- Servidores em suas próprias instalações (servidores on-premises)
- Dispositivos principais do AWS IoT Greengrass
- AWS IoT e dispositivos de borda que não são da AWS
- Máquinas virtuais (VMs), inclusive VMs em outros ambientes de nuvem

No console do Systems Manager, toda máquina com o prefixo “mi-” foi configurada como nó gerenciado usando uma ativação híbrida. Os dispositivos de borda exibem seu nome da coisa do AWS IoT.

> **Note**
>
> O único recurso suportado para instâncias do macOS está visualizando o sistema de arquivos.

Sobre camadas de instâncias do Systems Manager

O AWS Systems Manager oferece um nível de instâncias padrão e um nível de instâncias avançadas. As duas opções oferecem suporte a nós gerenciados em seu ambiente híbrido e multinuvem. O nível de instâncias padrão permite registrar no máximo 1.000 máquinas por Conta da AWS e por Região da AWS. Se precisar registrar mais de 1.000 máquinas em uma única conta e região, use o nível de instâncias avançadas. Você pode criar quantos nós gerenciados quiser no nível de instâncias avançadas. Todos os nós gerenciados configurados para o Systems Manager são cobrados de acordo com o uso. Para obter mais informações sobre como habilitar instâncias avançadas, consulte Ativar o nível de instâncias avançadas. Para obter mais informações sobre precificação, consulte Precificação do AWS Systems Manager.

Observe as seguintes informações adicionais sobre o nível de instâncias padrão e o nível de instâncias avançadas:

- Instâncias avançadas também permitem que você se conecte a seus nós que não são do EC2 em um ambiente híbrido e multinuvem usando o AWS Systems Manager Session Manager. O Session Manager fornece acesso via shell interativo às suas instâncias. Para obter mais informações, consulte AWS Systems Manager Session Manager.
- A cota de instâncias padrão também se aplica a instâncias do EC2 que usam uma ativação do Systems Manager on-premises (o que não é um cenário comum).
- Para aplicar patches em aplicações lançadas pela Microsoft em VMs (máquinas virtuais) e instâncias on-premises, ative o nível de instâncias avançadas. Há uma cobrança para o uso do nível de instâncias avançadas. Não há custo adicional para aplicar patches em aplicações lançadas pela Microsoft nas instâncias do Amazon Elastic Compute Cloud (Amazon EC2). Para obter mais informações, consulte Aplicações de patches lançadAs pela Microsoft no Windows Server.

Exibir nós gerenciados

Se você não conseguir ver os nós gerenciados listados no console, faça o seguinte:

1. Verifique se o console está aberto na Região da AWS em que você criou os nós gerenciados. Você pode trocar de região usando a lista no canto superior direito do console.
2. Verifique se as etapas de configuração de seus nós gerenciados atendem aos requisitos do Systems Manager. Para mais informações, consulte Configurar nós gerenciados para o AWS Systems Manager.
3. Para máquinas que não são do EC2, verifique se você concluiu o processo de ativação híbrida. Para obter mais informações, consulte Gerenciar nós em ambientes híbridos e multinuvem com o Systems Manager.

Observe as seguintes informações adicionais:

- O console do Fleet Manager não exibe nós do Amazon EC2 que foram terminados.
- O Systems Manager requer referências de tempo precisas para realizar operações em suas máquinas. Se a data e a hora dos nós gerenciados não estiverem definidas corretamente, talvez elas não correspondam à data de assinatura das solicitações da API. Para obter mais informações, consulte Casos de uso e melhores práticas.

- Quando você cria ou edita tags, o sistema pode levar até uma hora para exibir as alterações no filtro da tabela.
- Depois que o status de um nó gerenciado `Connection Lost` durar pelo menos 30 dias, talvez o nó não esteja mais listado no Fleet Manager console. Para restaurá-lo na lista, o problema que causou a perda da conexão deve ser resolvido. Para obter dicas de solução de problemas, consulte Solução de problemas de disponibilidade do nó gerenciado.

Verifique o suporte ao Systems Manager em um nó gerenciado

O AWS Config fornece as regras gerenciadas da AWS, que são regras predefinidas e personalizáveis que o AWS Config utiliza para avaliar se as configurações de recursos da AWS são compatíveis com as melhores práticas. AWS Config Regras gerenciadas incluem a regra ec2-instance-managed-by-systems-manager. Essa regra verifica se as instâncias do Amazon EC2 em sua conta são gerenciadas pelo Systems Manager. Para obter mais informações, consulte Regras gerenciadas do AWS Config.

Reforce o procedimento de segurança em nós gerenciados

Para obter informações sobre como reforçar seu procedimento de segurança em relação a comandos em nível raiz não autorizados em seus nós gerenciados, consulte Restringir o acesso aos comandos em nível raiz por meio do SSM Agent

Cancelar o registro de nós gerenciados

Você pode cancelar o registro dos nós gerenciados a qualquer momento. Por exemplo, se você estiver gerenciando vários nós com a mesma função do AWS Identity and Access Management (IAM) e você notar qualquer tipo de comportamento malicioso, você poderá cancelar o registro de qualquer número de máquinas em qualquer momento. (Para registrar novamente a mesma máquina, você deve usar um Código de ativação e ID de ativação híbridos diferentes dos usados anteriormente para registrá-la.) Para obter informações sobre como cancelar o registro de nós gerenciados, consulte Cancelar o registro de nós gerenciados em um ambiente híbrido e multinuvem.

Tópicos

- Configurar níveis de instâncias
- Redefinir senhas em nós gerenciados
- Cancelar o registro de nós gerenciados em um ambiente híbrido e multinuvem
- Trabalhar com sistemas de arquivos do sistema operacional usando o Fleet Manager

- Monitoramento da performance dos nós
- Trabalhando com processos
- Visualizar logs em nós gerenciados
- Gerenciar contas e grupos de usuários do sistema operacional em nós gerenciados usando o Fleet Manager
- Gerenciar o registro do Windows em nós gerenciados

Configurar níveis de instâncias

Este tópico descreve os cenários nos quais você deve ativar a camada de instância avançada.

O AWS Systems Manager oferece um nível de instâncias padrão e um nível de instâncias avançadas para máquinas que não são do EC2 em um ambiente híbrido e multinuvem.

Você pode registrar até mil nós ativados para ambiente híbrido padrão por conta por Região da AWS sem custo adicional. No entanto, para registrar mais de 1.000 nós híbridos, você precisa ativar o nível de instâncias avançadas. Há uma cobrança para o uso do nível de instâncias avançadas. Para obter mais informações, consulte Preços do AWS Systems Manager.

Mesmo com menos de mil nós ativados para ambiente híbrido registrados, há dois outros cenários exigem o nível de instâncias avançadas:

- Você quer usar o Session Manager para se conectar a nós que não são do EC2.
- Você deseja aplicar patches em aplicativos (não em sistemas operacionais) lançados pela Microsoft em nós que não são do EC2.

> **Note**
>
> Não há custo para aplicar patches em aplicativos lançados pela Microsoft nas instâncias do Amazon EC2.

Cenários detalhados do nível de instâncias avançadas

As informações a seguir fornecem detalhes sobre os três cenários para os quais você deve ativar o nível de instâncias avançadas.

Cenário 1: você deseja registrar mais de mil nós ativados para ambiente híbrido

Usando o nível de instâncias padrão, é possível registrar até mil nós que não são do EC2 em um ambiente híbrido e multinuvem por Região da AWS em uma conta específica sem custo adicional. Se precisar registrar mais de 1.000 nós que não sejam do EC2 em uma região, use o nível de instâncias avançadas. Então você poderá ativar quantas máquinas quiser em seu ambiente híbrido e multinuvem. As cobranças pelo nível de instâncias avançadas são feitas com base no número de nós avançados ativados como nós gerenciados do Systems Manager e nas horas de execução desses nós.

Todos os nós gerenciados do Systems Manager que usam o processo de ativação descrito em Criar uma ativação híbrida para registrar nós com o Systems Manager estarão sujeitos a cobranças se você exceder mil nós on-premises em uma região, em uma conta específica.

> **Note**
>
> Você também pode ativar instâncias existentes do Amazon Elastic Compute Cloud (Amazon EC2) usando ativações híbridas do Systems Manager e trabalhar com elas como instâncias não pertencentes ao EC2, p. ex., para fins de teste. Esses também se qualificariam como nós híbridos. Esse não é um cenário comum.

Cenário 2: aplicação de patches em aplicativos lançados pela Microsoft em nós habilitados para ambiente híbrido

Também é necessário usar o nível de instâncias avançadas se você quiser aplicar patches em aplicações lançadas pela Microsoft em nós que não são do EC2 em um ambiente híbrido e multinuvem. Se você ativar o nível de instâncias avançadas para aplicar patches em aplicativos da Microsoft em nós não pertencentes ao EC2, haverá cobranças para todos os nós on-premises, mesmo que você tenha menos de 1.000.

Não há custo adicional para aplicar patches em aplicações lançadas pela Microsoft nas instâncias do Amazon Elastic Compute Cloud (Amazon EC2). Para ter mais informações, consulte Aplicações de patches lançadAs pela Microsoft no Windows Server.

Cenário 3: estabelecendo conexão com nós habilitados para ambiente híbrido usando o Session Manager

O Session Manager fornece acesso ao shell interativo para suas instâncias. Para estabelecer conexão com nós gerenciados ativados para ambiente híbrido usando o Session Manager, é

necessário ativar o nível de instâncias avançadas. Consequentemente, haverá cobranças para todos os nós habilitados para ambiente híbrido, mesmo que você tenha menos de 1.000.

Resumo: quando preciso do nível de instâncias avançadas?

Use a tabela a seguir para analisar quando você deve usar o nível de instâncias avançadas e para quais cenários haverá cobranças adicionais.

Cenário	Requer o nível de instâncias avançadas?	Há cobranças adicionais?
O número de nós habilitados para ambiente híbrido na minha região em uma conta específica é superior a 1.000.	Sim	Sim
Quero usar o Patch Manager para aplicar patches em aplicativos lançados pela Microsoft em qualquer número de nós habilitados para ambiente híbrido, mesmo que o número seja inferior a 1.000.	Sim	Sim
Quero usar o Session Manager para estabelecer conexão com qualquer número de nós habilitados para ambiente híbrido, mesmo que o número seja inferior a 1.000.	Sim	Sim
1. O número de nós habilitados para ambiente híbrido em uma região em uma conta específica é de até 1.000 nós; e	Não	Não

Cenário	Requer o nível de instâncias avançadas?	Há cobranças adicionais?
2. Não estou aplicando patches em aplicativos da Microsoft em nenhum nó habilitado para ambiente híbrido; e 3. Não estou estabelecendo conexão com nenhum nó habilitado para ambiente híbrido usando o Session Manager.		

Tópicos

- Ativar o nível de instâncias avançadas
- Revesão do nível de instâncias avançadas para o nível de instâncias padrão

Ativar o nível de instâncias avançadas

O AWS Systems Manager oferece um nível de instâncias padrão e um nível de instâncias avançadas para máquinas que não são do EC2 em um ambiente híbrido e multinuvem. O nível de instâncias padrão permite registrar no máximo 1.000 máquinas habilitadas para ambiente híbrido por Conta da AWS e por Região da AWS. O nível de instâncias avançadas também é necessário para usar Patch Manager a fim de aplicar patches em aplicativos lançados pela Microsoft em nós que não sejam do EC2 e para se conectar a nós que não sejam do EC2 usando o Session Manager. Para obter mais informações, consulte Ativar o nível de instâncias avançadas.

Esta seção descreve como configurar seu ambiente híbrido e multinuvem para usar o nível de instâncias avançadas.

Antes de começar

Reveja os detalhes de preço de instâncias avançadas. Instâncias avançadas estão disponíveis conforme uso. Para obter mais informações, consulte Definição de preço da AWS Systems Manager.

Configurar permissões para ativar o nível de instâncias avançadas

Verifique se você tem permissão no AWS Identity and Access Management (IAM) para alterar o ambiente do nível de instâncias padrão para o nível de instâncias avançadas. Você deve ter a política do IAM `AdministratorAccess` anexada ao seu usuário, grupo ou perfil, ou deve ter permissões para alterar a configuração do serviço do nível de ativação do Systems Manager. A configuração de nível de ativação usa as seguintes ações de API:

- GetServiceSetting
- UpdateServiceSetting
- ResetServiceSetting

Use o procedimento a seguir para adicionar uma política em linha do IAM a uma conta de usuário. Essa política permite que um usuário visualize a configuração atual do nível de instância gerenciada. Essa política também permite que o usuário altere ou redefina a configuração atual na Conta da AWS e na Região da AWS especificadas.

1. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Usuários.
3. Na lista, selecione o nome do grupo, do usuário para incorporar uma política.
4. Escolha a aba Permissões.
5. No lado direito da página, em Permission policies (Políticas de permissões), escolha Add inline policy (Adicionar política em linha).
6. Selecione a guia JSON.
7. Substitua o conteúdo padrão pela declaração a seguir:

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetServiceSetting"
            ],
```

```
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:ResetServiceSetting",
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:servicesetting/
ssm/managed-instance/activation-tier"
        }
    ]
}
```

8. Selecione Revisar política.
9. Na página Revisar política, em Nome, digite um nome para a política em linha. Por exemplo: **`Managed-Instances-Tier`**.
10. Escolha Criar política.

Os administradores podem especificar permissão somente leitura atribuindo a seguinte política em linha ao usuário.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetServiceSetting"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "ssm:ResetServiceSetting",
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "*"
```

```
        }
    ]
}
```

Para obter mais informações sobre como criar e editar políticas de usuários do IAM, consulte Criar políticas do IAM, no Manual do usuário do IAM.

Ativar o nível de instâncias avançadas (console)

O procedimento a seguir mostra como usar o console do Systems Manager para alterar todos os nós que não são do EC2 que foram adicionados usando a ativação de instância gerenciada, na Conta da AWS e Região da AWS especificadas, para usar o nível de instâncias avançadas.

Antes de começar

Verifique se o console está aberto na Região da AWS em que você criou as instâncias gerenciadas. Você pode trocar de região usando a lista no canto superior direito do console.

Verifique se você concluiu os requisitos de configuração para instâncias do Amazon Elastic Compute Cloud (Amazon EC2) e máquinas que não são do EC2 em um ambiente híbrido e multinuvem. Para mais informações, consulte Configurar nós gerenciados para o AWS Systems Manager.

> **Important**
>
> O procedimento a seguir descreve como alterar uma configuração em nível de conta. Essa alteração resultará em cobranças para a sua conta.

Para ativar o nível de instâncias avançadas (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Escolha Configurações e, em seguida, escolha Alterar configurações de nível de instância.
4. Revise as informações na caixa de diálogo sobre a alteração das configurações da conta.
5. Se aprovar, escolha a opção para aceitar e depois escolha Alterar configuração.

O sistema pode demorar vários minutos para concluir o processo de mover todas as instâncias do nível de instâncias padrão para o nível de instâncias avançadas.

> **Note**
>
> Para obter informações sobre como mudar de volta para o nível de instâncias padrão, consulte Revesão do nível de instâncias avançadas para o nível de instâncias padrão.

Ativar o nível de instâncias avançadas (AWS CLI)

O procedimento a seguir mostra como usar a AWS Command Line Interface para alterar todos os servidores on-premises e VMs que foram adicionados usando a ativação de instância gerenciada, na Conta da AWS e Região da AWS especificadas, para usar o nível de instâncias avançadas.

> **Important**
>
> O procedimento a seguir descreve como alterar uma configuração em nível de conta. Essa alteração resultará em cobranças para a sua conta.

Para ativar o nível de instâncias avançadas usando a AWS CLI

1. Abra a AWS CLI e execute o comando a seguir. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   Linux & macOS

   ```
   aws ssm update-service-setting \
       --setting-id arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-instance/activation-tier \
       --setting-value advanced
   ```

   Windows

   ```
   aws ssm update-service-setting ^
       --setting-id arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-instance/activation-tier ^
       --setting-value advanced
   ```

   Não haverá saída se o comando for bem-sucedido.

2. Execute o comando a seguir para visualizar as configurações de serviço atuais para nós gerenciados na Conta da AWS e Região da AWS atuais.

   Linux & macOS

   ```
   aws ssm get-service-setting \
       --setting-id arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
   instance/activation-tier
   ```

   Windows

   ```
   aws ssm get-service-setting ^
       --setting-id arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
   instance/activation-tier
   ```

   O comando retorna informações como as seguintes.

   ```
   {
       "ServiceSetting": {
           "SettingId": "/ssm/managed-instance/activation-tier",
           "SettingValue": "advanced",
           "LastModifiedDate": 1555603376.138,
           "LastModifiedUser": "arn:aws:sts::123456789012:assumed-role/
   Administrator/User_1",
           "ARN": "arn:aws:ssm:us-east-2:123456789012:servicesetting/ssm/managed-
   instance/activation-tier",
           "Status": "PendingUpdate"
       }
   }
   ```

Ativar o nível de instâncias avançadas (PowerShell)

O procedimento a seguir mostra como usar a AWS Tools for Windows PowerShell para alterar todos os servidores on-premises e VMs que foram adicionados usando a ativação de instância gerenciada, na Conta da AWS e Região da AWS especificadas, para usar o nível de instâncias avançadas.

> **Important**
>
> O procedimento a seguir descreve como alterar uma configuração em nível de conta. Essa alteração resultará em cobranças para a sua conta.

Para ativar o nível de instâncias avançadas usando o PowerShell

1. Abra o AWS Tools for Windows PowerShell e execute o comando a seguir. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   ```
   Update-SSMServiceSetting `
       -SettingId "arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
   instance/activation-tier" `
       -SettingValue "advanced"
   ```

   Não haverá saída se o comando for bem-sucedido.

2. Execute o comando a seguir para visualizar as configurações de serviço atuais para nós gerenciados na Conta da AWS e Região da AWS atuais.

   ```
   Get-SSMServiceSetting `
       -SettingId "arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
   instance/activation-tier"
   ```

   O comando retorna informações como as seguintes.

   ```
   ARN:arn:aws:ssm:us-east-2:123456789012:servicesetting/ssm/managed-instance/
   activation-tier
   LastModifiedDate : 4/18/2019 4:02:56 PM
   LastModifiedUser : arn:aws:sts::123456789012:assumed-role/Administrator/User_1
   SettingId        : /ssm/managed-instance/activation-tier
   SettingValue     : advanced
   Status           : PendingUpdate
   ```

O sistema pode demorar vários minutos para concluir o processo de mover todas os nós do nível de instâncias padrão para o nível de instâncias avançadas.

> **Note**
>
> Para obter informações sobre como mudar de volta para o nível de instâncias padrão, consulte Revesão do nível de instâncias avançadas para o nível de instâncias padrão.

Revesão do nível de instâncias avançadas para o nível de instâncias padrão

Esta seção descreve como reverter nós ativados para ambientes híbridos em execução no nível de instâncias avançadas para o nível de instâncias padrão. Essa configuração se aplica a todos os nós ativados para ambiente híbrido em uma Conta da AWS e em uma única Região da AWS.

Antes de começar

Revise as informações importantes a seguir.

> **Note**
>
> - Não será possível reverter para o nível de instância padrão se estiver executando mais de mil nós ativados para ambientes híbridos na conta e na região. Primeiro, é necessário cancelar o registro de nós até ter 1.000 ou menos. Isso também se aplica a instâncias do Amazon Elastic Compute Cloud (Amazon EC2) que usam uma ativação híbrida do Systems Manager (o que não é um cenário comum). Para ter mais informações, consulte Cancelar o registro de nós gerenciados em um ambiente híbrido e multinuvem.
> - Após a reversão, não será possível usar o Session Manager, uma ferramenta do AWS Systems Manager, para acessar interativamente os nós ativados para ambientes híbridos.
> - Após a reversão, não será possível usar o Patch Manager, uma ferramenta do AWS Systems Manager, para aplicar patches em aplicações lançadas pela Microsoft em nós ativados para ambientes híbridos.
> - O processo de reverter todos nós ativados para ambientes híbridos para o nível de instância padrão pode levar 30 minutos ou mais para ser concluído.

Esta seção descreve como reverter todos os nós ativados para ambientes híbridos em uma Conta da AWS e Região da AWS do nível de instâncias avançadas para o nível de instâncias padrão.

Reverter para o nível de instâncias padrão (console)

O procedimento a seguir mostra como usar o console do Systems Manager para alterar todos os nós ativados para ambiente híbrido em seu ambiente híbrido e multinuvem para usar o nível de instâncias padrão na Conta da AWS e na Região da AWS especificadas.

Como reverter para o nível de instâncias padrão (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione a lista suspensa Account settings (Configurações da conta) e escolha Instance tier settings (Configurações do nível de instância).
4. Escolha Change account setting (Alterar configuração da conta).
5. Reveja as informações no pop-up sobre como alterar as configurações da conta e, se aprovar, escolha a opção para aceitar e continuar.

Reverter para o nível de instâncias padrão (AWS CLI)

O procedimento a seguir mostra como usar a AWS Command Line Interface para alterar os nós ativados para ambiente híbrido em seu ambiente híbrido e multinuvem para usar o nível de instâncias padrão na Conta da AWS e na Região da AWS especificadas.

Como reverter para o nível de instâncias padrão usando a AWS CLI

1. Abra a AWS CLI e execute o comando a seguir. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   Linux & macOS

   ```
   aws ssm update-service-setting \
       --setting-id arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-instance/activation-tier \
       --setting-value standard
   ```

   Windows

   ```
   aws ssm update-service-setting ^
       --setting-id arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-instance/activation-tier ^
   ```

```
    --setting-value standard
```

Não haverá saída se o comando for bem-sucedido.

2. Execute o comando a seguir 30 minutos depois para visualizar as configurações das instâncias gerenciadas na Conta da AWS e Região da AWS atuais.

   Linux & macOS

   ```
   aws ssm get-service-setting \
       --setting-id arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
   instance/activation-tier
   ```

   Windows

   ```
   aws ssm get-service-setting ^
       --setting-id arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
   instance/activation-tier
   ```

   O comando retorna informações como as seguintes.

   ```
   {
       "ServiceSetting": {
           "SettingId": "/ssm/managed-instance/activation-tier",
           "SettingValue": "standard",
           "LastModifiedDate": 1555603376.138,
           "LastModifiedUser": "System",
           "ARN": "arn:aws:ssm:us-east-2:123456789012:servicesetting/ssm/managed-
   instance/activation-tier",
           "Status": "Default"
       }
   }
   ```

   O status muda para Padrão após a solicitação ser aprovada.

Reverter para o nível de instâncias padrão (PowerShell)

O procedimento a seguir mostra como usar AWS Tools for Windows PowerShell para alterar os nós ativados para ambiente híbrido em seu ambiente híbrido e multinuvem para usar o nível de instâncias padrão na Conta da AWS e na Região da AWS especificadas.

Como reverter para o nível de instâncias padrão usando o PowerShell

1. Abra o AWS Tools for Windows PowerShell e execute o comando a seguir.

```
Update-SSMServiceSetting `
    -SettingId "arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
instance/activation-tier" `
    -SettingValue "standard"
```

   Não haverá saída se o comando for bem-sucedido.

2. Execute o comando a seguir 30 minutos depois para visualizar as configurações das instâncias gerenciadas na Conta da AWS e Região da AWS atuais.

```
Get-SSMServiceSetting `
    -SettingId "arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
instance/activation-tier"
```

   O comando retorna informações como as seguintes.

```
ARN: arn:aws:ssm:us-east-2:123456789012:servicesetting/ssm/managed-instance/
activation-tier
LastModifiedDate : 4/18/2019 4:02:56 PM
LastModifiedUser : System
SettingId        : /ssm/managed-instance/activation-tier
SettingValue     : standard
Status           : Default
```

   O status muda para Padrão após a solicitação ser aprovada.

Redefinir senhas em nós gerenciados

Você pode redefinir a senha de qualquer usuário em um nó gerenciado. Isso inclui instâncias do Amazon Elastic Compute Cloud (Amazon EC2), dispositivos principais do AWS IoT Greengrass,

servidores on-premises, dispositivos de borda e máquinas virtuais (VMs) gerenciadas pelo AWS Systems Manager. A funcionalidade de redefinição de senha é incorporada ao Session Manager, uma ferramenta do AWS Systems Manager. Você pode usar essa funcionalidade para se conectar aos nós gerenciados sem abrir portas de entrada, manter bastion hosts nem gerenciar chaves SSH.

A redefinição de senha útil quando um usuário esquecer a senha, ou quando você quiser atualizar rapidamente uma senha sem fazer uma conexão RDP ou SSH a um nó gerenciado.

Pré-requisitos

Antes de poder redefinir a senha em um nó gerenciado, os seguintes requisitos devem ser atendidos:

- O nó gerenciado no qual você quiser alterar uma senha deve ser um nó gerenciado do Systems Manager. Além disso, o SSM Agent versão 2.3.668.0 ou posterior deve ser instalado em um nó gerenciado.) Para obter informações sobre como instalar ou atualizar o SSM Agent, consulte Trabalhar com o SSM Agent.
- A funcionalidade de redefinição de senha usa a configuração do Session Manager que está configurada na sua conta para conexão ao nó gerenciado. Portanto, os pré-requisitos para usar o Session Manager devem ter sido concluídos para a conta na Região da AWS atual. Para obter mais informações, consulte Configurar o Session Manager.

  > **Note**
  >
  > O suporte do Session Manager para n[os on-premises é fornecido somente para o nível de instâncias avançadas. Para obter mais informações, consulte Ativar o nível de instâncias avançadas.

- O usuário do AWS que estiver alterando a senha deverá ter a permissão `ssm:SendCommand` para o nó gerenciado. Para obter mais informações, consulte Restringir o acesso ao Run Command com base em etiquetas.

Restringir acesso

Você pode limitar a capacidade de um usuário de redefinir senhas para nós gerenciados específicos. Isso é feito usando políticas baseadas em identidade para a operação do Session Manager `ssm:StartSession` com o documento SSM `AWS-PasswordReset`. Para obter mais informações, consulte Controlar o acesso à sessão do usuário para as instâncias.

Criptografar dados

Ative a criptografia completa do AWS Key Management Service (AWS KMS) para os dados do Session Manager, para usar a opção de redefinição de senha para nós gerenciados. Para obter mais informações, consulte Ativar a criptografia de chaves do KMS de dados de sessão (console).

Redefina uma senha em um nó gerenciado

Você pode redefinir uma senha em um nó gerenciado do Systems Manager usando o console do Fleet Manager no Systems Manager ou a AWS Command Line Interface (AWS CLI).

Para alterar a senha em um nó gerenciado (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o botão ao lado do nó gerenciado que precisa de uma nova senha.
4. Escolha Ações da instância, Redefinir senha.
5. Em User name (Nome do usuário), digite o nome do usuário para o qual você está alterando a senha. Esse pode ser qualquer nome de usuário que tenha uma conta em seu nó.
6. Selecione Enviar.
7. Siga os prompts na janela de comando Enter new password (Inserir nova senha) para especificar a nova senha.

> **Note**
>
> Se a versão do SSM Agent nesse nó gerenciado não oferecer suporte a redefinições de senha, você deverá instalar uma versão compatível usando o Run Command, uma ferramenta do AWS Systems Manager.

Para redefinir a senha em um nó gerenciado (AWS CLI)

1. Para redefinir a senha de um usuário em um nó gerenciado, execute o comando a seguir. Substitua cada *`espaço reservado para recurso de exemplo`* por suas próprias informações.

> **Note**
>
> Para usar a AWS CLI para redefinir uma senha, o plugin do Session Manager deve estar instalado na máquina local. Para mais informações, consulte Instalar o plug-in do Session Manager para a AWS CLI.

Linux & macOS

```
aws ssm start-session \
    --target instance-id \
    --document-name "AWS-PasswordReset" \
    --parameters '{"username": ["user-name"]}'
```

Windows

```
aws ssm start-session ^
    --target instance-id ^
    --document-name "AWS-PasswordReset" ^
    --parameters username="user-name"
```

2. Siga os prompts na janela de comando Enter new password (Inserir nova senha) para especificar a nova senha.

Solucionar problemas de redefinições de senha em nós gerenciados

Muitos problemas de redefinição de senha podem ser resolvidos garantindo que você tenha concluído os Pré-requisitos de redefinição de senha. Para outros problemas, use as seguintes informações para ajudar você a solucionar problemas de redefinição de senha.

Tópicos

- Nó gerenciado não disponível
- SSM Agent não atualizado (console)
- As opções de redefinição de senha não são fornecidas (AWS CLI)
- Sem autorização para executar ssm:SendCommand
- Mensagem de erro do Session Manager

Nó gerenciado não disponível

Problema: você quer redefinir a senha de um nó gerenciado na página do console Managed instances (Instâncias gerenciadas), mas o nó não está na lista.

- Solução A: o nó gerenciado ao qual você quer se conectar pode não ter sido configurado para o Systems Manager. Para usar uma instância do EC2 com o Systems Manager, um perfil de instância do AWS Identity and Access Management (IAM), que fornece permissão ao Systems Manager para executar ações em suas instâncias, deve ser anexado à instância. Para obter informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager.

  Para usar uma máquina que não é do EC2 com o Systems Manager, crie um perfil de serviço do IAM que dê permissão ao Systems Manager para executar ações em seus nós gerenciados. Para obter mais informações, consulte Criar o perfil de serviço do IAM obrigatório para o Systems Manager em ambientes híbridos e multinuvem. O suporte do Session Manager para servidores on-premises e VMs é fornecido apenas para o nível de instâncias avançadas. Para obter mais informações, consulte Ativar o nível de instâncias avançadas.

SSM Agent não atualizado (console)

Problema: uma mensagem informa que a versão do SSM Agent não oferece suporte à funcionalidade de redefinição de senha.

- Solução: a versão 2.3.668.0 ou posterior do SSM Agent é necessária para executar redefinições de senhas. No console, você pode atualizar o agente em um nó gerenciado escolhendo Update SSM Agent (Atualizar o Agente do SSM).

  Uma versão atualizada do SSM Agent é lançada sempre que novas ferramentas são adicionadas ao Systems Manager ou sempre que atualizações são feitas nas ferramentas existentes. Deixar de usar a versão mais recente do agente pode impedir que seu nó gerenciado use várias ferramentas do Systems Manager. Por isso, recomendamos automatizar o processo de manter o SSM Agent atualizado em suas máquinas. Para mais informações, consulte Automatizar atualizações do SSM Agent. Inscreva-se na página Notas de versão do SSM Agent no GitHub para receber notificações sobre atualizações do SSM Agent.

As opções de redefinição de senha não são fornecidas (AWS CLI)

Problema: você se conecta com êxito a um nó gerenciado usando o comando `start-session` da AWS CLI. Você especificou o documento `AWS-PasswordReset` do SSM e forneceu um nome de usuário válido, mas os prompts para alterar a senha não são exibidos.

- Solução: a versão do SSM Agent em seu nó gerenciado não está atualizada. A versão 2.3.668.0 ou posterior é necessária para executar redefinições de senhas.

  Uma versão atualizada do SSM Agent é lançada sempre que novas ferramentas são adicionadas ao Systems Manager ou sempre que atualizações são feitas nas ferramentas existentes. Deixar de usar a versão mais recente do agente pode impedir que seu nó gerenciado use várias ferramentas do Systems Manager. Por isso, recomendamos automatizar o processo de manter o SSM Agent atualizado em suas máquinas. Para mais informações, consulte Automatizar atualizações do SSM Agent. Inscreva-se na página Notas de versão do SSM Agent no GitHub para receber notificações sobre atualizações do SSM Agent.

Sem autorização para executar **`ssm:SendCommand`**

Problema: você tenta se conectar a um nó gerenciado para alterar sua senha, mas recebe uma mensagem de erro informando que você não está autorizado a executar o `ssm:SendCommand` em um nó gerenciado.

- Solução: sua política do IAM deve incluir permissões para executar o comando `ssm:SendCommand`. Para mais informações, consulte Restringir o acesso ao Run Command com base em etiquetas.

Mensagem de erro do Session Manager

Problema: você recebe uma mensagem de erro relacionada ao Session Manager.

- Solução: o suporte à redefinição de senha exige que o Session Manager esteja configurado corretamente. Para obter informações, consulte Configurar o Session Manager e Solução de problemas do Session Manager.

Cancelar o registro de nós gerenciados em um ambiente híbrido e multinuvem

Se você não quiser mais gerenciar um servidor, um dispositivo de borda ou uma máquina virtual (VM) on-premises usando o AWS Systems Manager, poderá cancelar o registro. Cancelar o registro

de um nó ativado para ambientes híbridos o removerá da lista de nós gerenciados no Systems Manager. AWS Systems Manager O agente (SSM Agent) em execução no nó ativado para ambiente híbrido não poderá atualizar seu token de autorização porque ele não está mais registrado. O SSM Agent hibernará e reduzirá a frequência de ping para o Systems Manager na nuvem para uma vez a cada hora. O Systems Manager armazena o histórico de comandos para um nó gerenciado com registro cancelado por 30 dias.

Se você quiser registrar novamente um servidor, um dispositivo de borda ou uma máquina virtual (VM) on-premises, deverá usar um Código de ativação e um ID de ativação diferentes dos usados para registrar a máquina anteriormente. O Código de ativação e o ID de ativação não devem ter sido usados no número máximo de ativações especificado quando foram criados.

O procedimento a seguir descreve como cancelar o registro de um nó ativado para ambientes híbridos usando o console do Systems Manager. Para obter informações sobre como fazer isso usando a AWS Command Line Interface, consulte deregister-managed-instance (Cancelar o registro da instância gerenciada).

Para obter informações mais detalhadas, consulte os seguintes tópicos relacionados:

- Cancele o registro e registre um nó gerenciado novamente (Linux) (Linux)
- Cancelar o registro e registrar um nó gerenciado novamente (Windows Server) (Windows Server)

Para cancelar o registro de um nó ativado para ambiente híbrido (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Marque a caixa de seleção ao lado do nó gerenciado para o qual você deseja cancelar o registro.
4. Escolha Ações do nó, Ferramentas, Cancele o registro deste nó gerenciado.
5. Analise as informações na caixa de diálogo Cancelar o registro deste nó gerenciado. Se você aprovar, escolha Cancelar registro.

Trabalhar com sistemas de arquivos do sistema operacional usando o Fleet Manager

Você pode usar o Fleet Manager, uma ferramenta do AWS Systems Manager, para trabalhar com o sistema de arquivos em seus nós gerenciados. Usando o Fleet Manager, você pode exibir informações sobre os dados do diretório e do arquivo armazenados nos volumes anexados aos

nós gerenciados. Por exemplo, você pode visualizar o nome, o tamanho, a extensão, o proprietário e as permissões para suas pastas e arquivos. Até 10.000 linhas de dados de arquivo podem ser visualizadas como texto do console do Fleet Manager. Você também pode usar esse recurso para arquivos `tail`. Ao usar o `tail` para exibir os dados do arquivo, as últimas 10 linhas do arquivo são exibidas inicialmente. À medida que novas linhas de dados são gravadas no arquivo, a exibição é atualizada em tempo real. Como resultado, você pode revisar os dados de log do console, o que pode melhorar a eficiência da solução de problemas e da administração de sistemas. Além disso, você pode criar diretórios e copiar, cortar, colar, renomear ou excluir arquivos e diretórios.

Recomendamos criar um backup do registro ou tirar um snapshot dos volumes do Amazon Elastic Block Store (Amazon EBS) anexados ao nó gerenciado. Ao copiar ou cortar e colar arquivos, os arquivos e diretórios existentes no caminho de destino com o mesmo nome dos novos arquivos ou diretórios são substituídos. Problemas sérios podem ocorrer se você substituir ou modificar arquivos e diretórios do sistema. A AWS não garante que esses problemas possam ser resolvidos. Modifique os arquivos do sistema por sua conta e risco. Você é responsável por todas as alterações feitas em arquivos e diretórios e por garantir que tenha backups. A exclusão ou substituição de arquivos e diretórios não pode ser desfeita.

> **Note**
>
> O Fleet Manager usa o Session Manager, uma ferramenta do AWS Systems Manager, para obter prévias do texto e arquivos `tail`. Em instâncias do Amazon Elastic Compute Cloud (Amazon EC2), o perfil anexado às instâncias gerenciadas deve fornecer permissões para o Session Manager usar esse recurso. Para obter mais informações sobre como adicionar permissões do Session Manager a um perfil de instância, consulte Adicionar permissões do Session Manager a um perfil do IAM existente.

Tópicos

- Visualizar o sistema de arquivos do sistema operacional usando o Fleet Manager
- Visualizar arquivos do sistema operacional usando Fleet Manager
- Acompanhar arquivos do sistema operacional usando o Fleet Manager
- Copiar, recortar e colar arquivos ou diretórios do sistema operacional usando o Fleet Manager
- Renomeando arquivos e diretórios do sistema operacional usando o Fleet Manager
- Renomear arquivos e diretórios do sistema operacional usando o Fleet Manager
- Criar diretórios do sistema operacional usando o Fleet Manager

- Copiar, recortar e colar diretórios do sistema operacional usando o Fleet Manager

Visualizar o sistema de arquivos do sistema operacional usando o Fleet Manager

É possível usar o Fleet Manager para visualizar o sistema de arquivos do sistema operacional em um nó gerenciado do Systems Manager.

Para visualizar o sistema de arquivos do sistema operacional usando o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o link do nó gerenciado com o sistema de arquivos que deseja visualizar.
4. Escolha Ferramentas, Sistema de arquivos.

Visualizar arquivos do sistema operacional usando Fleet Manager

Você pode usar Fleet Manager para visualizar arquivos de texto em um sistema operacional.

Para visualizar o texto de arquivos com o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o link do nó gerenciado com os arquivos que deseja previsualizar.
4. Escolha Ferramentas, Sistema de arquivos.
5. Selecione o File name (Nome do arquivo) no diretório que contém o arquivo que você quer previsualizar.
6. Selecione o botão ao lado do arquivo cujo conteúdo você deseja visualizar.
7. Escolha Ações, Visualizar como texto.

Acompanhar arquivos do sistema operacional usando o Fleet Manager

Você pode usar Fleet Manager para acompanhar um arquivo em um nó gerenciado.

Para acompanhar arquivos do sistema operacional com o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.

3. Selecione o link do nó gerenciado com os arquivos que deseja acompanhar.
4. Escolha Ferramentas, Sistema de arquivos.
5. Selecione o File name (Nome do arquivo) no diretório que contém o arquivo que você quer acompanhar.
6. Selecione o botão ao lado do arquivo cujo conteúdo você deseja acompanhar.
7. Escolha Ações, Arquivo final.

Copiar, recortar e colar arquivos ou diretórios do sistema operacional usando o Fleet Manager

O Fleet Manager pode ser usado para copiar, recortar e colar arquivos do sistema operacional em um nó gerenciado.

Para copiar ou cortar e colar arquivos ou diretórios usando o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o link do nó gerenciado com os arquivos que deseja copiar ou cortar e colar.
4. Escolha Ferramentas, Sistema de arquivos.
5. Para copiar ou cortar um arquivo, selecione a opção File name (Nome do arquivo) no diretório que contém o arquivo que você quiser copiar ou cortar. Para copiar ou cortar um diretório, escolha o botão ao lado do diretório que você quiser copiar ou cortar e prossiga para a etapa 8.
6. Selecione o botão ao lado do arquivo cujo conteúdo você deseja copiar ou cortar.
7. No menu Actions (Ações), escolha Copy (Copiar) ou Cut (Cortar).
8. Na exibição File system (Sistema de arquivos), selecione o botão ao lado do diretório no qual deseja colar o arquivo.
9. No menu Actions (Ações), escolha Start (Iniciar).

Renomeando arquivos e diretórios do sistema operacional usando o Fleet Manager

O Fleet Manager pode ser usado para renomear arquivos diretórios em um nó gerenciado na sua conta.

Para renomear arquivos ou diretórios com o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.

2. No painel de navegação, escolha Fleet Manager.
3. Selecione o link do nó gerenciado com os arquivos ou diretórios que deseja renomear.
4. Escolha Ferramentas, Sistema de arquivos.
5. Para renomear um arquivo, selecione a opção File name (Nome do arquivo) no diretório que contém o arquivo que você deseja renomear. Para renomear um diretório, escolha o botão ao lado do diretório que você deseja renomear e, em seguida, prossiga para a etapa 8.
6. Selecione o botão ao lado do arquivo cujo conteúdo você deseja renomear.
7. Escolha Ações, Renomear.
8. Em Nome do arquivo, insira o novo nome para o arquivo e selecione Renomear.

Renomear arquivos e diretórios do sistema operacional usando o Fleet Manager

O Fleet Manager pode ser usado para excluir arquivos diretórios em um nó gerenciado na sua conta.

Para excluir arquivos ou diretórios usando o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o link do nó gerenciado com os arquivos ou diretórios que deseja excluir.
4. Escolha Ferramentas, Sistema de arquivos.
5. Para excluir um arquivo, selecione a opção File name (Nome do arquivo) no diretório que contém o arquivo que você deseja renomear. Para excluir um diretório, escolha o botão ao lado do diretório que você quer excluir e prossiga para a etapa 7.
6. Selecione o botão ao lado do arquivo cujo conteúdo você deseja excluir.
7. Escolha Ações, Excluir.

Criar diretórios do sistema operacional usando o Fleet Manager

O Fleet Manager pode ser usado para criar diretórios em um nó gerenciado na sua conta.

Para criar um diretório usando o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.

3. Selecione o link para o nó gerenciado no qual você deseja criar um diretório.
4. Escolha Ferramentas, Sistema de arquivos.
5. Selecione o File name (Nome do arquivo) no diretório em que você deseja criar um novo diretório.
6. Selecione Create directory (Criar diretório).
7. No campo Nome do diretório, insira o nome do novo diretório e selecione Criar diretório.

Copiar, recortar e colar diretórios do sistema operacional usando o Fleet Manager

O Fleet Manager pode ser usado para cortar, copiar e colar diretórios em um nó gerenciado na sua conta.

Para copiar ou cortar e colar diretórios com o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o link do nó gerenciado com os arquivos que deseja copiar ou cortar e colar.
4. Escolha Ferramentas, Sistema de arquivos.
5. Escolha o botão ao lado do diretório que você quiser copiar ou cortar e prossiga para a etapa 8.
6. No menu Actions (Ações), escolha Copy (Copiar) ou Cut (Cortar).
7. Na exibição File system (Sistema de arquivos), selecione o botão ao lado do diretório no qual deseja colar o arquivo.
8. No menu Actions (Ações), escolha Start (Iniciar).

Monitoramento da performance dos nós

É possível usar o Fleet Manager, uma ferramenta do AWS Systems Manager, para exibir dados de performance dos nós gerenciados em tempo real. Os dados de performance são recuperados dos contadores de performance.

Os contadores de performance a seguir estão disponíveis no Fleet Manager:

- Utilização da CPU
- Utilização de entrada/saída (E/S) do disco
- Tráfego de rede

- Uso de memória

> **Note**
>
> O Fleet Manager usa o Session Manager, uma ferramenta do AWS Systems Manager, para recuperar dados de performance. Em instâncias do Amazon Elastic Compute Cloud (Amazon EC2), o perfil anexado às instâncias gerenciadas deve fornecer permissões para o Session Manager usar esse recurso. Para obter mais informações sobre como adicionar permissões do Session Manager a um perfil de instância, consulte Adicionar permissões do Session Manager a um perfil do IAM existente.

Para visualizar os dados de performance com o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o botão ao lado do nó gerenciado cuja performance você deseja monitorar.
4. Escolha Exibir detalhes.
5. Escolha Ferramentas, Contadores de performance.

Trabalhando com processos

Você pode usar o Fleet Manager, uma ferramenta do AWS Systems Manager, para trabalhar com processos em seus nós gerenciados. Usando o Fleet Manager, você pode visualizar informações sobre processos. Por exemplo, você pode ver a utilização da CPU e o uso da memória dos processos, além de identificadores e threads. Com o Fleet Manager, você pode iniciar e encerrar processos no console.

> **Note**
>
> O Fleet Manager usa o Session Manager, uma ferramenta do AWS Systems Manager, para recuperar dados do processo. Em instâncias do Amazon Elastic Compute Cloud (Amazon EC2), o perfil anexado às instâncias gerenciadas deve fornecer permissões para o Session Manager usar esse recurso. Para obter mais informações sobre como adicionar permissões do Session Manager a um perfil de instância, consulte Adicionar permissões do Session Manager a um perfil do IAM existente.

Tópicos

- Visualizar detalhes sobre os processos do sistema operacional usando Fleet Manager
- Iniciar um processo de sistema operacional em um nó gerenciado usando o Fleet Manager
- Encerrar um processo do sistema operacional usando o Fleet Manager

Visualizar detalhes sobre os processos do sistema operacional usando Fleet Manager

É possível usar o Fleet Manager para visualizar detalhes sobre os processos em seus nós gerenciados.

Para visualizar detalhes sobre processos com o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o link do nó cujos processos você deseja visualizar.
4. Escolha Ferramentas, Processos.

Iniciar um processo de sistema operacional em um nó gerenciado usando o Fleet Manager

É possível usar o Fleet Manager para iniciar um processo em um nó gerenciado.

Para iniciar um processo com o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o link do nó gerenciado em que você deseja iniciar um processo.
4. Escolha Ferramentas, Processos.
5. Selecione Start new process (Iniciar um novo processo).
6. Em Nome do processo ou caminho completo, informe o nome do processo ou o caminho completo para o executável.
7. (Opcional) Em Diretório de trabalho, insira o caminho do diretório onde deseja que o processo seja executado.

Encerrar um processo do sistema operacional usando o Fleet Manager

Para encerrar um processo do sistema operacional usando o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o link do nó gerenciado em que você deseja iniciar um processo.
4. Escolha Ferramentas, Processos.
5. Selecione o botão ao lado do processo que você deseja encerrar.
6. Escolha Ações, Encerrar processo ou Ações, Encerrar árvore de processos.

> **Note**
>
> O encerramento de uma árvore de processos também encerra todos os processos e aplicações que usam esse processo.

Visualizar logs em nós gerenciados

É possível usar o Fleet Manager, uma ferramenta do AWS Systems Manager, para exibir dados de log armazenados em seus nós gerenciados. Para nós gerenciados do Windows, você pode visualizar logs de eventos do Windows e copiar os detalhes deles do console. Para ajudar você a pesquisar eventos, filtre os logs de eventos do Windows por Event level (Nível do evento), Event ID (ID do evento), Event source (Origem do evento), e Time created (Hora da criação). Você também pode exibir outros dados de log usando o procedimento para exibir o sistema de arquivos. Para obter mais informações sobre como visualizar o sistema de arquivos com o Fleet Manager, consulte Trabalhar com sistemas de arquivos do sistema operacional usando o Fleet Manager.

Para visualizar logs de eventos do Windows com o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o botão ao lado do nó gerenciado cujos logs de eventos você deseja exibir.
4. Escolha Exibir detalhes.
5. Escolha Ferramentas, Logs de eventos do Windows.
6. Selecione o Nome do log que contém os eventos que você deseja visualizar.

7. Selecione o botão ao lado do Log name (Nome do log) que você deseja visualizar e escolha View events (Visualizar eventos).
8. Selecione o botão ao lado do evento que você deseja visualizar e escolha View events (Visualizar eventos).
9. (Opcional) Selecione COPY as JSON (Copiar como JSON) para copiar os detalhes do evento para a área de transferência.

Gerenciar contas e grupos de usuários do sistema operacional em nós gerenciados usando o Fleet Manager

É possível usar o Fleet Manager, uma ferramenta do AWS Systems Manager, para gerenciar contas e grupos de usuários do sistema operacional (SO) em seus nós gerenciados. Por exemplo, você pode criar e excluir usuários e grupos. Além disso, você pode exibir detalhes como associação de grupo, funções de usuário e status.

> **Important**
>
> O Fleet Manager usa Run Command e Session Manager, ferramentas do AWS Systems Manager, para várias operações de gerenciamento de usuários. Como resultado, um usuário poderia conceder permissões a uma conta de usuário do sistema operacional que, de outra forma, não poderia fazê-lo. Isso ocorre porque o AWS Systems Manager Agent (SSM Agent) é executado em instâncias do Amazon Elastic Compute Cloud (Amazon EC2) usando permissões raiz (Linux) ou permissões do SISTEMA (Windows Server). Para obter mais informações sobre como restringir o acesso a comandos de nível raiz por meio do SSM Agent, consulte Restringir o acesso aos comandos em nível raiz por meio do SSM Agent. Para restringir o acesso a esse recurso, recomendamos criar políticas do AWS Identity and Access Management(IAM) para seus usuários que só permitem acesso às ações definidas por você. Para obter mais informações sobre como criar políticas do IAM, consulte Fleet Manager e Controlar o acesso ao Fleet Manager.

Tópicos

- Criar um usuário ou grupo do sistema operacional usando o Fleet Manager
- Atualizar a associação do usuário ou grupo com o Fleet Manager
- Excluir um usuário ou grupo do sistema operacional usando o Fleet Manager

Criar um usuário ou grupo do sistema operacional usando o Fleet Manager

> **Note**
>
> O Fleet Manager usa o Session Manager para definir senhas para novos usuários. Para instâncias do Amazon EC2, o perfil anexado às instâncias gerenciadas deve fornecer permissões para o Session Manager usar esse recurso. Para obter mais informações sobre como adicionar permissões do Session Manager a um perfil de instância, consulte Adicionar permissões do Session Manager a um perfil do IAM existente.

Em vez de fazer login diretamente em um servidor para criar uma conta de usuário ou um grupo, você pode usar o console do Fleet Manager para realizar as mesmas tarefas.

Para criar uma conta de usuário do sistema operacional usando o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o botão ao lado do nó gerenciado no qual você deseja criar um novo usuário.
4. Escolha Exibir detalhes.
5. Escolha Ferramentas, usuários e grupos.
6. Selecione a guia Users (Usuários) e, em seguida, Add users (Adicionar usuários).
7. Insira um valor para o Nome do novo usuário.
8. (Recomendado) Marque a caixa de seleção ao lado de Set password (Definir senha). Você será solicitado a fornecer uma senha para o novo usuário no final do procedimento.
9. Selecione Create user (Criar usuário). Se você marcou a caixa de seleção para criar uma senha para o novo usuário, você será solicitado a inserir um valor para a senha e selecionar Done (Concluído). Se a senha especificada não atender aos requisitos definidos pelas políticas locais ou de domínio do nó gerenciado, um erro será retornado.

Para criar um grupo do sistema operacional com o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o botão ao lado do nó gerenciado no qual você deseja criar um grupo.

4. Escolha Exibir detalhes.
5. Escolha Ferramentas, usuários e grupos.
6. Escolha a guia Groups (Grupos) e escolha Create group (Criar grupo).
7. Insira um valor para o Nome do novo grupo.
8. (Opcional) Insira um valor para a Descrição do novo grupo.
9. (Opcional) Selecione os usuários a serem adicionados aos Membros do grupo para o novo grupo.
10. Selecione Create group (Criar grupo).

Atualizar a associação do usuário ou grupo com o Fleet Manager

Em vez de fazer login diretamente em um servidor para atualizar uma conta de usuário ou um grupo, você pode usar o console do Fleet Manager para realizar as mesmas tarefas.

Para adicionar uma conta de usuário do SO a um novo grupo com o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o botão ao lado do nó gerenciado onde a conta de usuário existir e a qual você deseja atualizar.
4. Escolha Exibir detalhes.
5. Escolha Ferramentas, usuários e grupos.
6. Escolha a guia Users.
7. Selecione o botão ao lado do usuário que você quiser atualizar.
8. Escolha Ações, Adicionar usuário ao grupo.
9. Selecione o grupo ao qual você deseja adicionar o usuário em Add to group (Adicionar ao grupo).
10. Selecione Add user to group (Adicionar usuário ao grupo).

Para editar a associação de um grupo de sistema operacional usando o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o botão ao lado do nó gerenciado onde o grupo existir e o qual você deseja atualizar.

4. Escolha Exibir detalhes.
5. Escolha Ferramentas, usuários e grupos.
6. Selecione a guia Grupos.
7. Selecione o botão ao lado do grupo que você quiser atualizar.
8. Escolha Ações, Modificar grupo.
9. Selecione os usuários que você deseja adicionar ou remover em Group members (Membros do grupo).
10. Selecione Modify group (Modificar grupo).

Excluir um usuário ou grupo do sistema operacional usando o Fleet Manager

Em vez de fazer login diretamente em um servidor para excluir uma conta de usuário ou um grupo, você pode usar o console do Fleet Manager para realizar as mesmas tarefas.

Para excluir uma conta de usuário do sistema operacional usando o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o botão ao lado do nó gerenciado onde a conta de usuário existir e a qual você deseja excluir.
4. Escolha Exibir detalhes.
5. Escolha Usuários e grupos.
6. Escolha a guia Users.
7. Selecione o botão ao lado do usuário que você quiser excluir.
8. Escolha Ações, Excluir usuário local.

Para excluir um grupo do sistema operacional usando o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o botão ao lado do nó gerenciado onde o grupo existir e o qual você deseja excluir.
4. Escolha Exibir detalhes.
5. Escolha Ferramentas, usuários e grupos.

6. Selecione a guia Groups (Grupos).
7. Selecione o botão ao lado do grupo que você quiser atualizar.
8. Escolha Ações, Excluir grupo local.

Gerenciar o registro do Windows em nós gerenciados

Você pode usar o Fleet Manager, uma ferramenta do AWS Systems Manager, para gerenciar o Registro em seus nós gerenciados do Windows Server. No console do Fleet Manager, você pode criar, copiar, atualizar e excluir entradas e valores do registro.

> **Important**
>
> Recomendamos criar um backup do registro ou tirar um snapshot do volume raiz do Amazon Elastic Block Store (Amazon EBS) anexado ao nó gerenciado, antes de modificar o registro. Problemas sérios podem ocorrer se você modificar o registro incorretamente. Esses problemas podem exigir que você reinstale o sistema operacional ou restaure o volume raiz do nó com um snapshot. A AWS não garante que esses problemas possam ser resolvidos. Modifique o registro por sua conta e risco. Você é responsável por todas as alterações no Registro e por garantir que tenha backups.

Criar uma entrada ou do registro do Windows

Para criar uma chave do registro do Windows com o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o botão ao lado do nó gerenciado no qual você deseja criar uma chave de registro.
4. Escolha Exibir detalhes.
5. Escolha Ferramentas, Registro do Windows.
6. Selecione o hive em que você deseja criar uma nova chave do registro escolhendo Registry name (Nome do registro).
7. Escolha Criar chave do Registro.
8. Selecione o botão ao lado da entrada de registro na qual você quer criar uma nova chave.
9. Selecione Create registry key (Criar chave de registro).

10. Insira um valor para o Nome da nova chave do registro e selecione Submit (Enviar).

Para criar uma entrada do registro do Windows com o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o botão ao lado da instância na qual você quer criar uma entrada de registro.
4. Escolha Exibir detalhes.
5. Escolha Ferramentas, Registro do Windows.
6. Selecione o hive e a chave de registro subsequente em que você deseja criar uma nova entrada do registro, escolhendo Registry name (Nome do registro).
7. Escolha Criar, Criar entrada do Registro.
8. Insira um valor para o Nome da nova entrada de registro.
9. Selecione o Tipo de valor que você quer criar para a entrada do registro. Para obter mais informações sobre os tipos de valor do registro, consulte Tipos de valor do registro.
10. Insira um valor para o Valor da nova entrada de registro.

Atualize uma entrada do registro do Windows

Para atualizar uma entrada do registro do Windows com o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o botão ao lado do nó gerenciado no qual você deseja atualizar uma entrada de registro.
4. Escolha Exibir detalhes.
5. Escolha Ferramentas, Registro do Windows.
6. Selecione o hive e a chave de registro subsequente que você quiser atualizar, escolhendo Registry name (Nome do registro).
7. Selecione o botão ao lado da entrada de registro que você quiser atualizar.
8. Escolha Ações, Atualizar entrada do Registro.
9. Insira o novo valor para o Valor da entrada do registro.
10. Selecione Atualizar.

Exclua uma chave ou entrada do registro do Windows

Para excluir uma chave do registro do Windows com o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o botão ao lado do nó gerenciado no qual você deseja excluir uma chave de registro.
4. Escolha Ferramentas, Registro do Windows.
5. Selecione o hive e a chave de registro subsequente que você quiser excluir, escolhendo Registry name (Nome do registro).
6. Selecione o botão ao lado da chave de registro que você quiser excluir.
7. Escolha Ações, Excluir chave do Registro.

Para excluir uma entrada do registro do Windows com o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o botão ao lado do nó gerenciado no qual você deseja excluir uma entrada de registro.
4. Escolha Exibir detalhes.
5. Escolha Ferramentas, Registro do Windows.
6. Selecione o hive e a chave de registro subsequente contendo a entrada que você quiser excluir, escolhendo Registry name (Nome do registro).
7. Selecione o botão ao lado da entrada de registro que você quiser excluir.
8. Escolha Ações, Excluir entrada do Registro.

Gerenciar instâncias do EC2 automaticamente com a Configuração de gerenciamento de hosts padrão

A opção Configuração de gerenciamento de hosts padrão permite que o AWS Systems Manager gerencie as instâncias do Amazon EC2 automaticamente na forma de instâncias gerenciadas. Uma instância gerenciada é uma instância do EC2 que foi configurada para uso com o Systems Manager.

Os benefícios de gerenciar instâncias com o Systems Manager incluem o seguinte:

- Conecte-se às suas instâncias do EC2 com segurança usando o Session Manager.
- Realizar verificações de patches automatizadas usando o Patch Manager.
- Visualizar informações detalhadas sobre suas instâncias usando o Inventário do Systems Manager.
- Acompanhar e gerenciar instâncias usando o Fleet Manager.
- Mantenha o SSM Agent atualizado automaticamente.

Fleet Manager, Inventory, Patch Manager e Session Manager são ferramentas do Systems Manager.

Usando a configuração padrão de gerenciamento de host, é possível gerenciar instâncias do EC2 sem precisar criar manualmente um perfil de instância do AWS Identity and Access Management (IAM). Em vez disso, a Configuração de gerenciamento de hosts padrão cria e aplica um perfil do IAM padrão para garantir que o Systems Manager tenha permissões para gerenciar todas as instâncias na Conta da AWS e na Região da AWS em que ela está ativada.

Se as permissões fornecidas não forem suficientes para seu caso de uso, também é possível adicionar políticas ao perfil do IAM padrão criado pela configuração de gerenciamento de host padrão. Como alternativa, se você não precisar de permissões para todas as funcionalidades fornecidas pelo perfil do IAM padrão, poderá criar seu próprio perfil e as políticas personalizadas. Quaisquer alterações realizadas no perfil do IAM que você escolher para a configuração de gerenciamento de host padrão se aplicarão a todas as instâncias do Amazon EC2 gerenciadas na região e na conta.

Para obter mais informações sobre a política usada pela configuração de gerenciamento de host padrão, consulte Política gerenciada pela AWS: AmazonSSMManagedEC2InstanceDefaultPolicy.

Implemente o acesso de privilégio mínimo

Os procedimentos neste tópico devem ser executados somente por administradores. Portanto, é recomendável implementar o acesso com privilégio mínimo para evitar que usuários não administrativos configurem ou modifiquem a configuração de gerenciamento do host padrão. Para ver exemplos de políticas que restringem o acesso à configuração de gerenciamento do host padrão, consulte Exemplos de política de privilégio mínimo para configuração de gerenciamento do host padrão, mais adiante neste tópico.

> **Important**
>
> As instâncias registradas usando a Configuração de gerenciamento de hosts padrão armazenam informações de registro localmente nos diretórios `var/lib/amazon/ssm` ou `C:\ProgramData\Amazon`. A remoção desses diretórios ou de seus arquivos impedirá que a instância adquira as credenciais necessárias para se conectar ao Systems Manager usando a Configuração Padrão de Gerenciamento de Host. Nesses casos, você deve usar um perfil de instância do IAM para fornecer as permissões necessárias à sua instância, ou então recriar a instância.

Tópicos

- Pré-requisitos
- Ativar a opção Configuração de gerenciamento de hosts padrão
- Desativar a opção Configuração de gerenciamento de hosts padrão
- Exemplos de política de privilégio mínimo para configuração de gerenciamento do host padrão

Pré-requisitos

Para usar a Configuração de gerenciamento de hosts padrão na Região da AWS e na Conta da AWS em que a opção é ativada, é necessário cumprir os requisitos a seguir.

- A instância a ser gerenciada deve usar o Instance Metadata Service Version 2 (IMDSv2).

  A configuração de gerenciamento de host padrão não oferece suporte à versão 1 do serviço de metadados da instância. Para obter informações sobre como fazer a transição para IMDSv2, consulte Transição para usar o Serviço de metadados da instância versão 2 no Guia do usuário do Amazon EC2
- O SSM Agent versão 3.2.582.0 ou posterior deve estar instalado na instância a ser usada.

  Para obter informações sobre como verificar a versão do SSM Agent instalada em sua instância, consulte Verificar o número de versão do SSM Agent.

  Para obter informações sobre como atualizar o SSM Agent, consulte Atualizar automaticamente o SSM Agent.
- Como administrador que executa as tarefas deste tópico, você deve ter permissões para as operações de API GetServiceSetting, ResetServiceSetting e UpdateServiceSetting. Além

disso, você deve ter permissões para a permissão `iam:PassRole` para o perfil do IAM `AWSSystemsManagerDefaultEC2InstanceManagementRole`. Veja aqui um exemplo de política que concede estas permissões. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetServiceSetting",
                "ssm:ResetServiceSetting",
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:servicesetting/ssm/
managed-instance/default-ec2-instance-management-role"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/service-role/
AWSSystemsManagerDefaultEC2InstanceManagementRole",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ssm.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

- Se um perfil de instância IAM já estiver anexado a uma instância EC2 que deve ser gerenciada usando o Systems Manager, você deve remover quaisquer permissões que permitam a operação `ssm:UpdateInstanceInformation`. O SSM Agent tentará usar as permissões do perfil de instância antes de usar as permissões de Configuração Padrão de Gerenciamento de Host. Se

você permitir a operação `ssm:UpdateInstanceInformation` em seu próprio perfil de instância do IAM, a instância não usará as permissões de configuração de gerenciamento do host padrão.

Ativar a opção Configuração de gerenciamento de hosts padrão

É possível ativar a Configuração de gerenciamento de hosts padrão no console do Fleet Manager ou via AWS Command Line Interface ou AWS Tools for Windows PowerShell.

É necessário ativar a Configuração de gerenciamento de hosts padrão individualmente em cada região em que você deseja que suas instâncias do Amazon EC2 sejam gerenciadas por essa opção.

Após a ativação da Configuração de gerenciamento de hosts padrão, até 30 minutos poderão ser necessários para que as instâncias usem as credenciais do perfil escolhido na etapa 5 do procedimento a seguir.

Para ativar a configuração de gerenciamento do host padrão (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Escolha Gerenciamento de contas, Definir configuração padrão de gerenciamento de host.
4. Ative Habilitar a configuração de gerenciamento de host padrão.
5. Escolha o perfil do AWS Identity and Access Management (IAM) usado para habilitar as ferramentas do Systems Manager para suas instâncias. Recomendamos usar o perfil padrão fornecido pela configuração de gerenciamento de host padrão. Ele contém o conjunto mínimo de permissões necessárias para gerenciar as instâncias do Amazon EC2 usando o Systems Manager. Se você preferir usar um perfil personalizado, a política de confiança do perfil deve permitir que o Systems Manager seja uma entidade confiável.
6. Escolha Configurar para concluir a configuração.

Para ativar a configuração de gerenciamento do host padrão (linha de comando)

1. Crie um arquivo JSON em sua máquina local contendo a política de relacionamento de confiança a seguir.

   JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"",
            "Effect":"Allow",
            "Principal":{
                "Service":"ssm.amazonaws.com"
            },
            "Action":"sts:AssumeRole"
        }
    ]
}
```

2. Abra a AWS CLI ou as ferramentas para Windows PowerShell e execute um dos comandos a seguir, de acordo com o tipo de sistema operacional de sua máquina local, para criar um perfil de serviço em sua conta. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   Linux & macOS

   ```
   aws iam create-role \
   --role-name AWSSystemsManagerDefaultEC2InstanceManagementRole \
   --path /service-role/ \
   --assume-role-policy-document file://trust-policy.json
   ```

   Windows

   ```
   aws iam create-role ^
   --role-name AWSSystemsManagerDefaultEC2InstanceManagementRole ^
   --path /service-role/ ^
   --assume-role-policy-document file://trust-policy.json
   ```

   PowerShell

   ```
   New-IAMRole `
   -RoleName "AWSSystemsManagerDefaultEC2InstanceManagementRole" `
   -Path "/service-role/" `
   -AssumeRolePolicyDocument "file://trust-policy.json"
   ```

3. Execute o comando a seguir para anexar a política gerenciada `AmazonSSMManagedEC2InstanceDefaultPolicy` ao seu perfil recém-criado. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   Linux & macOS

   ```
   aws iam attach-role-policy \
   --policy-arn arn:aws:iam::aws:policy/AmazonSSMManagedEC2InstanceDefaultPolicy \
   --role-name AWSSystemsManagerDefaultEC2InstanceManagementRole
   ```

   Windows

   ```
   aws iam attach-role-policy ^
   --policy-arn arn:aws:iam::aws:policy/AmazonSSMManagedEC2InstanceDefaultPolicy ^
   --role-name AWSSystemsManagerDefaultEC2InstanceManagementRole
   ```

   PowerShell

   ```
   Register-IAMRolePolicy `
   -PolicyArn "arn:aws:iam::aws:policy/AmazonSSMManagedEC2InstanceDefaultPolicy" `
   -RoleName "AWSSystemsManagerDefaultEC2InstanceManagementRole"
   ```

4. Abra a AWS CLI ou as ferramentas para Windows PowerShell e execute o comando a seguir. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   Linux & macOS

   ```
   aws ssm update-service-setting \
   --setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/managed-instance/
   default-ec2-instance-management-role \
   --setting-value service-role/AWSSystemsManagerDefaultEC2InstanceManagementRole
   ```

   Windows

   ```
   aws ssm update-service-setting ^
   --setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/managed-instance/
   default-ec2-instance-management-role ^
   --setting-value service-role/AWSSystemsManagerDefaultEC2InstanceManagementRole
   ```

PowerShell

```
Update-SSMServiceSetting `
-SettingId "arn:aws:ssm:region:account-id:servicesetting/ssm/managed-instance/
default-ec2-instance-management-role" `
-SettingValue "service-role/AWSSystemsManagerDefaultEC2InstanceManagementRole"
```

Não haverá saída se o comando for bem-sucedido.

5. Execute o comando a seguir para visualizar as configurações de serviço atuais para a configuração de gerenciamento de host padrão na Conta da AWS e na Região da AWS atuais.

Linux & macOS

```
aws ssm get-service-setting \
--setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/managed-instance/
default-ec2-instance-management-role
```

Windows

```
aws ssm get-service-setting ^
--setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/managed-instance/
default-ec2-instance-management-role
```

PowerShell

```
Get-SSMServiceSetting `
-SettingId "arn:aws:ssm:region:account-id:servicesetting/ssm/managed-instance/
default-ec2-instance-management-role"
```

O comando retorna informações como as seguintes.

```
{
    "ServiceSetting": {
        "SettingId": "/ssm/managed-instance/default-ec2-instance-management-role",
        "SettingValue": "service-role/
AWSSystemsManagerDefaultEC2InstanceManagementRole",
        "LastModifiedDate": "2022-11-28T08:21:03.576000-08:00",
        "LastModifiedUser": "System",
```

```
        "ARN": "arn:aws:ssm:us-east-2:-123456789012:servicesetting/ssm/managed-
instance/default-ec2-instance-management-role",
        "Status": "Custom"
    }
}
```

Desativar a opção Configuração de gerenciamento de hosts padrão

É possível desativar a Configuração de gerenciamento de hosts padrão no console do Fleet Manager ou via AWS Command Line Interface ou AWS Tools for Windows PowerShell.

É necessário desativar a definição de Configuração de gerenciamento de hosts padrão individualmente em cada região em que você não deseja mais que suas instâncias do Amazon EC2 sejam gerenciadas por essa opção. Desativar em uma região não desativará em todas as regiões.

Se você desativar a configuração de gerenciamento do host padrão e não tiver anexado um perfil de instância às instâncias do Amazon EC2 que permita acesso ao Systems Manager, elas não serão mais gerenciadas pelo Systems Manager.

Para desativar a configuração de gerenciamento do host padrão (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Escolha Gerenciamento de contas, Configuração padrão de gerenciamento de host.
4. Desative Habilitar a configuração de gerenciamento de host padrão.
5. Escolha Configurar para desabilitar a configuração de gerenciamento de host padrão.

Para desativar a configuração de gerenciamento do host padrão (linha de comando)

- Abra a AWS CLI ou as ferramentas para Windows PowerShell e execute o comando a seguir. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

  Linux & macOS

  ```
  aws ssm reset-service-setting \
  --setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/managed-instance/
  default-ec2-instance-management-role
  ```

Windows

```
aws ssm reset-service-setting ^
--setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/managed-instance/
default-ec2-instance-management-role
```

PowerShell

```
Reset-SSMServiceSetting `
-SettingId "arn:aws:ssm:region:account-id:servicesetting/ssm/managed-instance/
default-ec2-instance-management-role"
```

Exemplos de política de privilégio mínimo para configuração de gerenciamento do host padrão

Os exemplos de políticas a seguir demonstram como impedir que membros de sua organização façam alterações na configuração de gerenciamento do host padrão de sua conta.

Política de controle de serviço para o AWS Organizations

A política a seguir demonstra como impedir que membros não administrativos de sua AWS Organizations atualizem sua configuração de gerenciamento do host padrão. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Deny",
            "Action": [
                "ssm:UpdateServiceSetting",
                "ssm:ResetServiceSetting"
            ],
            "Resource": "arn:aws:ssm:*:*:servicesetting/ssm/managed-instance/
default-ec2-instance-management-role",
            "Condition": {
                "StringNotEqualsIgnoreCase": {
                    "aws:PrincipalTag/job-function": [
                        "administrator"
                    ]
```

```
                }
            }
        },
        {
            "Effect": "Deny",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::*:role/service-role/
AWSSystemsManagerDefaultEC2InstanceManagementRole",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ssm.amazonaws.com"
                },
                "StringNotEqualsIgnoreCase": {
                    "aws:PrincipalTag/job-function": [
                        "administrator"
                    ]
                }
            }
        },
        {
            "Effect": "Deny",
            "Resource": "arn:aws:iam::*:role/service-role/
AWSSystemsManagerDefaultEC2InstanceManagementRole",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:DeleteRole"
            ],
            "Condition": {
                "StringNotEqualsIgnoreCase": {
                    "aws:PrincipalTag/job-function": [
                        "administrator"
                    ]
                }
            }
        }
    ]
}
```

Política para entidades principais do IAM

A política a seguir demonstra como impedir que grupos, perfis e usuários do IAM de sua AWS Organizations atualizem sua configuração de gerenciamento do host padrão. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "ssm:UpdateServiceSetting",
                "ssm:ResetServiceSetting"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:servicesetting/ssm/
managed-instance/default-ec2-instance-management-role"
        },
        {
            "Effect": "Deny",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:DeleteRole",
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/service-role/
AWSSystemsManagerDefaultEC2InstanceManagementRole"
        }
    ]
}
```

Conectar a uma instância gerenciada pelo Windows Server usando o Remote Desktop

Você pode usar o Fleet Manager, uma ferramenta do AWS Systems Manager, para se conectar às instâncias Windows Server do Amazon Elastic Compute Cloud (Amazon EC2) usando o Remote Desktop Protocol (RDP). Fleet Manager O Remote Desktop, baseado no Amazon DCV, fornece conectividade segura às suas instâncias do Windows Server diretamente do console do Systems Manager. É possível ter até quatro conexões simultâneas em uma única janela de navegador.

A API Fleet Manager da área de trabalho remota é denominada AWS Systems Manager GUI Connect. Para obter informações sobre como usar a API Systems Manager GUI Connect, consulte a Referência da API AWS Systems Manager GUI Connect.

Atualmente, apenas é possível usar a Área de Trabalho Remota com instâncias que estejam executando o Windows Server 2012 RTM ou versões posteriores. Atualmente, a Área de Trabalho Remoto oferece suporte somente a entradas no idioma inglês.

O Fleet Manager Remote Desktop é um serviço exclusivo para console e não oferece suporte a conexões de linha de comando com suas instâncias gerenciadas. Para se conectar a uma instância gerenciada pelo Windows Server por meio de um shell, é possível usar o Session Manager, outra ferramenta do AWS Systems Manager. Para obter mais informações, consulte AWS Systems Manager Session Manager.

> **Note**
>
> A duração de uma conexão RDP não é determinada pela duração de suas credenciais (IAM) AWS Identity and Access Management. Em vez disso, a conexão persiste até que a duração máxima da conexão ou o limite de tempo de inatividade seja atingido, o que ocorrer primeiro. Para obter mais informações, consulte Duração e simultaneidade da conexão remota.

Para obter informações sobre como configurar as permissões do AWS Identity and Access Management (IAM) para permitir que suas instâncias interajam com o Systems Manager, consulte Configurar permissões de instâncias para o Systems Manager.

Tópicos

- Configuração de seu ambiente
- Configurar permissões do IAM para a Área de Trabalho Remota
- Autenticar conexões da Área de Trabalho Remota
- Duração e simultaneidade da conexão remota
- Conectar-se a um nó gerenciado usando a Área de Trabalho Remota
- Visualizar informações sobre conexões atuais e concluídas

Configuração de seu ambiente

Antes de usar a Área de Trabalho Remota, verifique se o ambiente atende aos seguintes requisitos:

- Configuração de nós gerenciados

  Confirme se suas instâncias do Amazon EC2 estão configuradas como nós gerenciados no Systems Manager.

- Versão mínima do SSM Agent

  Verifique se os nós estão executando o SSM Agent versão 3.0.222.0 ou posterior. Para obter informações sobre como verificar qual versão do agente está sendo executada em um nó, consulte Verificar o número de versão do SSM Agent. Para obter informações sobre como instalar ou atualizar o SSM Agent, consulte Trabalhar com o SSM Agent.

- Configuração da porta RDP

  Para aceitar conexões remotas, o serviço do Remote Desktop Services em seus nós do Windows Server deve usar a porta RDP padrão 3389. Essa é a configuração padrão em Amazon Machine Images (AMIs) fornecida pela AWS. Não é necessário abrir explicitamente nenhuma porta de entrada para usar a Área de Trabalho Remota.

- Versão do módulo do PSReadLine para funcionalidade do teclado

  Para garantir que o teclado funcione corretamente no PowerShell, verifique se os nós que executam o Windows Server 2022 têm a versão 2.2.2 ou posterior do módulo do PSReadLine instalada. Se estiverem executando uma versão mais antiga, você poderá instalar a versão necessária usando os comandos a seguir.

  ```
  Install-PackageProvider -Name NuGet -MinimumVersion 2.8.5.201 -Force
  ```

  Depois que o provedor de pacotes NuGet estiver instalado, execute o comando a seguir.

  ```
  Install-Module `
   -Name PSReadLine `
   -Repository PSGallery `
   -MinimumVersion 2.2.2 -Force
  ```

- Configuração do Gerenciador de Sessões

  Antes de usar a Área de Trabalho Remota, é necessário atender aos pré-requisitos para a configuração do Gerenciador de Sessões. Quando você se conecta a uma instância usando a Área de Trabalho Remota, aplicam-se todas as preferências de sessão definidas para sua Conta da AWS e Região da AWS. Para obter mais informações, consulte Configurar o Session Manager.

> **Note**
>
> Se você registrar em log as atividades do Gerenciador de Sessões usando o Amazon Simple Storage Service (Amazon S3), as conexões da Área de Trabalho Remota gerarão o erro a seguir em `bucket_name/Port/stderr`. Esse erro é um comportamento esperado e pode ser ignorado com segurança.
>
> ```
> Setting up data channel with id SESSION_ID failed: failed to create websocket
>  for datachannel with error: CreateDataChannel failed with no output or
>  error: createDataChannel request failed: unexpected response from the service
>  <BadRequest>
> <ClientErrorMessage>Session is already terminated</ClientErrorMessage>
> </BadRequest>
> ```

Configurar permissões do IAM para a Área de Trabalho Remota

Além das permissões do IAM necessárias para o Systems Manager e o Session Manager, o usuário ou perfil que você utilizar deverá ter permissões para iniciar conexões.

Permissões para iniciar conexões

Para fazer conexões RDP com instâncias do EC2 no console, as seguintes permissões são necessárias:

- `ssm-guiconnect:CancelConnection`
- `ssm-guiconnect:GetConnection`
- `ssm-guiconnect:StartConnection`

Permissões para listar conexões

Para visualizar listas de conexões no console, a seguinte permissão é necessária:

`ssm-guiconnect:ListConnections`

Veja a seguir exemplos de políticas do IAM que você pode associar a um usuário ou perfil para permitir diferentes tipos de interação com a Área de Trabalho Remota. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

Política padrão para conexão com instâncias do EC2

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EC2",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:GetPasswordData"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SSM",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeInstanceProperties",
                "ssm:GetCommandInvocation",
                "ssm:GetInventorySchema"
            ],
            "Resource": "*"
        },
        {
            "Sid": "TerminateSession",
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/aws:ssmmessages:session-id": [
                        "${aws:userid}"
                    ]
                }
            }
        },
        {
            "Sid": "SSMStartSession",
```

```
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ec2:*:111122223333:instance/*",
                "arn:aws:ssm:*:111122223333:managed-instance/*",
                "arn:aws:ssm:*::document/AWS-StartPortForwardingSession"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": "ssm-guiconnect.amazonaws.com"
                }
            }
        },
        {
            "Sid": "GuiConnect",
            "Effect": "Allow",
            "Action": [
                "ssm-guiconnect:CancelConnection",
                "ssm-guiconnect:GetConnection",
                "ssm-guiconnect:StartConnection",
                "ssm-guiconnect:ListConnections"
            ],
            "Resource": "*"
        }
    ]
}
```

Política para conectar-se a instâncias do EC2 com etiquetas específicas

> **Note**
>
> Na política do IAM a seguir, a seção `SSMStartSession` exige um nome do recurso da Amazon (ARN) para a ação `ssm:StartSession` Conforme mostrado, o ARN que você especifica não exige uma ID de Conta da AWS. Se você especificar um ID de conta, o Fleet Manager retornará uma `AccessDeniedException`.
>
> A seção `AccessTaggedInstances`, localizada na parte inferior da política de exemplo, também exige ARNs para `ssm:StartSession`. Para esses ARNs, é necessário especificar IDs de Conta da AWS.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EC2",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:GetPasswordData"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SSM",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeInstanceProperties",
                "ssm:GetCommandInvocation",
                "ssm:GetInventorySchema"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SSMStartSession",
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ssm:*::document/AWS-StartPortForwardingSession"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": "ssm-guiconnect.amazonaws.com"
                }
            }
        },
        {
            "Sid": "AccessTaggedInstances",
            "Effect": "Allow",
            "Action": [
```

```
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ec2:*:111122223333:instance/*",
                "arn:aws:ssm:*:111122223333:managed-instance/*"
            ],
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/tag key": [
                        "tag value"
                    ]
                }
            }
        },
        {
            "Sid": "GuiConnect",
            "Effect": "Allow",
            "Action": [
                "ssm-guiconnect:CancelConnection",
                "ssm-guiconnect:GetConnection",
                "ssm-guiconnect:StartConnection",
                "ssm-guiconnect:ListConnections"
            ],
            "Resource": "*"
        }
    ]
}
```

Política para os usuários do AWS IAM Identity Center se conectarem a instâncias do EC2

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SSO",
            "Effect": "Allow",
            "Action": [
                "sso:ListDirectoryAssociations*",
                "identitystore:DescribeUser"
            ],
```

```
            "Resource": "*"
        },
        {
            "Sid": "EC2",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:GetPasswordData"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SSM",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeInstanceInformation",
                "ssm:DescribeInstanceProperties",
                "ssm:GetCommandInvocation",
                "ssm:GetInventorySchema"
            ],
            "Resource": "*"
        },
        {
            "Sid": "TerminateSession",
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/aws:ssmmessages:session-id": [
                        "${aws:userName}"
                    ]
                }
            }
        },
        {
            "Sid": "SSMStartSession",
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
```

```
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:ssm:*:*:managed-instance/*",
                "arn:aws:ssm:*:*:document/AWS-StartPortForwardingSession"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": "ssm-guiconnect.amazonaws.com"
                }
            }
        },
        {
            "Sid": "SSMSendCommand",
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:ssm:*:*:managed-instance/*",
                "arn:aws:ssm:*:*:document/AWSSSO-CreateSSOUser"
            ]
        },
        {
            "Sid": "GuiConnect",
            "Effect": "Allow",
            "Action": [
                "ssm-guiconnect:CancelConnection",
                "ssm-guiconnect:GetConnection",
                "ssm-guiconnect:StartConnection",
                "ssm-guiconnect:ListConnections"
            ],
            "Resource": "*"
        }
    ]
}
```

Autenticar conexões da Área de Trabalho Remota

Ao estabelecer uma conexão remota, você pode se autenticar usando credenciais do Windows ou o par de chaves do Amazon EC2 (arquivo `.pem`) que está associado à instância. Para obter mais informações sobre pares de chaves, consulte Pares de chaves do Amazon EC2 e instâncias do Windows no Guia do usuário do Amazon EC2.

Como alternativa, se você estiver autenticado para o AWS Management Console usando AWS IAM Identity Center, você pode se conectar às instâncias sem fornecer credenciais adicionais. Para ver um exemplo de uma política para permitir a autenticação de conexão remota usando o Centro de Identidade do IAM, consulte Configurar permissões do IAM para a Área de Trabalho Remota.

Antes de começar

Observe as seguintes condições para utilizar a autenticação do IAM Identity Center antes de iniciar a conexão usando o Remote Desktop.

- A Área de Trabalho Remota é compatível com a autenticação do Centro de Identidade do IAM para nós na mesma Região da AWS em que você habilitou o Centro de Identidade do IAM.
- A Área de Trabalho Remota é compatível com nomes de usuário do Centro de Identidade do IAM de até 16 caracteres.
- A Área de Trabalho Remota é compatível com nomes de usuário do Centro de Identidade do IAM que consistem em caracteres alfanuméricos e nos seguintes caracteres especiais: `. - _`

> **Important**
>
> As conexões não terão êxito para nomes de usuário do Centro de Identidade do IAM que contenham estes caracteres: `+ = ,`
> O Centro de Identidade do IAM é compatível com esses caracteres nos nomes de usuário, mas as conexões RDP do Fleet Manager não são.
> Além disso, se um nome de usuário do IAM Identity Center contiver um ou mais símbolos `@`, o Fleet Manager desconsiderará o primeiro símbolo `@` e todos os caracteres que o seguem, independentemente de `@` introduzir ou não a parte do domínio de um endereço de e-mail. Por exemplo, para o nome de usuário do Centro de Identidade do IAM `diego_ramirez@example.com`, a parte `@example.com` é ignorada, e o nome do usuário Fleet Manager se torna `diego_ramirez`. Para `diego_r@mirez@example.com`, o Fleet Manager desconsidera `@mirez@example.com`, e o nome de usuário do Fleet Manager se torna `diego_r`.

- Quando uma conexão é autenticada usando o Centro de Identidade do IAM, a Área de Trabalho Remota cria um usuário local do Windows no grupo Administradores locais da instância. O usuário persiste após o término da conexão remota.
- A Área de Trabalho Remota não permite a autenticação do Centro de Identidade do IAM para nós que são controladores de domínio do Microsoft Active Directory.

- Embora a Área de Trabalho Remota permita usar a autenticação do Centro de Identidade do IAM para nós associados a um domínio do Active Directory, isso não é recomendável. Esse método de autenticação concede permissões administrativas aos usuários, o que poderá substituir as permissões mais restritivas concedidas pelo domínio.

Regiões com suporte para a autenticação do Centro de Identidade do IAM

As conexões Remote Desktop usando a autenticação do Centro de Identidade do IAM são compatíveis com o seguinte Regiões da AWS:

- Leste dos EUA (Ohio) (us-east-2)
- Leste dos EUA (Norte da Virgínia) (us-east-1)
- Oeste dos EUA (Norte da Califórnia) (us-west-1)
- Oeste dos EUA (Oregon) (us-west-2)
- África (Cidade do Cabo) (af-south-1)
- Ásia-Pacífico (Hong Kong) (ap-east-1)
- Ásia-Pacífico (Mumbai) (ap-south-1)
- Ásia Pacific (Tóquio) (ap-northeast-1)
- Ásia-Pacífico (Seul) (ap-northeast-2)
- Ásia-Pacífico (Osaka) (ap-northeast-3)
- Ásia-Pacífico (Singapura) (ap-southeast-1)
- Ásia-Pacífico (Sydney) (ap-southeast-2)
- Ásia-Pacífico (Jacarta) (ap-southeast-3)
- Canadá (Central) (ca-central-1)
- Europa (Frankfurt) (eu-central-1)
- UE (Estocolmo) (eu-north-1)
- Europa (Irlanda) (eu-west-1)
- Europa (Londres) (eu-west-2)
- Europa (Paris) (eu-west-3)
- Israel (Tel Aviv) (il-central-1)
- América do Sul (São Paulo) (sa-east-1)
- UE (Milão) (eu-south-1)

- Oriente Médio (Bahrein) (me-south-1)
- AWS GovCloud (Leste dos EUA) (us-gov-east-1)
- AWS GovCloud (Oeste dos EUA) (us-gov-wast-1)

Duração e simultaneidade da conexão remota

Estas condições se aplicam às conexões ativas da Área de Trabalho Remota:

- Duração da conexão

  Por padrão, uma conexão da Área de Trabalho Remota é desconectada após 60 minutos. Para evitar que a conexão seja desconectada, você pode escolher Renovar sessão antes de ser desconectada para redefinir o cronômetro de duração.
- Tempo limite da conexão

  Uma conexão da Área de Trabalho Remota se desconecta depois de ficar ociosa por mais de dez minutos.
- Persistência de conexão

  Depois de se conectar a um Windows Server usando a Área de Trabalho Remota, a conexão persiste até que a duração máxima da conexão (60 minutos) ou o limite de tempo de inatividade (10 minutos) seja atingido. A duração de uma conexão não é determinada pela duração de suas credenciais (IAM) AWS Identity and Access Management. A conexão persiste após a expiração das credenciais do IAM, se os limites de duração da conexão não forem atingidos. Ao usar a Área de Trabalho Remota, você deve encerrar sua conexão após a expiração das credenciais do IAM saindo da página do navegador.
- Conexões simultâneas

  Por padrão, você pode ter no máximo cinco conexões da Área de Trabalho Remota ativas ao mesmo tempo para a mesma Conta da AWS e Região da AWS. Para solicitar o aumento da cota de serviço para até 25 conexões simultâneas, consulte Requesting a Quota Increase no Guia do usuário do Service Quotas.

> **Note**
>
> A licença padrão para o Windows Server permite duas conexões RDP simultâneas. Para oferecer suporte a mais conexões, você deve comprar licenças de acesso para cliente (CALs) adicionais da Microsoft ou licenças dos Serviços de Área de Trabalho Remota

da Microsoft na AWS. Para obter mais informações sobre licenciamento suplementar, consulte os seguintes tópicos:

- Licenças de acesso para clientes e licenças de gerenciamento no site da Microsoft
- Use assinaturas baseadas no usuário do License Manager para produtos de software compatíveis no Guia do usuário do License Manager

Conectar-se a um nó gerenciado usando a Área de Trabalho Remota

Suporte a copiar/colar texto em um navegador

Usando os navegadores Google Chrome e Microsoft Edge, você pode copiar e colar texto de um nó gerenciado em sua máquina local e de sua máquina local em um nó gerenciado ao qual você está conectado.

Usando o navegador Mozilla Firefox, você pode copiar e colar texto de um nó gerenciado somente na sua máquina local. Não há suporte à cópia da máquina local para o nó gerenciado.

Para conectar-se a um nó gerenciado usando a Área de Trabalho Remota do Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Escolha o nó ao qual deseja se conectar. Marque a caixa de seleção ou o nome do nó.
4. No menu Ações do nó, selecione Conectar Área de Trabalho Remota.
5. Escolha o seu Tipo de autenticação preferido. Se você escolher Credenciais do usuário, insira o nome de usuário e a senha de uma conta de usuário do Windows no nó ao qual está se conectando. Se você escolher Par de chaves, poderá fornecer autenticação usando um destes métodos:

   a. Escolha Procurar máquina local para selecionar a chave PEM associada à instância no sistema de arquivos local.

      - ou -

   b. Escolha Colar conteúdo de par de chaves para copiar o conteúdo do arquivo PEM e colá-lo no campo fornecido.
6. Selecione Connect (Conectar-se).

7. Para escolher a resolução de tela de sua preferência, no menu Ações, escolha Resoluções e selecione uma destas opções:

   - Adaptar automaticamente
   - 1920 x 1080
   - 1400 x 900
   - 1366 x 768
   - 800 x 600

   A opção Adaptar automaticamente define a resolução com base no tamanho da tela detectada.

Visualizar informações sobre conexões atuais e concluídas

É possível usar a seção Fleet Manager do console do Systems Manager para visualizar informações sobre conexões RDP que foram feitas em sua conta. Usando um conjunto de filtros, é possível limitar a lista de conexões exibidas a um intervalo de tempo, a uma instância específica, ao usuário que fez as conexões e às conexões de um status específico. O console também fornece guias que mostram informações sobre todas as conexões ativas no momento e todas as conexões passadas.

Para visualizar informações sobre conexões atuais e concluídas

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Escolha Gerenciamento de contas, Conectar com Remote Desktop.
4. Escolha uma das seguintes guias:

   - Conexões ativas
   - Histórico de conexão

5. Para restringir ainda mais a lista de resultados de conexão exibidos, especifique um ou mais filtros na caixa de pesquisa
   ( ).
   Você também pode inserir um termo de pesquisa de texto livre.

Gerenciar volumes do Amazon EBS em instâncias gerenciadas

O Amazon Elastic Block Store (Amazon EBS) oferece volumes de armazenamento ao nível do bloco para uso com instâncias do Amazon Elastic Compute Cloud (EC2). Os volumes do EBS se comportam como dispositivos de bloco brutos e não formatados. É possível montar esses volumes como dispositivos em suas instâncias.

É possível usar o Fleet Manager, uma ferramenta do AWS Systems Manager, para gerenciar volumes do Amazon EBS em suas instâncias gerenciadas. Por exemplo, é possível inicializar um volume do EBS, formatar uma partição e montar o volume para disponibilizá-lo para uso.

> **Note**
>
> Atualmente, o Fleet Manager oferece suporte ao gerenciamento de volumes do Amazon EBS somente para instâncias Windows Server.

Visualizar detalhes de um volume do EBS

Para visualizar detalhes de um volume do EBS com o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione o botão ao lado da instância gerenciada cujos detalhes do volume do EBS você deseja visualizar.
4. Escolha Exibir detalhes.
5. Escolha Ferramentas, volumes do EBS.
6. Para ver os detalhes de um volume do EBS, escolha sua ID na coluna ID do volume.

Inicializar e formatar um volume do EBS

Para inicializar e formatar um volume do EBS com o Fleet Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.

3. Selecione o botão ao lado da instância gerenciada para a qual você deseja inicializar, formatar e montar um volume do EBS. Apenas será possível inicializar um volume do EBS se o disco estiver vazio.
4. Escolha Exibir detalhes.
5. No menu Ferramentas, escolha Volumes do EBS.
6. Selecione o botão ao lado do volume do EBS que você deseja inicializar e formatar.
7. Escolha Inicializar e formatar.
8. Em Estilo de partição, escolha o estilo de partição desejado para o volume do EBS.
9. (Opcional) Escolha uma Letra de unidade para a partição.
10. (Opcional) Insira um Nome de partição para identificar a partição.
11. Escolha o Sistema de arquivos a ser usado para organizar arquivos e dados armazenados na partição.
12. Escolha Confirmar para disponibilizar o volume do EBS para uso. Não será possível alterar a configuração da partição no AWS Management Console após a confirmação. Porém, você pode usar SSH ou RDP para fazer login na instância e alterar essa configuração.

Como acessar o portal da base de conhecimento do Red Hat

É possível usar o Fleet Manager, uma ferramenta do AWS Systems Manager, para acessar o portal da base de conhecimento se você for um cliente da Red Hat. Você é considerado um cliente da Red Hat se você executar instâncias Red Hat Enterprise Linux (RHEL) ou usar os serviços do RHEL no AWS. O portal da base de conhecimento inclui binários, fóruns de compartilhamento de conhecimento e discussões para suporte da comunidade, que estão disponíveis apenas para clientes licenciados pela Red Hat.

Além das permissões do AWS Identity and Access Management (IAM) necessárias para o Systems Manager e o Fleet Manager, o usuário ou o perfil usados para acessar o console deverá permitir que a ação `rhelkb:GetRhelURL` acesse o portal da base de conhecimento.

Para acessar o Red Hat Knowledgebase Portal

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Selecione a instância RHEL que você deseja usar para se conectar ao Red Hat Knowledgebase Portal.

4. Escolha Gerenciamento de contas, Acessar Red Hat Knowledgebase para abrir a página da base de conhecimento da Red Hat.

Se você usar RHEL na AWS para executar workloads do RHEL totalmente compatíveis, você também poderá acessar a base de conhecimento da Red Hat por meio do site da Red Hat usando suas credenciais da AWS.

Solução de problemas de disponibilidade do nó gerenciado

Para várias ferramentas do AWS Systems Manager como Run Command, Distributor e Session Manager, você pode optar por selecionar manualmente os nós gerenciados nos quais deseja executar uma operação. Em casos como esses, depois de especificar que você deseja escolher os nós manualmente, o sistema exibirá uma lista de nós gerenciados nos quais você poderá executar a operação.

Este tópico fornece informações para ajudar você a identificar por que um nó gerenciado, confirmadamente em execução, não está incluído em suas listas de nós gerenciados no Systems Manager.

Para que um nó seja gerenciado pelo Systems Manager e disponível nas listas de nós gerenciados, ele deverá atender a três requisitos principais:

- O SSM Agent deve ser instalado e executado em um nó com um sistema operacional compatível.

> **Note**
>
> Algumas Amazon Machine Images (AMIs) gerenciadas pela AWS estão configuradas para iniciar instâncias com o SSM Agent pré-instalado. (Você também pode configurar uma AMI para pré-instalar o SSM Agent.) Para obter mais informações, consulte Encontrar AMIs com o SSM Agent pré-instalado.

- Para instâncias do Amazon Elastic Compute Cloud (Amazon EC2), você deve anexar um perfil do AWS Identity and Access Management (IAM) para a instância. O perfil da instância permite que ela se comunique com o serviço do Systems Manager. Se você não atribuir um perfil de instância à instância, registre-a usando uma ativação híbrida, o que não é uma situação comum.
- O SSM Agent deve ser capaz de se conectar a um endpoint do Systems Manager para se registrar no serviço. Posteriormente, o nó gerenciado deve estar disponível para o serviço e, para confirmar isso, o serviço envia um sinal a cada cinco minutos para verificar a integridade da instância.

- Depois que o status de um nó gerenciado `Connection Lost` durar pelo menos 30 dias, talvez o nó não esteja mais listado no Fleet Manager console. Para restaurá-lo na lista, o problema que causou a perda da conexão deve ser resolvido.

Depois de verificar se um nó gerenciado está em execução, será possível usar o comando a seguir para verificar se o SSM Agent foi registrado com êxito no serviço Systems Manager. Este comando não retorna resultados até que um registro bem-sucedido tenha ocorrido.

Linux & macOS

```
aws ssm describe-instance-associations-status \
    --instance-id instance-id
```

Windows

```
aws ssm describe-instance-associations-status ^
    --instance-id instance-id
```

PowerShell

```
Get-SSMInstanceAssociationsStatus `
    -InstanceId instance-id
```

Se o registro foi bem-sucedido e o nó gerenciado estiver agora disponível para operações do Systems Manager, o comando retornará resultados semelhantes aos mostrados a seguir.

```
{
    "InstanceAssociationStatusInfos": [
        {
            "AssociationId": "fa262de1-6150-4a90-8f53-d7eb5EXAMPLE",
            "Name": "AWS-GatherSoftwareInventory",
            "DocumentVersion": "1",
            "AssociationVersion": "1",
            "InstanceId": "i-02573cafcfEXAMPLE",
            "Status": "Pending",
            "DetailedStatus": "Associated"
        },
        {
            "AssociationId": "f9ec7a0f-6104-4273-8975-82e34EXAMPLE",
```

```
            "Name": "AWS-RunPatchBaseline",
            "DocumentVersion": "1",
            "AssociationVersion": "1",
            "InstanceId": "i-02573cafcfEXAMPLE",
            "Status": "Queued",
            "AssociationName": "SystemAssociationForScanningPatches"
        }
    ]
}
```

Se o registro ainda não foi concluído ou não foi bem-sucedido, o comando retornará resultados semelhantes aos seguintes:

```
{
    "InstanceAssociationStatusInfos": []
}
```

Se o comando não retornar resultados depois de mais ou menos 5 minutos, use as informações a seguir para ajudar você a solucionar problemas com os nós gerenciados.

Tópicos

- Solução 1: verifique se o SSM Agent está instalado e executando em seus nós gerenciados.
- Solução 2: verifique se um perfil da instância do IAM foi especificado para a instância (somente instâncias do EC2)
- Solução 3: Verifique a conectividade dos endpoints do serviço
- Solução 4: Verifique o suporte do sistema operacional de destino
- Solução 5: verifique se você está trabalhando na mesma Região da AWS da instância do Amazon EC2
- Solução 6: verifique a configuração de proxy aplicada ao SSM Agent em seu nó gerenciado
- Solução 7: instale um certificado TLS em instâncias gerenciadas
- Solucionar problemas de disponibilidade do nó gerenciado usando a ssm-cli

Solução 1: verifique se o SSM Agent está instalado e executando em seus nós gerenciados.

Verifique se a versão mais recente do SSM Agent está instalada em seu nó gerenciado.

Para determinar se o SSM Agent está instalado e em execução em um nó gerenciado, consulte Verificar o status do SSM Agent e iniciar o agente.

Para instalar ou reinstalar o SSM Agent em um nó gerenciado, consulte os seguintes tópicos:

- Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux
- Como instalar o SSM Agent em nós híbridos do Linux
- Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Windows Server
- Como instalar o SSM Agent em nós híbridos do Windows

Solução 2: verifique se um perfil da instância do IAM foi especificado para a instância (somente instâncias do EC2)

Para instâncias do Amazon Elastic Compute Cloud (Amazon EC2), verifique se a instância está configurada com um perfil da instância do AWS Identity and Access Management (IAM) que permite que a ela se comunique com a API do Systems Manager. Verifique também se o usuário tem uma política de confiança do IAM que permite que o usuário se comunique com a API do Systems Manager.

> **Note**
>
> Os servidores on-premises, dispositivos de borda e máquinas virtuais (VMs) usam uma função de serviço do IAM em vez de um perfil da instância. Para obter mais informações, consulte Criar o perfil de serviço do IAM obrigatório para o Systems Manager em ambientes híbridos e multinuvem.

Para determinar se um perfil da instância com as permissões necessárias está anexado a uma instância do EC2

1. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, escolha Instances (Instâncias).
3. Selecione a instância na qual verificar um perfil de instância.
4. Na guia Description (Descrição) do painel inferior, localize IAM role (Função do IAM) e escolha o nome da função.
5. Na página Summary (Resumo) da função para o perfil da instância, na guia Permissions (Permissões), verifique se `AmazonSSMManagedInstanceCore` está listado em Permissions policies (Políticas de permissões).

Se uma política personalizada for usada, verifique se ela fornece as mesmas permissões que o `AmazonSSMManagedInstanceCore`.

Abra `AmazonSSMManagedInstanceCore` no console.

Para obter informações sobre outras políticas que podem ser anexadas a um perfil de instância para o Systems Manager, consulte Configurar permissões de instância obrigatórias para o Systems Manager.

Solução 3: Verifique a conectividade dos endpoints do serviço

Verifique se a instância tem conectividade com os endpoints de serviço do Systems Manager. Essa conectividade é fornecida com a criação e configuração de endpoints da VPC para o Systems Manager ou com a permissão do tráfego de saída HTTPS (porta 443) para os endpoints de serviço.

Para instâncias do Amazon EC2, o endpoint de serviço do Systems Manager para a Região da AWS é usado para registrar a instância se sua configuração de nuvem privada virtual (VPC) permitir o tráfego de saída. No entanto, se a configuração da VPC na qual a instância foi executada não permitir tráfego de saída e você não puder alterar essa configuração para permitir conectividade com os endpoints de serviço público, configure endpoints de interface para sua VPC.

Para obter mais informações, consulte Melhorar a segurança das instâncias do EC2 usando endpoints da VPC para o Systems Manager.

Solução 4: Verifique o suporte do sistema operacional de destino

Verifique se a operação escolhida pode ser executada no tipo de nó gerenciado que você espera ver listado. Algumas operações do Systems Manager podem visar somente instâncias do Windows ou somente instâncias do Linux. Por exemplo, os documentos do Systems Manager (SSM) `AWS-InstallPowerShellModule` e `AWS-ConfigureCloudWatch` podem ser executados somente em instâncias do Windows. Na página Run a command (Executar um comando), se você escolher um desses documentos e selecionar Choose instances manually, (Escolher instâncias manualmente), somente as instâncias do Windows serão listadas e disponibilizadas para seleção.

Solução 5: verifique se você está trabalhando na mesma Região da AWS da instância do Amazon EC2

Instâncias do Amazon EC2 são criadas e disponibilizadas nas Regiões da AWS específicas, como a Região Leste dos EUA (Ohio) (us-east-2) ou Europa (Irlanda) (eu-west-1). Verifique se você está

trabalhando na mesma Região da AWS da instância do Amazon EC2 com a qual você deseja trabalhar. Para obter mais informações, consulte Selecionar uma região, em Conceitos básicos do AWS Management Console.

Solução 6: verifique a configuração de proxy aplicada ao SSM Agent em seu nó gerenciado

Verifique se a configuração de proxy aplicada ao SSM Agent em seu nó gerenciado está correta. Se a configuração de proxy estiver incorreta, o nó não poderá se conectar aos endpoints de serviço necessários ou o Systems Manager poderá identificar incorretamente o sistema operacional do nó gerenciado. Para obter mais informações, consulte Configurar o SSM Agent para usar um proxy em nós do Linux e Configurar o SSM Agent para usar um proxy para instâncias do Windows Server.

Solução 7: instale um certificado TLS em instâncias gerenciadas

Um certificado Transport Layer Security (TLS - Segurança de camada de transporte) deve ser instalado em cada instância gerenciada que você usa com o AWS Systems Manager. Os Serviços da AWS usam esses certificados para criptografar chamadas para outros Serviços da AWS.

Por padrão, um certificado TLS já está instalado em cada instância do Amazon EC2 criada em qualquer Amazon Machine Image (AMI). A maioria dos sistemas operacionais modernos inclui o certificado TLS necessário das autoridades de certificação do Amazon Trust Services em seu armazenamento confiável.

Para verificar se o certificado necessário está instalado na instância, execute o seguinte comando com base no sistema operacional da instância. Substitua a parte referente à *região* do URL por Região da AWS onde o nó gerenciado estiver localizado.

Linux & macOS

```
curl -L https://ssm.region.amazonaws.com
```

Windows

```
Invoke-WebRequest -Uri https://ssm.region.amazonaws.com
```

O comando deve retornar um erro `UnknownOperationException`. Se você receber uma mensagem de erro SSL/TLS, o certificado necessário talvez não esteja instalado.

Se você achar que os certificados CA necessários do Amazon Trust Services não estão instalados nos sistemas operacionais de base, em instâncias criadas de AMIs que não são fornecidas pela

Amazon ou em seus próprios servidores on-premises e VMs, será necessário instalar e habilitar um certificado do Amazon Trust Services ou usar o AWS Certificate Manager (ACM) para criar e gerenciar certificados para um serviço integrado compatível.

Cada instância gerenciada deve ter um dos seguintes certificados Transport Layer Security (TLS) instalado.

- Amazon Root CA 1
- Starfield Services Root Certificate Authority – G2
- Starfield Class 2 Certificate Authority

Para obter informações sobre o ACM, consulte o Guia do usuário do AWS Certificate Manager.

Se os certificados em seu ambiente de computação forem gerenciados por um Group Policy Object (GPO - Objeto de política de grupo), poderá ser necessário configurar a política de grupo para incluir um desses certificados.

Para obter mais informações sobre os certificados Amazon Root e Starfield, consulte a publicação do blog How to Prepare for AWS's Move to Its Own Certificate Authority.

Solucionar problemas de disponibilidade do nó gerenciado usando a **`ssm-cli`**

O `ssm-cli` é uma ferramenta de linha de comando autônoma incluída na instalação do SSM Agent. Ao instalar o SSM Agent 3.1.501.0 ou posterior em uma máquina, você pode executar comandos do `ssm-cli` nessa máquina. A saída desses comandos ajuda a determinar se a máquina atende aos requisitos mínimos para que uma instância do Amazon EC2 ou uma máquina que não é do EC2 que deve ser gerenciada pelo AWS Systems Manager e, portanto, adicionada às listas de nós gerenciados no Systems Manager. (A SSM Agent versão 3.1.501.0 foi lançada em novembro de 2021.)

Requisitos mínimos

Para que uma instância do Amazon EC2 ou máquina que não é do EC2 seja gerenciada pelo AWS Systems Manager e esteja disponível em listas de nós gerenciados, ela deve atender a três requisitos principais:

- O SSM Agent deve ser instalado e executado em um nó com um sistema operacional compatível.

  Algumas Amazon Machine Images (AMIs) gerenciadas pela AWS para o EC2 estão configuradas para iniciar instâncias com o SSM Agent pré-instalado. (Você também pode configurar uma AMI

para pré-instalar o SSM Agent.) Para ter mais informações, consulte Encontrar AMIs com o SSM Agent pré-instalado.

- É necessário anexar à máquina um perfil de instância do AWS Identity and Access Management (IAM), para instâncias do EC2, ou um perfil de serviço do IAM, para instâncias que não são do EC2, que fornece as permissões necessárias para se comunicar com o serviço Systems Manager.
- O SSM Agent deve ser capaz de se conectar a um endpoint do Systems Manager para se registrar no serviço. Posteriormente, o nó gerenciado deve estar disponível para o serviço e, para confirmar isso, o serviço envia um sinal a cada cinco minutos para verificar a integridade do nó gerenciado.

Comandos pré-configurados em **`ssm-cli`**

Comandos pré-configurados estão incluídos para coletar as informações necessárias que ajudam a identificar por que a máquina que você confirmou que está em execução não está incluída nas listas de nós gerenciados no Systems Manager. Esses comandos são executados quando você especifica a opção `get-diagnostics`.

Na máquina, execute o comando a seguir para usar o `ssm-cli` para ajudar você a solucionar problemas de disponibilidade de nós gerenciados.

Linux & macOS

```
ssm-cli get-diagnostics --output table
```

Windows

Em máquinas Windows Server, você deve navegar até o diretório `C:\Program Files\Amazon\SSM` antes de executar o comando.

```
ssm-cli.exe get-diagnostics --output table
```

PowerShell

Em máquinas Windows Server, você deve navegar até o diretório `C:\Program Files\Amazon\SSM` antes de executar o comando.

```
.\ssm-cli.exe get-diagnostics --output table
```

Esse comando retorna uma saída como uma tabela semelhante à seguinte.

> **Note**
>
> As verificações de conectividade para os endpoints `ssmmessages`, `s3`, `kms`, `logs` e `monitoring` são para recursos opcionais adicionais, como o Session Manager, que pode fazer login no Amazon Simple Storage Service (Amazon S3) ou no Amazon CloudWatch Logs e usar a criptografia AWS Key Management Service (AWS KMS).

Linux & macOS

```
[root@instance]# ssm-cli get-diagnostics --output table
##############################################################################################
# Check                                  # Status  # Note
                                        #
##############################################################################################
# EC2 IMDS                               # Success # IMDS is accessible and has
 instance id i-0123456789abcdefa in Region  #
#                                       #         # us-east-2
                                        #
##############################################################################################
# Hybrid instance registration           # Skipped # Instance does not have hybrid
 registration                            #
##############################################################################################
# Connectivity to ssm endpoint           # Success # ssm.us-east-2.amazonaws.com is
 reachable                              #
##############################################################################################
# Connectivity to ec2messages endpoint  # Success # ec2messages.us-
east-2.amazonaws.com is reachable                    #
##############################################################################################
# Connectivity to ssmmessages endpoint  # Success # ssmmessages.us-
east-2.amazonaws.com is reachable                    #
##############################################################################################
# Connectivity to s3 endpoint            # Success # s3.us-east-2.amazonaws.com is
 reachable                               #
##############################################################################################
# Connectivity to kms endpoint           # Success # kms.us-east-2.amazonaws.com is
 reachable                              #
##############################################################################################
# Connectivity to logs endpoint         # Success # logs.us-east-2.amazonaws.com is
 reachable                             #
##############################################################################################
```

```
# Connectivity to monitoring endpoint   # Success # monitoring.us-
east-2.amazonaws.com is reachable                        #
##############################################################################
# AWS Credentials                       # Success # Credentials are for
                                      #
#                                       #         #
 arn:aws:sts::123456789012:assumed-role/Fullaccess/i-0123456789abcdefa #
#                                       #         # and will expire at 2021-08-17
 18:47:49 +0000 UTC                      #
##############################################################################
# Agent service                         # Success # Agent service is running and is
 running as expected user              #
##############################################################################
# Proxy configuration                   # Skipped # No proxy configuration detected
                                      #
##############################################################################
# SSM Agent version                     # Success # SSM Agent version is 3.0.1209.0,
 latest available agent version is    #
#                                       #         # 3.1.192.0
                                      #
##############################################################################
```

Windows Server and PowerShell

```
PS C:\Program Files\Amazon\SSM> .\ssm-cli.exe get-diagnostics --output table
##############################################################################
# Check                                 # Status  # Note
                                    #
##############################################################################
# EC2 IMDS                              # Success # IMDS is accessible and has
 instance id i-0123456789EXAMPLE in       #
#                                       #         # Region us-east-2
                                    #
##############################################################################
# Hybrid instance registration          # Skipped # Instance does not have hybrid
 registration                          #
##############################################################################
# Connectivity to ssm endpoint          # Success # ssm.us-east-2.amazonaws.com is
 reachable                            #
##############################################################################
# Connectivity to ec2messages endpoint  # Success # ec2messages.us-
east-2.amazonaws.com is reachable                    #
##############################################################################
```

```
# Connectivity to ssmmessages endpoint  # Success # ssmmessages.us-
east-2.amazonaws.com is reachable                    #
##############################################################################################
# Connectivity to s3 endpoint           # Success # s3.us-east-2.amazonaws.com is
 reachable                             #
##############################################################################################
# Connectivity to kms endpoint          # Success # kms.us-east-2.amazonaws.com is
 reachable                            #
##############################################################################################
# Connectivity to logs endpoint         # Success # logs.us-east-2.amazonaws.com is
 reachable                           #
##############################################################################################
# Connectivity to monitoring endpoint   # Success # monitoring.us-
east-2.amazonaws.com is reachable                     #
##############################################################################################
# AWS Credentials                       # Success # Credentials are for
                                    #
#                                       #         #
 arn:aws:sts::123456789012:assumed-role/SSM-Role/i-123abc45EXAMPLE  #
#                                       #         #  and will expire at 2021-09-02
 13:24:42 +0000 UTC                   #
##############################################################################################
# Agent service                         # Success # Agent service is running and is
 running as expected user            #
##############################################################################################
# Proxy configuration                   # Skipped # No proxy configuration detected
                                    #
##############################################################################################
# Windows sysprep image state           # Success # Windows image state value is at
 desired value IMAGE_STATE_COMPLETE  #
##############################################################################################
# SSM Agent version                     # Success # SSM Agent version is 3.2.815.0,
 latest agent version in us-east-2   #
#                                       #         # is 3.2.985.0
                                    #
##############################################################################################
```

A tabela a seguir fornece detalhes adicionais para cada uma das verificações realizadas pela `ssm-cli`.

Verificações de diagnóstico do **`ssm-cli`**

Verificar	Detalhes
Serviço de metadados da instância do Amazon EC2	Indica se o nó gerenciado consegue acessar o serviço de metadados. Um teste com falha indica um problema de conectividade para `http://169.254.169.254` , que pode ser causado por configurações de proxy e firewall de rota local, proxy ou sistema operacional (SO).
Registro de instância híbrida	Indica se o SSM Agent está registrado usando uma ativação híbrida.
Conectividade com o endpoint `ssm`	Indica se o nó é capaz de alcançar endpoints de serviço do Systems Manager na porta TCP 443. Um teste com falha indica problemas de conectividade com `https://ssm.` *`region`*`.amazonaws.com` , dependendo da Região da AWS onde o nó está localizado. Problemas de conectividade podem ser causados pela configuração da VPC, incluindo grupos de segurança, listas de controle de acesso à rede, tabelas de rotas ou firewalls e proxies do SO.
Conectividade com o endpoint `ec2messages`	Indica se o nó é capaz de alcançar endpoints de serviço do Systems Manager na porta TCP 443. Um teste com falha indica problemas de conectividade com `https://ec2messages.` *`region`*`.amazonaws.com` , dependendo da Região da AWS onde o nó está localizado. Problemas de conectividade podem ser causados pela configuração da VPC, incluindo grupos de segurança, listas de controle de acesso à rede, tabelas de rotas ou firewalls e proxies do SO.

Verificar	Detalhes
Conectividade com o endpoint `ssmmessages`	Indica se o nó é capaz de alcançar endpoints de serviço do Systems Manager na porta TCP 443. Um teste com falha indica problemas de conectividade com `https://ssmmessages.` *region* `.amazonaws.com` , dependendo da Região da AWS onde o nó está localizado. Problemas de conectividade podem ser causados pela configuração da VPC, incluindo grupos de segurança, listas de controle de acesso à rede, tabelas de rotas ou firewalls e proxies do SO.
Conectividade com o endpoint `s3`	Indica se o nó é capaz de alcançar os endpoints de serviço do Amazon Simple Storage Service na porta TCP 443. Um teste com falha indica problemas de conectividade com `https://s3.` *region* `.amazonaws.com` , dependendo da Região da AWS onde o nó está localizado. A conectividade com esse endpoint não é necessária para que o nó seja exibido na lista de nós gerenciados.
Conectividade com o endpoint `kms`	Indica se o nó é capaz de alcançar o endpoint de serviço do AWS Key Management Service na porta TCP 443. Um teste com falha indica problemas de conectividade com `https://kms.` *region* `.amazonaws.com` , dependendo da Região da AWS onde o nó está localizado. A conectividade com esse endpoint não é necessária para que o nó seja exibido na lista de nós gerenciados.

Verificar	Detalhes
Conectividade com o endpoint `logs`	Indica se o nó é capaz de alcançar o endpoint de serviço do Amazon CloudWatch Logs na porta TCP 443. Um teste com falha indica problemas de conectividade com `https://logs.` *`region`*`.amazonaws.com`, dependendo da Região da AWS onde o nó está localizado. A conectividade com esse endpoint não é necessária para que o nó seja exibido na lista de nós gerenciados.
Conectividade com o endpoint `monitoring`	Indica se o nó é capaz de alcançar o endpoint de serviço do Amazon CloudWatch na porta TCP 443. Um teste com falha indica problemas de conectividade com `https://monitoring.` *`region`*`.amazonaws.com`, dependendo da Região da AWS onde o nó está localizado. A conectividade com esse endpoint não é necessária para que o nó seja exibido na lista de nós gerenciados.
Credenciais da AWS	Indica se o SSM Agent tem as credenciais necessárias com base no perfil de instância do IAM (para instâncias do EC2) ou no perfil de serviço do IAM (para máquinas que não são do EC2) anexadas à máquina. Um teste com falha indica que nenhum perfil de instância do IAM ou perfil de serviço do IAM está anexado à máquina ou que não contém as permissões necessárias para o Systems Manager.

Verificar	Detalhes
Serviço de agente	Indica se o serviço do SSM Agent está em execução e se está sendo executado como raiz para Linux ou macOS como SYSTEM para Windows Server. Um teste com falha indica que o serviço do SSM Agent não está em execução ou não está sendo executado como raiz ou SYSTEM.
Configuração do proxy	Indica se o SSM Agent está configurado para usar um proxy.
Estado da imagem do Sysprep (somente Windows)	Indica o estado do `Sysprep` no nó. O SSM Agent não iniciará no nó se o estado do `Sysprep` for um valor diferente de `IMAGE_STATE_COMPLETE` .
Versão do SSM Agent	Indica se a versão mais recente disponível do SSM Agent está instalada.

Ativações híbridas do AWS Systems Manager

Para configurar máquinas que não são do EC2 para uso com o AWS Systems Manager em um ambiente híbrido e multinuvem, crie uma ativação híbrida. Entre os tipos de máquinas não EC2 que podem ser usadas como nós gerenciados estão:

- Servidores em suas próprias instalações (servidores on-premises)
- Dispositivos principais do AWS IoT Greengrass
- AWS IoT e dispositivos de borda que não são da AWS
- Máquinas virtuais (VMs), inclusive VMs em outros ambientes de nuvem

Ao executar o comando create-activation para iniciar um processo de ativação híbrida, você receberá um código de ativação e um ID na resposta do comando. Em seguida, inclua o código de ativação e o ID com o comando para instalar o SSM Agent na máquina, conforme descrito na etapa 3 de

Instalar o SSM Agent em nós híbridos do Linux e na etapa 4 de Instalar o SSM Agent em nós híbridos do Windows Server.

Esse processo de ativação se aplica a todos os tipos de máquinas que não são do EC2, exceto aos dispositivos principais do AWS IoT Greengrass. Para obter informações sobre como configurar os dispositivos principais do AWS IoT Greengrass para o Systems Manager, consulte Gerenciar dispositivos de borda com o Systems Manager.

> **Note**
>
> Atualmente, não há suporte para máquinas macOS que não sejam do EC2.

Sobre camadas de instâncias do Systems Manager

O AWS Systems Manager oferece um nível de instâncias padrão e um nível de instâncias avançadas. As duas opções oferecem suporte a nós gerenciados em seu ambiente híbrido e multinuvem. O nível de instâncias padrão permite registrar no máximo 1.000 máquinas por Conta da AWS e por Região da AWS. Se precisar registrar mais de 1.000 máquinas em uma única conta e região, use o nível de instâncias avançadas. Você pode criar quantos nós gerenciados quiser no nível de instâncias avançadas. Todos os nós gerenciados configurados para o Systems Manager são cobrados de acordo com o uso. Para obter mais informações sobre como habilitar instâncias avançadas, consulte Ativar o nível de instâncias avançadas. Para obter mais informações sobre precificação, consulte Precificação do AWS Systems Manager.

Observe as seguintes informações adicionais sobre o nível de instâncias padrão e o nível de instâncias avançadas:

- Instâncias avançadas também permitem que você se conecte a seus nós que não são do EC2 em um ambiente híbrido e multinuvem usando o AWS Systems Manager Session Manager. O Session Manager fornece acesso via shell interativo às suas instâncias. Para obter mais informações, consulte AWS Systems Manager Session Manager.
- A cota de instâncias padrão também se aplica a instâncias do EC2 que usam uma ativação do Systems Manager on-premises (o que não é um cenário comum).
- Para aplicar patches em aplicações lançadas pela Microsoft em VMs (máquinas virtuais) e instâncias on-premises, ative o nível de instâncias avançadas. Há uma cobrança para o uso do nível de instâncias avançadas. Não há custo adicional para aplicar patches em aplicações lançadas pela Microsoft nas instâncias do Amazon Elastic Compute Cloud (Amazon EC2). Para

obter mais informações, consulte Aplicações de patches lançadAs pela Microsoft no Windows Server.

AWS Systems Manager Inventory

O inventário do AWS Systems Manager fornece visibilidade de seu ambiente de computação da AWS. Você pode usar o Inventory para coletar metadados dos nós gerenciados. Você pode armazenar esses metadados em um bucket central do Amazon Simple Storage Service (Amazon S3) e usar ferramentas integradas para consultar os dados e determinar rapidamente quais nós estão executando o software, as configurações exigidas pela política de software e quais nós precisam ser atualizados. Você pode configurar o Inventário em todos os nós gerenciados usando um procedimento de apenas um clique. Você também pode configurar e visualizar dados do inventário de várias Regiões da AWS e Contas da AWS usando o Amazon Athena. Para começar a usar o inventário, abra o console do Systems Manager. No painel de navegação, escolha Inventory.

Se os tipos pré-configurados de metadados coletados pelo Systems Manager Inventory não atenderem às suas necessidades, você poderá criar um inventário personalizado. O inventário personalizado é simplesmente um arquivo JSON com informações que você fornece e adiciona ao nó gerenciado em um diretório específico. Quando o Systems Manager Inventory coleta dados, ele captura esses dados de inventário personalizados. Por exemplo, se você executa um datacenter grande, poderá especificar o local do rack de cada um dos seus servidores como um inventário personalizado. Dessa forma, é possível visualizar os dados de espaço do rack junto com outros dados de inventário.

> **Important**
>
> O Systems Manager Inventory coleta somente metadados dos nós gerenciados. O Inventory não acessa informações ou dados proprietários.

A tabela a seguir descreve os tipos de dados que você pode coletar com o Systems Manager Inventory. A tabela também descreve diferentes ofertas para nós de destino e os intervalos de coleta que você pode especificar.

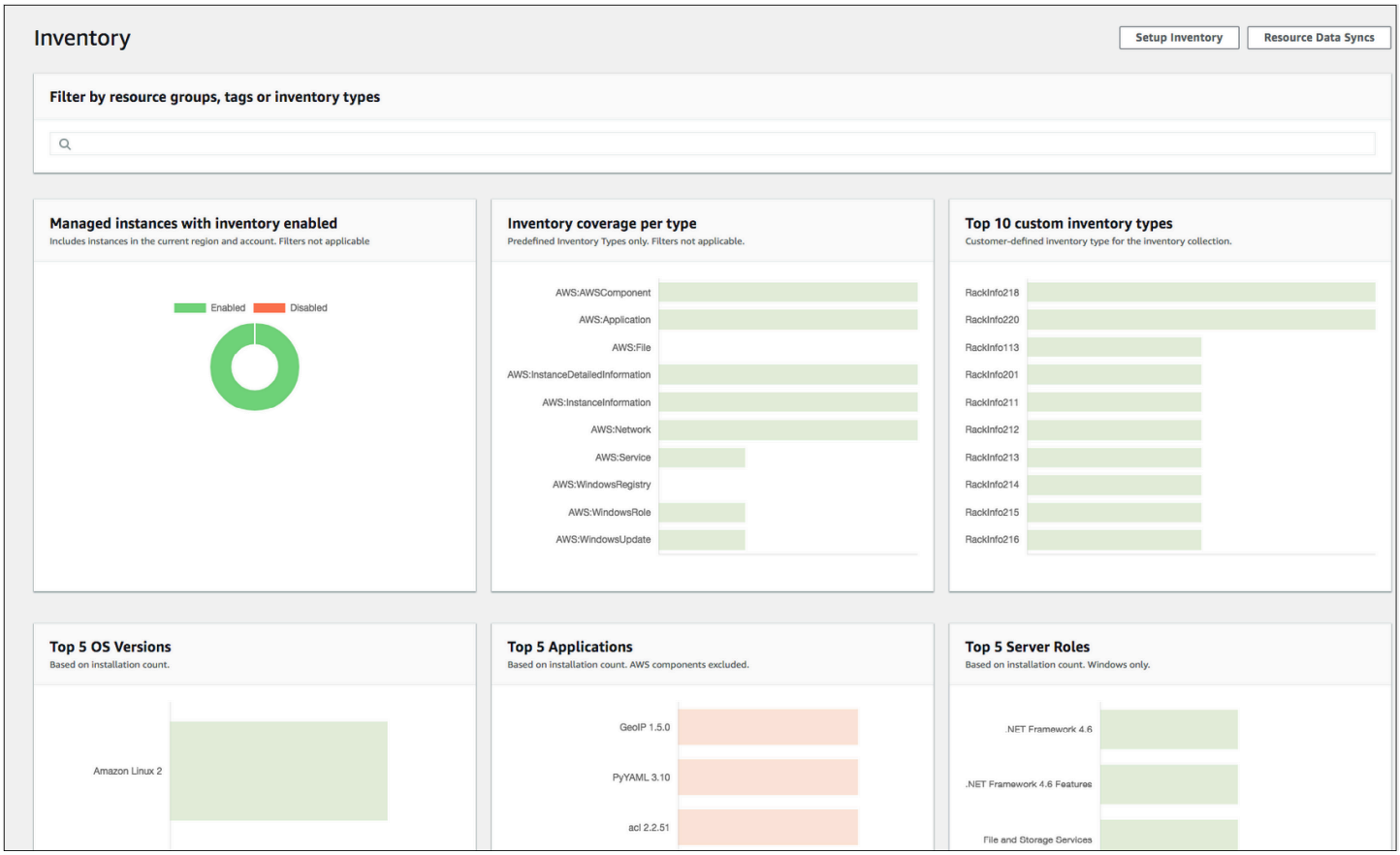
Configuração	Detalhes
Tipos de metadados	Você pode configurar o inventário para coletar os seguintes tipos de dados: • Aplicativos: nomes de aplicativos, editores, versões etc. • Componentes da AWS: driver do EC2, agentes, versões etc. • Arquivos: nome, tamanho, versão, data de instalação, horário de modificação e último acesso etc. • Configuração de rede:: endereço IP, endereço MAC, DNS, gateway, máscara de sub-rede etc. • Atualizações do Windows: ID de hotfix, instalado por, instalado por data etc. • Detalhes da instância: CPUModel, CPUCores, CPUs, CPUSpeedMHz, CPUSockets, CPUHyperThreadEnabled, OSServicePacketc. • Serviços: nome, nome de exibição, status, serviços dependentes, tipo de serviço, tipo de início etc. • Tags (Etiquetas): tags atribuídas a seus nós. • Registro do Windows: caminho da chave do registro, nome do valor, tipo de valor e valor. • Funções do Windows: nome, nome de exibição, caminho, tipo de recurso, estado de instalação etc. • Inventário personalizado: metadados que foram atribuídos a um nó gerenciado, conforme descrito em Trabalhar com inventário personalizado.

Configuração	Detalhes
	Note Para ver uma lista de todos os metadados coletados pelo Inventário, consulte Metadata collected by Inventory.
Nós para destino	Você pode optar por inventariar todos os nós gerenciados em sua Conta da AWS. Selecione nós ou grupos de destinos de nós individualmente usando tags. Para obter mais informações sobre como executar a coleta de inventário em todos seus nós gerenciados, consulte Inventário de todos os nós gerenciados na sua Conta da AWS.
Quando coletar informações	Você pode especificar um intervalo de coleta em termos de minutos, horas e dias. O menor intervalo de coleta é a cada 30 minutos.

Note

Dependendo da quantidade de dados coletados, o sistema pode demorar vários minutos para informar os dados na saída que você especificou. Após a coleta das informações, os dados são enviados por um canal HTTPS seguro para um repositório da AWS de texto simples, acessível apenas na sua conta da Conta da AWS.

Você pode ver os dados na página Inventory (Inventário) do console do Systems Manager, que inclui vários cartões predefinidos para ajudar a consultar os dados.



> **Note**
>
> Os cartões do Inventory filtram automaticamente as instâncias gerenciadas do Amazon EC2, cujo estado sejam Terminated (Terminada) e Stopped (Interrompida). Para nós gerenciados do dispositivo principal do AWS IoT Greengrass e on-premises, os cartões do Inventory removem automaticamente os nós cujo estado é Terminated (Encerrado).

Se você criar uma sincronização de dados de recursos para sincronizar e armazenar todos os seus dados em um único bucket do Amazon S3, você poderá detalhar esses dados na página Inventory Detailed View (Visualização detalhada do Inventory). Para obter mais informações, consulte Consultar dados de inventário de várias regiões e contas.

Suporte ao EventBridge

Essa ferramenta do Systems Manager pode ser usada como um tipo de evento nas regras do Amazon EventBridge. Para obter informações, consulte Monitorar eventos do Systems Manager com

o Amazon EventBridge e Referência: padrões e tipos de eventos do Amazon EventBridge para o Systems Manager.

Conteúdo

- Saiba mais sobre o Systems Manager Inventory
- Configurar o Systems Manager Inventory
- Configurar a coleta de inventário
- Consultar dados de inventário de várias regiões e contas
- Consultar uma coleta de inventário usando filtros
- Agregar dados do inventário
- Trabalhar com inventário personalizado
- Visualizar o histórico do inventário e o controle de alterações
- interromper a coleta de dados e excluir os dados do inventário
- Assigning custom inventory metadata to a managed node
- Using the AWS CLI to configure inventory data collection
- Demonstração: usar a sincronização de dados de recursos para agregar dados do inventário
- Solucionar problemas com o Systems Manager Inventory

Saiba mais sobre o Systems Manager Inventory

Ao configurar o AWS Systems Manager Inventory, você especifica o tipo de metadados a ser coletado, os nós gerenciados de onde os metadados devem ser coletados e um cronograma para a coleta de metadados. Essas configurações são salvas com a Conta da AWS como uma associação do AWS Systems Manager State Manager. Uma associação é simplesmente uma configuração.

> **Note**
>
> O Inventário apenas coleta metadados. Ele não coleta dados pessoais ou de propriedade.

Tópicos

- Metadata collected by Inventory
- Trabalhar com o inventário de arquivos e do Registro do Windows

Metadata collected by Inventory

O exemplo a seguir mostra a lista completa de metadados coletados por cada plugin do AWS Systems Manager Inventory.

```
{
    "typeName": "AWS:InstanceInformation",
    "version": "1.0",
    "attributes":[
      { "name": "AgentType",                                "dataType" : "STRING"},
      { "name": "AgentVersion",                             "dataType" : "STRING"},
      { "name": "ComputerName",                             "dataType" : "STRING"},
      { "name": "InstanceId",                               "dataType" : "STRING"},
      { "name": "IpAddress",                                "dataType" : "STRING"},
      { "name": "PlatformName",                             "dataType" : "STRING"},
      { "name": "PlatformType",                             "dataType" : "STRING"},
      { "name": "PlatformVersion",                          "dataType" : "STRING"},
      { "name": "ResourceType",                             "dataType" : "STRING"},
      { "name": "AgentStatus",                              "dataType" : "STRING"},
      { "name": "InstanceStatus",                           "dataType" : "STRING"}
    ]
  },
  {
    "typeName" : "AWS:Application",
    "version": "1.1",
    "attributes":[
      { "name": "Name",               "dataType": "STRING"},
      { "name": "ApplicationType",    "dataType": "STRING"},
      { "name": "Publisher",          "dataType": "STRING"},
      { "name": "Version",            "dataType": "STRING"},
      { "name": "Release",            "dataType": "STRING"},
      { "name": "Epoch",              "dataType": "STRING"},
      { "name": "InstalledTime",      "dataType": "STRING"},
      { "name": "Architecture",       "dataType": "STRING"},
      { "name": "URL",                "dataType": "STRING"},
      { "name": "Summary",            "dataType": "STRING"},
      { "name": "PackageId",          "dataType": "STRING"}
    ]
  },
  {
    "typeName" : "AWS:File",
    "version": "1.0",
    "attributes":[
      { "name": "Name",               "dataType": "STRING"},
```

```
      { "name": "Size",           "dataType": "STRING"},
      { "name": "Description",        "dataType": "STRING"},
      { "name": "FileVersion",        "dataType": "STRING"},
      { "name": "InstalledDate",      "dataType": "STRING"},
      { "name": "ModificationTime",   "dataType": "STRING"},
      { "name": "LastAccessTime",     "dataType": "STRING"},
      { "name": "ProductName",        "dataType": "STRING"},
      { "name": "InstalledDir",       "dataType": "STRING"},
      { "name": "ProductLanguage",    "dataType": "STRING"},
      { "name": "CompanyName",        "dataType": "STRING"},
      { "name": "ProductVersion",       "dataType": "STRING"}
    ]
  },
  {
    "typeName": "AWS:AWSComponent",
    "version": "1.0",
    "attributes":[
      { "name": "Name",               "dataType": "STRING"},
      { "name": "ApplicationType",    "dataType": "STRING"},
      { "name": "Publisher",          "dataType": "STRING"},
      { "name": "Version",            "dataType": "STRING"},
      { "name": "InstalledTime",      "dataType": "STRING"},
      { "name": "Architecture",       "dataType": "STRING"},
      { "name": "URL",                "dataType": "STRING"}
    ]
  },
  {
    "typeName": "AWS:WindowsUpdate",
    "version":"1.0",
    "attributes":[
      { "name": "HotFixId",           "dataType": "STRING"},
      { "name": "Description",        "dataType": "STRING"},
      { "name": "InstalledTime",      "dataType": "STRING"},
      { "name": "InstalledBy",        "dataType": "STRING"}
    ]
  },
  {
    "typeName": "AWS:Network",
    "version":"1.0",
    "attributes":[
      { "name": "Name",               "dataType": "STRING"},
      { "name": "SubnetMask",         "dataType": "STRING"},
      { "name": "Gateway",            "dataType": "STRING"},
      { "name": "DHCPServer",         "dataType": "STRING"},
```

```
      { "name": "DNSServer",          "dataType": "STRING"},
      { "name": "MacAddress",         "dataType": "STRING"},
      { "name": "IPV4",               "dataType": "STRING"},
      { "name": "IPV6",               "dataType": "STRING"}
    ]
  },
  {
    "typeName": "AWS:PatchSummary",
    "version":"1.0",
    "attributes":[
      { "name": "PatchGroup",                      "dataType": "STRING"},
      { "name": "BaselineId",                      "dataType": "STRING"},
      { "name": "SnapshotId",                      "dataType": "STRING"},
      { "name": "OwnerInformation",                "dataType": "STRING"},
      { "name": "InstalledCount",                  "dataType": "NUMBER"},
      { "name": "InstalledPendingRebootCount",     "dataType": "NUMBER"},
      { "name": "InstalledOtherCount",             "dataType": "NUMBER"},
      { "name": "InstalledRejectedCount",          "dataType": "NUMBER"},
      { "name": "NotApplicableCount",              "dataType": "NUMBER"},
      { "name": "UnreportedNotApplicableCount",    "dataType": "NUMBER"},
      { "name": "MissingCount",                    "dataType": "NUMBER"},
      { "name": "FailedCount",                     "dataType": "NUMBER"},
      { "name": "OperationType",                   "dataType": "STRING"},
      { "name": "OperationStartTime",              "dataType": "STRING"},
      { "name": "OperationEndTime",                "dataType": "STRING"},
      { "name": "InstallOverrideList",             "dataType": "STRING"},
      { "name": "RebootOption",                    "dataType": "STRING"},
      { "name": "LastNoRebootInstallOperationTime", "dataType": "STRING"},
      { "name": "ExecutionId",                     "dataType": "STRING",
   "isOptional": "true"},
      { "name": "NonCompliantSeverity",            "dataType": "STRING",
   "isOptional": "true"},
      { "name": "SecurityNonCompliantCount",       "dataType": "NUMBER",
   "isOptional": "true"},
      { "name": "CriticalNonCompliantCount",       "dataType": "NUMBER",
   "isOptional": "true"},
      { "name": "OtherNonCompliantCount",          "dataType": "NUMBER",
   "isOptional": "true"}
    ]
  },
  {
    "typeName": "AWS:PatchCompliance",
    "version":"1.0",
    "attributes":[
```

```
      { "name": "Title",                          "dataType": "STRING"},
      { "name": "KBId",                           "dataType": "STRING"},
      { "name": "Classification",                 "dataType": "STRING"},
      { "name": "Severity",                       "dataType": "STRING"},
      { "name": "State",                          "dataType": "STRING"},
      { "name": "InstalledTime",                  "dataType": "STRING"}
    ]
  },
  {
    "typeName": "AWS:ComplianceItem",
    "version":"1.0",
    "attributes":[
      { "name": "ComplianceType",                 "dataType": "STRING",
"isContext": "true"},
      { "name": "ExecutionId",                    "dataType": "STRING",
"isContext": "true"},
      { "name": "ExecutionType",                  "dataType": "STRING",
"isContext": "true"},
      { "name": "ExecutionTime",                  "dataType": "STRING",
"isContext": "true"},
      { "name": "Id",                             "dataType": "STRING"},
      { "name": "Title",                          "dataType": "STRING"},
      { "name": "Status",                         "dataType": "STRING"},
      { "name": "Severity",                       "dataType": "STRING"},
      { "name": "DocumentName",                   "dataType": "STRING"},
      { "name": "DocumentVersion",                "dataType": "STRING"},
      { "name": "Classification",                 "dataType": "STRING"},
      { "name": "PatchBaselineId",                "dataType": "STRING"},
      { "name": "PatchSeverity",                  "dataType": "STRING"},
      { "name": "PatchState",                     "dataType": "STRING"},
      { "name": "PatchGroup",                     "dataType": "STRING"},
      { "name": "InstalledTime",                  "dataType": "STRING"},
      { "name": "InstallOverrideList",            "dataType": "STRING",
"isOptional": "true"},
      { "name": "DetailedText",                   "dataType": "STRING",
"isOptional": "true"},
      { "name": "DetailedLink",                   "dataType": "STRING",
"isOptional": "true"},
      { "name": "CVEIds",                         "dataType": "STRING",
"isOptional": "true"}
    ]
  },
  {
    "typeName": "AWS:ComplianceSummary",
```

```
    "version":"1.0",
    "attributes":[
      { "name": "ComplianceType",                "dataType": "STRING"},
      { "name": "PatchGroup",                    "dataType": "STRING"},
      { "name": "PatchBaselineId",               "dataType": "STRING"},
      { "name": "Status",                        "dataType": "STRING"},
      { "name": "OverallSeverity",               "dataType": "STRING"},
      { "name": "ExecutionId",                   "dataType": "STRING"},
      { "name": "ExecutionType",                 "dataType": "STRING"},
      { "name": "ExecutionTime",                 "dataType": "STRING"},
      { "name": "CompliantCriticalCount",        "dataType": "NUMBER"},
      { "name": "CompliantHighCount",            "dataType": "NUMBER"},
      { "name": "CompliantMediumCount",          "dataType": "NUMBER"},
      { "name": "CompliantLowCount",             "dataType": "NUMBER"},
      { "name": "CompliantInformationalCount",   "dataType": "NUMBER"},
      { "name": "CompliantUnspecifiedCount",     "dataType": "NUMBER"},
      { "name": "NonCompliantCriticalCount",     "dataType": "NUMBER"},
      { "name": "NonCompliantHighCount",         "dataType": "NUMBER"},
      { "name": "NonCompliantMediumCount",       "dataType": "NUMBER"},
      { "name": "NonCompliantLowCount",          "dataType": "NUMBER"},
      { "name": "NonCompliantInformationalCount", "dataType": "NUMBER"},
      { "name": "NonCompliantUnspecifiedCount",  "dataType": "NUMBER"}
    ]
  },
  {
    "typeName": "AWS:InstanceDetailedInformation",
    "version":"1.0",
    "attributes":[
      { "name": "CPUModel",                    "dataType": "STRING"},
      { "name": "CPUCores",                    "dataType": "NUMBER"},
      { "name": "CPUs",                        "dataType": "NUMBER"},
      { "name": "CPUSpeedMHz",                 "dataType": "NUMBER"},
      { "name": "CPUSockets",                  "dataType": "NUMBER"},
      { "name": "CPUHyperThreadEnabled",       "dataType": "STRING"},
      { "name": "OSServicePack",               "dataType": "STRING"}
    ]
   },
   {
     "typeName": "AWS:Service",
     "version":"1.0",
     "attributes":[
       { "name": "Name",                       "dataType": "STRING"},
       { "name": "DisplayName",                "dataType": "STRING"},
       { "name": "ServiceType",                "dataType": "STRING"},
```

```
        { "name": "Status",                          "dataType": "STRING"},
        { "name": "DependentServices",              "dataType": "STRING"},
        { "name": "ServicesDependedOn",             "dataType": "STRING"},
        { "name": "StartType",                      "dataType": "STRING"}
     ]
    },
    {
      "typeName": "AWS:WindowsRegistry",
      "version":"1.0",
      "attributes":[
        { "name": "KeyPath",                           "dataType": "STRING"},
        { "name": "ValueName",                         "dataType": "STRING"},
        { "name": "ValueType",                         "dataType": "STRING"},
        { "name": "Value",                             "dataType": "STRING"}
      ]
    },
    {
      "typeName": "AWS:WindowsRole",
      "version":"1.0",
      "attributes":[
        { "name": "Name",                           "dataType": "STRING"},
        { "name": "DisplayName",                    "dataType": "STRING"},
        { "name": "Path",                           "dataType": "STRING"},
        { "name": "FeatureType",                    "dataType": "STRING"},
        { "name": "DependsOn",                      "dataType": "STRING"},
        { "name": "Description",                    "dataType": "STRING"},
        { "name": "Installed",                      "dataType": "STRING"},
        { "name": "InstalledState",                 "dataType": "STRING"},
        { "name": "SubFeatures",                    "dataType": "STRING"},
        { "name": "ServerComponentDescriptor",      "dataType": "STRING"},
        { "name": "Parent",                         "dataType": "STRING"}
      ]
    },
    {
      "typeName": "AWS:Tag",
      "version":"1.0",
      "attributes":[
        { "name": "Key",                      "dataType": "STRING"},
        { "name": "Value",                    "dataType": "STRING"}
      ]
    },
    {
      "typeName": "AWS:ResourceGroup",
      "version":"1.0",
```

```
      "attributes":[
        { "name": "Name",                    "dataType": "STRING"},
        { "name": "Arn",                     "dataType": "STRING"}
      ]
    },
    {
      "typeName": "AWS:BillingInfo",
      "version": "1.0",
      "attributes": [
        { "name": "BillingProductId",        "dataType": "STRING"}
      ]
    }
```

Note

- Para `"typeName": "AWS:InstanceInformation"`, o `InstanceStatus` pode ser um dos seguintes: Active (Ativo), ConnectionLost, Stopped (Parado) ou Terminated (Encerrado).
- Com o lançamento da versão 2.5, RPM Package Manager substitui o atributo serial com Epoch. O atributo Epoch é um número inteiro crescente de forma uniforme, como Serial. Quando você faz um inventário usando o tipo `AWS:Application`, um valor maior para Epoch significa uma versão mais recente. Se valores Epoch forem iguais ou vazios, em seguida, use o valor do atributo Versão para determinar a versão mais recente.
- Alguns metadados não estão disponíveis em instâncias do Linux. Especificamente, para “typeName”: “AWS:Network”, os seguintes tipos de metadados ainda não são compatíveis com instâncias do Linux. Eles SÃO compatíveis com Windows.
  - { "name": "SubnetMask", "dataType": "STRING"},
  - { "name": "DHCPServer", "dataType": "STRING"},
  - { "name": "DNSServer", "dataType": "STRING"},
  - { "name": "Gateway", "dataType": "STRING"},

Trabalhar com o inventário de arquivos e do Registro do Windows

O AWS Systems Manager Inventory permite que você pesquise e faça o inventário de arquivos em sistemas operacionais Windows Linux e macOS. Você pode também pesquisar e inventariar o Registro do Windows.

Arquivos: você pode coletar informações de metadados sobre arquivos, incluindo nomes de arquivo, hora em que os arquivos foram criados, hora da última modificação e acesso e tamanhos de arquivo, entre outras. Para iniciar a coleta do inventário de arquivos, especifique um caminho de arquivo em que você deseja fazer o inventário, um ou mais padrões que definem os tipos de arquivo que deseja inventariar e se o caminho deve ser percorrido recursivamente. O Systems Manager faz o inventário de todos os metadados de arquivos que estiverem no caminho especificado e que corresponderem ao padrão. O inventário de arquivo usa a entrada de parâmetro a seguir.

```
{
"Path": string,
"Pattern": array[string],
"Recursive": true,
"DirScanLimit" : number // Optional
}
```

- Path: o caminho do diretório no qual você deseja inventariar arquivos. No Windows, você pode usar variáveis do ambiente, como %PROGRAMFILES%, desde que a variável aponte para um único caminho de diretório. Por exemplo, se você usar um %PATH% que aponte para vários caminhos de diretório, o Inventário lança um erro.
- Pattern: uma matriz de padrões para identificar arquivos.
- Recursive: um valor booleano que indica se o Inventário deve percorrer recursivamente os diretórios.
- DirScanLimit: um valor opcional que especifica quantos diretórios devem ser percorridos. Use este parâmetro para minimizar o impacto sobre a performance de seus nós gerenciados. O Inventário verifica 5.000 diretórios no máximo.

> **Note**
>
> O Inventário coleta metadados para 500 arquivos no máximo, em todos os caminhos especificados.

Veja alguns exemplos de como especificar os parâmetros ao executar um inventário de arquivos.

- No Linux e macOS, colete metadados de arquivos .sh no diretório `/home/ec2-user`, excluindo todos os subdiretórios.

```
[{"Path":"/home/ec2-user","Pattern":["*.sh", "*.sh"],"Recursive":false}]
```

- No Windows, colete metadados de todos os arquivos ".exe" na pasta Arquivos de programa, incluindo subdiretórios recursivamente.

```
[{"Path":"C:\Program Files","Pattern":["*.exe"],"Recursive":true}]
```

- No Windows, colete metadados de padrões de log específicos.

```
[{"Path":"C:\ProgramData\Amazon","Pattern":["*amazon*.log"],"Recursive":true}]
```

- Limite a contagem de diretórios ao executar uma coleta recursiva.

```
[{"Path":"C:\Users","Pattern":["*.ps1"],"Recursive":true, "DirScanLimit": 1000}]
```

Registro do Windows: você pode coletar chaves e valores do Registro do Windows. Você pode escolher um caminho de chaves e coletar todos os valores e chaves recursivamente. Você pode também coletar determinada chave de registro e o respectivo valor para um caminho específico. O inventário coleta o caminho da chave, o nome, o tipo e o valor.

```
{
"Path": string,
"Recursive": true,
"ValueNames": array[string] // optional
}
```

- Path: o caminho para a chave do Registro.
- Recursive: um valor booleano que indica se o Inventário deve percorrer recursivamente os caminhos do Registro.
- ValueNames: uma matriz de nomes de valor para a realização de inventário de chaves do Registro. Se você usar esse parâmetro, o Systems Manager criará um inventário de apenas os nomes de valor especificados para o caminho especificado.

> **Note**
>
> O Inventário coleta no máximo 250 valores de chave do Registro para todos os caminhos especificados.

Veja alguns exemplos de como especificar os parâmetros ao realizar um inventário de do Registro do Windows.

- Colete todos os valores e chaves recursivamente para um caminho específico.

```
[{"Path":"HKEY_LOCAL_MACHINE\SOFTWARE\Amazon","Recursive": true}]
```

- Colete todos os valores e chaves para um caminho específico (pesquisa recursiva desativada).

```
[{"Path":"HKEY_LOCAL_MACHINE\SOFTWARE\Intel\PSIS\PSIS_DECODER", "Recursive": false}]
```

- Colete uma chave específica usando a opção `ValueNames`.

```
{"Path":"HKEY_LOCAL_MACHINE\SOFTWARE\Amazon\MachineImage","ValueNames":["AMIName"]}
```

Configurar o Systems Manager Inventory

Antes de usar o AWS Systems Manager Inventory para coletar metadados sobre as aplicações, os serviços, os componentes da AWS e outros em execução nos nós gerenciados, recomendamos configurar a sincronização de dados dos recursos para centralizar o armazenamento de dados do inventário em um único bucket do Amazon Simple Storage Service (Amazon S3). Recomendamos também configurar o monitoramento do Amazon EventBridge em eventos de inventário. Esses processos facilitam a visualização e gerenciam os dados e a coleta do inventário.

Tópicos

- Creating a resource data sync for Inventory
- Sobre o monitoramento de eventos do Inventory do EventBridge

Creating a resource data sync for Inventory

Este tópico descreve como instalar e configurar a sincronização de dados de recursos para o inventário do AWS Systems Manager. Para obter informações sobre sincronização de dados de

recursos para Systems Manager Explorer, consulte Configurar o Systems Manager Explorer para exibir dados de várias contas e regiões.

Sobre a sincronização de dados de recursos

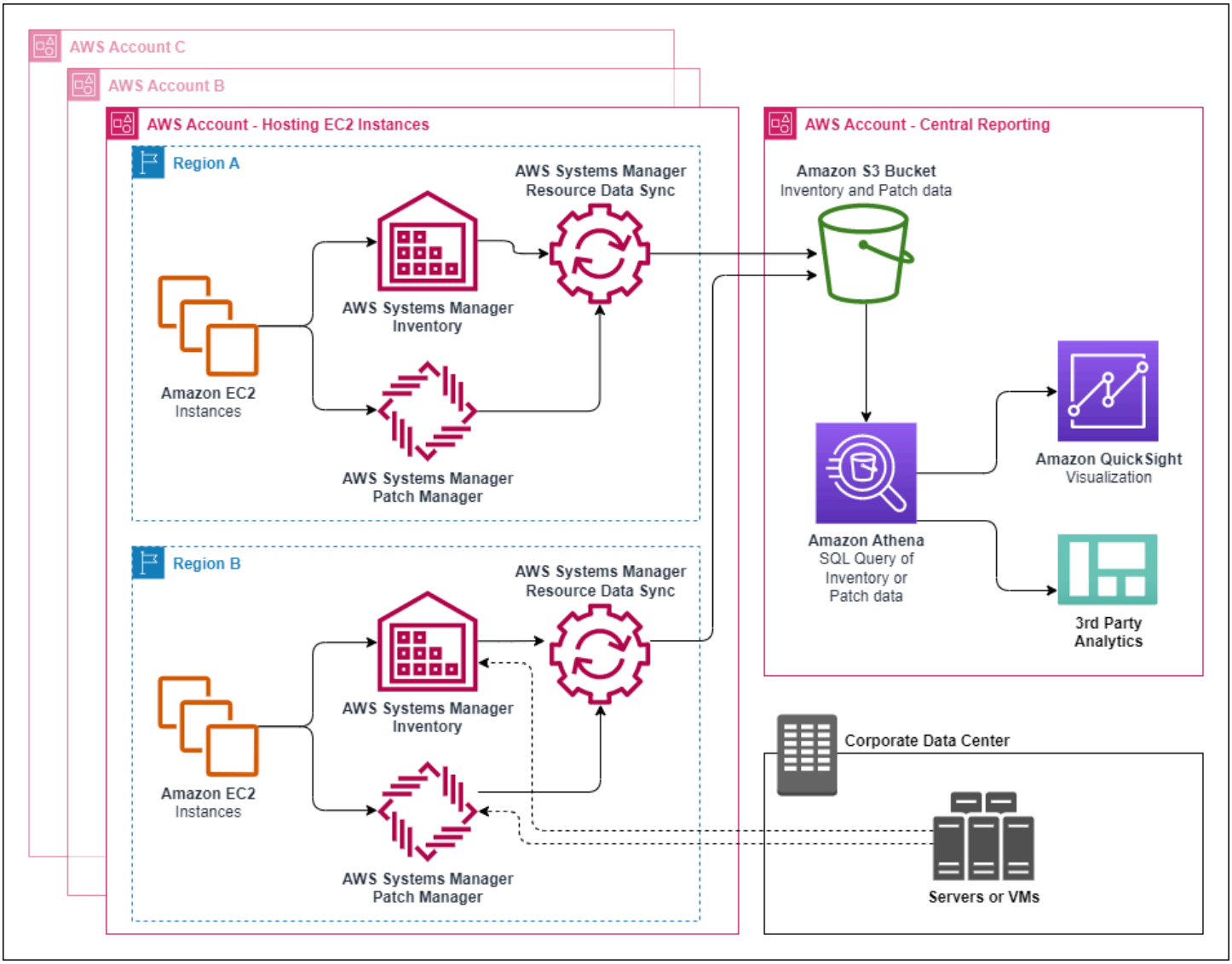
Você pode usar a sincronização de dados de recursos do Systems Manager para enviar dados do inventário coletados de todas os nós gerenciados para um único bucket do Amazon Simple Storage Service (Amazon S3). A sincronização de dados de recursos atualizará automaticamente os dados centralizados quando novos dados de inventário forem coletados. Com todos os dados do inventário armazenados em um bucket do Amazon S3 de destino, você pode usar serviços como o Amazon Athena e Amazon QuickSight para consultar e analisar os dados agregados.

Por exemplo, digamos que você tenha configurado o inventário para coletar dados sobre o sistema operacional (SO) e as aplicações em execução em uma frota de 150 nós gerenciados. Alguns desses nós estão localizados em um data center on-premises e outros são executados no Amazon Elastic Compute Cloud (Amazon EC2), em várias Regiões da AWS. Se você não tiver configurado a sincronização de dados de recursos, precisará coletar os dados de inventário coletados em cada nó ou criar scripts para coletar essas informações. Você precisaria então compatibilizar os dados em um aplicativo para poder executar consultas e analisá-los.

Com a sincronização de dados de recursos, você executa uma operação única que sincroniza todos os dados de inventário de todas os nós gerenciados. Depois que a sincronização for criada com êxito, o Systems Manager criará uma lista de referência de todos os dados do inventário e salvará essa lista no bucket do Amazon S3 de destino. Quando novos dados de inventário são coletados, o Systems Manager atualiza automaticamente os dados no bucket do Amazon S3. Você pode então fazer a portabilidade dos dados de forma rápida e econômica para o Amazon Athena e o Amazon QuickSight.

O Diagrama 1 mostra como a sincronização de dados de recursos agrega dados de inventário do Amazon EC2 outros tipos de máquina em um ambiente híbrido e multinuvem para um bucket do Amazon S3 de destino. Esse diagrama também mostra como a sincronização de dados de recursos funciona com várias contas Contas da AWS e Regiões da AWS.

Diagrama 1: sincronização de dados de recursos com as Contas da AWS e Regiões da AWS



Se você excluir um nó gerenciado, a sincronização de dados de recursos preservará o arquivo de inventário do nó excluído. Para nós em execução, a sincronização de dados de recursos substitui automaticamente os arquivos de inventário antigos quando arquivos forem criados e gravados no bucket do Amazon S3. Se quiser acompanhar as alterações de inventário ao longo do tempo, você poderá usar o serviço AWS Config para rastrear o tipo de recurso `SSM:ManagedInstanceInventory`. Para obter mais informações, consulte o tópico Conceitos básicos sobre a AWS Config.

Use os procedimentos desta seção para criar uma sincronização de dados de recursos para o Inventory usando os consoles do Amazon S3 e do AWS Systems Manager. Você também pode usar o AWS CloudFormation para criar ou excluir uma sincronização de dados de recursos. Para usar

o AWS CloudFormation, adicione o recurso `AWS::SSM::ResourceDataSync` ao modelo AWS CloudFormation. Para obter informações, consulte um dos seguintes recursos da documentação:

- O recurso do AWS CloudFormation para sincronização de dados de recursos no AWS Systems Manager (blog)
- Trabalhar com modelos do AWS CloudFormation no Manual do usuário do AWS CloudFormation

> **Note**
>
> Você pode usar o AWS Key Management Service (AWS KMS) para criptografar dados de inventário no bucket do Amazon S3. Para obter um exemplo de como criar uma sincronização criptografada usando a AWS Command Line Interface (AWS CLI) e como trabalhar com os dados centralizados no Amazon Athena e no Amazon QuickSight, consulte Demonstração: usar a sincronização de dados de recursos para agregar dados do inventário.

Antes de começar

Antes de criar uma sincronização de dados de recurso, use o procedimento a seguir para criar um bucket central do Amazon S3 para armazenar dados de inventário agregados. O procedimento descreve como atribuir uma política de bucket que permite que o Systems Manager grave os dados do inventário no bucket de várias contas. Se você já tiver um bucket do Amazon S3 que deseja usar para agregar dados de inventário para sincronização de dados de recursos, configure o bucket para usar a política no procedimento a seguir.

> **Note**
>
> O Systems Manager Inventory não pode adicionar dados a um bucket do Amazon S3 especificado se esse bucket estiver configurado para usar o Object Lock. Verifique se o bucket do Amazon S3 que você cria ou escolhe para a sincronização de dados de recursos não está configurado para usar o Amazon S3 Object Lock. Para obter mais informações, consulte Como o bloqueio de objetos do Amazon S3 funciona no Guia do usuário do Amazon Simple Storage Service.

Para criar e configurar um bucket do Amazon S3 para a sincronização de dados de recursos

1. Abra o console do Amazon S3, em https://console.aws.amazon.com/s3/.

2. Crie um bucket para armazenar os dados agregados do Inventory. Para obter mais informações, consulte Criar um bucket no Guia do usuário do Amazon Simple Storage Service. Anote o nome do bucket e a Região da AWS em que você o criou.
3. Escolha a guia Permissions e depois escolha Bucket Policy.
4. Copie e cole a seguinte política de bucket no editor de políticas. Substitua *amzn-s3-demo-bucket* pelo nome do bucket do S3 que você criou. Substitua *account_ID_number* por um número de ID de Conta da AWS válido.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SSMBucketPermissionsCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
        },
        {
            "Sid": " SSMBucketDelivery",
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/*/
accountid=account_ID_number/*",
                "arn:aws:s3:::amzn-s3-demo-bucket/*/
accountid=account_ID_number/*",
                "arn:aws:s3:::amzn-s3-demo-bucket/*/
accountid=account_ID_number/*",
                "arn:aws:s3:::amzn-s3-demo-bucket/*/
accountid=account_ID_number/*"
            ],
            "Condition": {
                "StringEquals": {
```

```
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": "ID_number"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:ssm:*:111122223333:resource-
data-sync/*"
                }
            }
        }
    ]
}
```

5. Salve as alterações.

Criar uma sincronização de dados do recurso para o Inventory

Use o procedimento a seguir para criar uma sincronização de dados de recursos do Systems Manager Inventory usando o console do Systems Manager. Para obter informações sobre como criar uma sincronização de dados de recursos usando a AWS CLI, consulte Using the AWS CLI to configure inventory data collection.

Para criar uma sincronização de dados de recurso

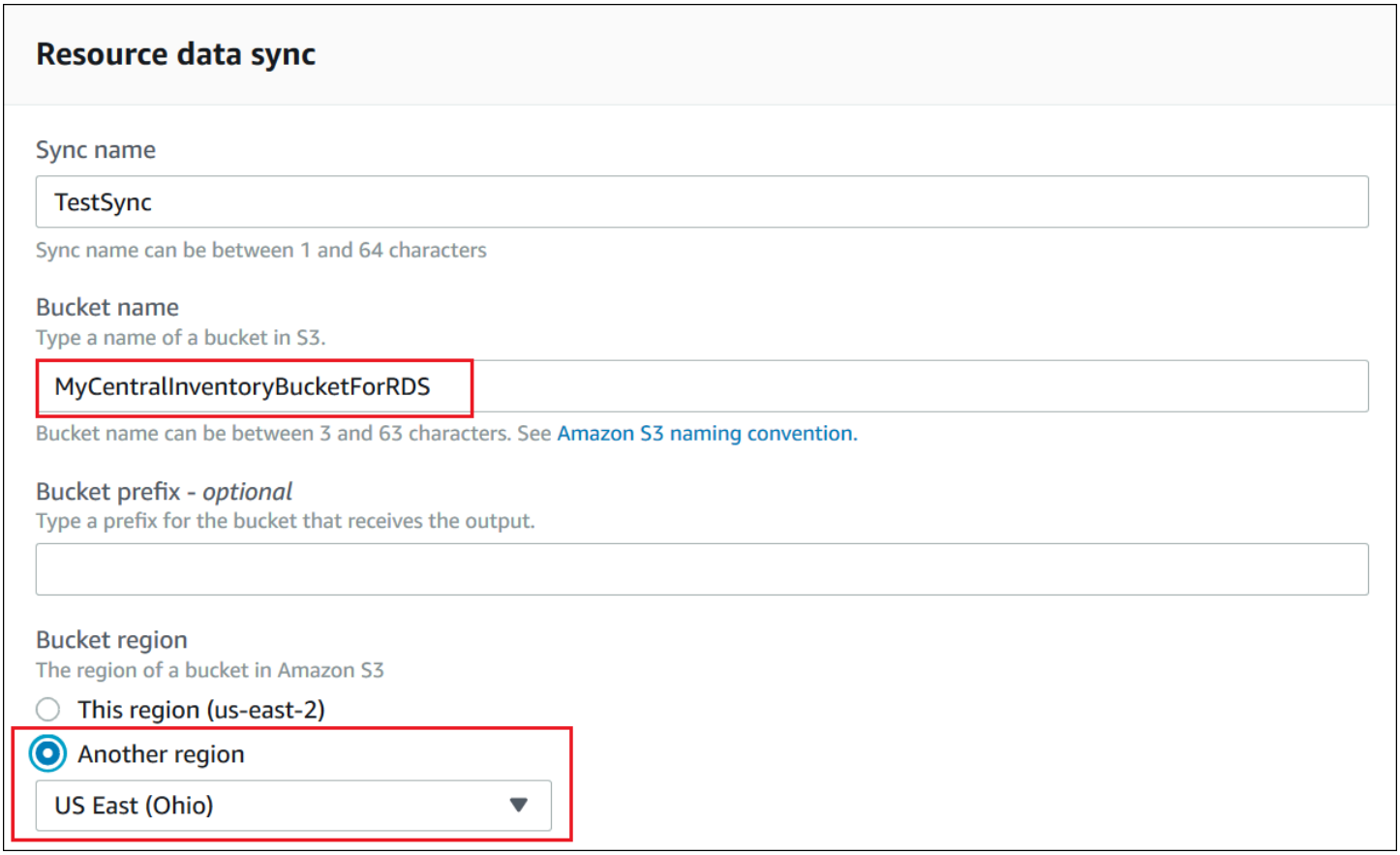
1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. No menu Account management (Gerenciamento de contas), selecione Resource data syncs (Sincronizações de dados do recurso).
4. Escolha Create resource data sync (Criar sincronização de dados de recurso).
5. No campo Sync name (Nome da sincronização), digite um nome para a configuração de sincronização.
6. No campo Bucket name (Nome do bucket), insira o nome do bucket do Amazon S3 que você criou usando o procedimento Como criar e configurar um bucket do Amazon S3 para sincronização de dados de recursos.
7. (Opcional) No campo Bucket prefix (Prefixo do bucket), digite o nome de um prefixo de bucket do Amazon S3 (subdiretório).
8. No campo Bucket region (Região do bucket), escolha This region (Esta região) se o bucket do Amazon S3 que você criou estiver localizado na região atual da Região da AWS. Se o bucket

estiver localizado em uma Região da AWS diferente, escolha Another region (Outra região) e digite o nome da região.

> **Note**
>
> Se a sincronização e o bucket do Amazon S3 de destino estiverem localizados em diferentes regiões, você poderá estar sujeito ao preço da transferência de dados. Para obter mais informações, consulte Preços do Amazon S3.

9. (Opcional) No campo KMS Key ARN (ARN da chave do KMS), digite ou cole um ARN da chave do KMS para criptografar dados de inventário no Amazon S3.
10. Escolha Criar.

Para sincronizar os dados do inventário de várias Regiões da AWS, você deve criar uma sincronização de dados de recursos em cada região. Repita esse procedimento em cada Região da AWS onde você quiser coletar dados de inventário e envie-os para o bucket central do Amazon S3. Ao criar a sincronização em cada região, especifique o bucket central do Amazon S3 no campo Bucket name (Nome do bucket). Em seguida, use a opção Bucket region (Região do bucket) para escolher a região em que você criou o bucket central do Amazon S3, conforme mostrado na captura de tela a seguir. Na próxima vez que a associação for executada para coletar dados de inventário, o Systems Manager armazenará os dados no bucket central do Amazon S3.



Criar uma sincronização de dados de recursos do inventário para várias contas definidas no AWS Organizations

Você pode sincronizar os dados do inventário de Contas da AWS definidas no AWS Organizations com um bucket central do Amazon S3 Depois de concluir o procedimento a seguir, os dados de inventário serão sincronizados com prefixos de chave individuais do Amazon S3 no bucket central. Cada prefixo de chave representa um ID de Conta da AWS diferente.

Antes de começar

Antes de começar, verifique se você definiu e configurou várias Contas da AWS e AWS Organizations. Para obter mais informações, consulte a no Manual do usuário do AWS Organizations.

Além disso, será necessário criar a sincronização de dados do recurso baseada na organização para cada Região da AWS e Conta da AWS definido em AWS Organizations.

Criar um bucket central do Amazon S3

Use o procedimento a seguir para criar um bucket central do Amazon S3 para armazenar dados do inventário agregados. O procedimento descreve como atribuir uma política de bucket que permite que o Systems Manager grave os dados do inventário no bucket no ID da conta do AWS Organizations. Se você já tiver um bucket do Amazon S3 que deseja usar para agregar dados de inventário para sincronização de dados de recursos, configure o bucket para usar a política no procedimento a seguir.

Para criar e configurar um bucket do Amazon S3 para a sincronização de dados de recursos de várias contas definidas no AWS Organizations

1. Abra o console do Amazon S3, em https://console.aws.amazon.com/s3/.
2. Crie um bucket para armazenar os dados de inventário agregados. Para obter mais informações, consulte Criar um bucket no Guia do usuário do Amazon Simple Storage Service. Anote o nome do bucket e a Região da AWS em que você o criou.
3. Escolha a guia Permissions e depois escolha Bucket Policy.
4. Copie e cole a seguinte política de bucket no editor de políticas. Substitua *amzn-s3-demo-bucket* e *organization-id* pelo nome do bucket do Amazon S3 que você criou e um ID de conta do AWS Organizations válido.

   Opcionalmente, substitua *bucket-prefix* pelo nome de um prefixo do Amazon S3 (subdiretório). Se você não tiver criado um prefixo, remova o *bucket-prefix/* do ARN na seguinte política.

   JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SSMBucketPermissionsCheck",
      "Effect": "Allow",
      "Principal": {
        "Service": "ssm.amazonaws.com"
      },
      "Action": "s3:GetBucketAcl",
      "Resource": "arn:aws:s3:::S3_bucket_name"
    },
```

```
    {
      "Sid": " SSMBucketDelivery",
      "Effect": "Allow",
      "Principal": {
        "Service": "ssm.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/bucket-prefix/*/accountid=*/*"
      ],
      "Condition": {
        "StringEquals": {
          "s3:x-amz-acl": "bucket-owner-full-control",
          "aws:SourceOrgID": "organization-id"
                  }
      }
    },
    {
      "Sid": " SSMBucketDeliveryTagging",
      "Effect": "Allow",
      "Principal": {
        "Service": "ssm.amazonaws.com"
      },
      "Action": "s3:PutObjectTagging",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/bucket-prefix/*/accountid=*/*"
      ]
    }
  ]
}
```

Crie uma sincronização de dados de recursos do inventário para várias contas definidas no AWS Organizations

O procedimento a seguir descreve como usar a AWS CLI para criar uma sincronização de dados de recursos para contas que estiverem definidas no AWS Organizations. Use a AWS CLI para executar estas etapas. Você também deve executar este procedimento para cada Região da AWS e Conta da AWS definidas em AWS Organizations.

Para criar uma sincronização de dados de recursos de várias contas definidas na AWS Organizations (AWS CLI)

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Execute o comando a seguir para verificar se você não tem nenhuma outra sincronização de dados de recursos baseada em AWS Organizations. Você pode ter várias sincronizações padrão, incluindo várias sincronizações padrão e uma sincronização baseada em organizações. Contudo, você só pode ter uma sincronização de dados de recurso baseada em organização.

   ```
   aws ssm list-resource-data-sync
   ```

   Se o comando retornar outra sincronização de dados de recurso, baseada em organização, você deverá excluí-la ou optar por não criar uma nova.

3. Execute o comando a seguir para criar uma sincronização de dados de recursos para uma conta definida no AWS Organizations. Para o amzn-s3-demo-bucket, especifique o nome do bucket do Amazon S3 criado anteriormente neste tópico. Se você criou um prefixo (subdiretório) para o bucket, especifique essas informações em *prefix-name*.

   ```
   aws ssm create-resource-data-sync --sync-name name --s3-
   destination "BucketName=amzn-s3-demo-bucket,Prefix=prefix-
   name,SyncFormat=JsonSerDe,Region=Região da AWS, for example us-
   east-2,DestinationDataSharing={DestinationDataSharingType=Organization}"
   ```

4. Repita as etapas 2 e 3 para cada Região da AWS e Conta da AWS onde deseja sincronizar dados com o bucket central do Amazon S3.

Gerenciar a sincronização de dados de recursos

Cada Conta da AWS pode conter cinco sincronizações de dados de recursos por Região da AWS. Você pode usar o console do Fleet Manager do AWS Systems Manager para gerenciar sincronizações de dados de recursos.

Para visualizar sincronizações de dados de recursos

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.

3. No menu suspenso Gerenciamento de contas, selecione Sincronizações de dados do recurso.
4. Selecione uma sincronização de dados de recursos na tabela e escolha Visualizar detalhes para ver informações sobre a sincronização de dados de recursos.

Para excluir uma sincronização de dados de recurso

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. No menu suspenso Gerenciamento de contas, selecione Sincronizações de dados do recurso.
4. Selecione uma sincronização de dados de recursos na tabela e escolha Excluir.

Sobre o monitoramento de eventos do Inventory do EventBridge

Você pode configurar uma regra no Amazon EventBridge para criar um evento em resposta às alterações no estado do recurso do AWS Systems Manager Inventory. O EventBridge oferece suporte a eventos para as seguintes alterações de estado do Inventory. Todos os eventos são enviados com base no melhor esforço.

Tipo de inventário personalizado excluído para uma instância específica: se uma regra estiver configurada para monitorar esse evento, o EventBridge criará um evento quando um tipo de inventário personalizado em uma instância específica for excluído. O EventBridge envia um evento por nó para cada tipo de inventário personalizado. Aqui está um exemplo de padrão de evento.

```
{
    "timestampMillis": 1610042981103,
    "source": "SSM",
    "account": "123456789012",
    "type": "INVENTORY_RESOURCE_STATE_CHANGE",
    "startTime": "Jan 7, 2021 6:09:41 PM",
    "resources": [
        {
            "arn": "arn:aws:ssm:us-east-1:123456789012:managed-instance/i-12345678"
        }
    ],
    "body": {
        "action-status": "succeeded",
        "action": "delete",
        "resource-type": "managed-instance",
        "resource-id": "i-12345678",
```

```
        "action-reason": "",
        "type-name": "Custom:MyCustomInventoryType"
    }
}
```

Tipo de inventário personalizado excluiu o evento para todas as instâncias: se uma regra estiver configurada para monitorar esse evento, o EventBridge criará um evento quando um tipo de inventário personalizado para todos os nós gerenciados for excluído. Aqui está um exemplo de padrão de evento.

```
{
    "timestampMillis": 1610042904712,
    "source": "SSM",
    "account": "123456789012",
    "type": "INVENTORY_RESOURCE_STATE_CHANGE",
    "startTime": "Jan 7, 2021 6:08:24 PM",
    "resources": [

    ],
    "body": {
        "action-status": "succeeded",
        "action": "delete-summary",
        "resource-type": "managed-instance",
        "resource-id": "",
        "action-reason": "The delete for type name Custom:SomeCustomInventoryType
 was completed. The deletion summary is: {\"totalCount\":1,\"remainingCount\":0,
\"summaryItems\":[{\"version\":\"1.1\",\"count\":1,\"remainingCount\":0}]}",
        "type-name": "Custom:MyCustomInventoryType"
    }
}
```

Chamada do PutInventory com o evento de versão do esquema antigo: se uma regra estiver configurada para monitorar esse evento, o EventBridge criará um evento quando uma chamada PutInventory, que usa uma versão de esquema inferior ao esquema atual, for feita. Este evento se aplica a todos os tipos de inventário. Aqui está um exemplo de padrão de evento.

```
{
    "timestampMillis": 1610042629548,
    "source": "SSM",
    "account": "123456789012",
    "type": "INVENTORY_RESOURCE_STATE_CHANGE",
    "startTime": "Jan 7, 2021 6:03:49 PM",
```

```
    "resources": [
        {
            "arn": "arn:aws:ssm:us-east-1:123456789012:managed-instance/i-12345678"
        }
    ],
    "body": {
        "action-status": "failed",
        "action": "put",
        "resource-type": "managed-instance",
        "resource-id": "i-01f017c1b2efbe2bc",
        "action-reason": "The inventory item with type name
 Custom:MyCustomInventoryType was sent with a disabled schema verison 1.0. You must
 send a version greater than 1.0",
        "type-name": "Custom:MyCustomInventoryType"
    }
}
```

Para obter informações sobre como configurar o EventBridge para monitorar esses eventos, consulte Configurar o EventBridge para eventos do Systems Manager.

Configurar a coleta de inventário

Esta seção descreve como configurar a coleta do AWS Systems Manager Inventory em um ou mais nós gerenciados usando o console do Systems Manager. Para obter um exemplo de como configurar a coleta de inventário usando a AWS Command Line Interface (AWS CLI), consulte Using the AWS CLI to configure inventory data collection.

Quando você configura a coleta de inventário, primeiro cria uma associação do AWS Systems Manager do State Manager. O Systems Manager coleta os dados de inventário quando a associação é executada. Se você não criar a associação primeiro e tentar invocar o plugin `aws:softwareInventory` usando, por exemplo, o AWS Systems Manager Run Command, o sistema retornará o seguinte erro: `The aws:softwareInventory plugin can only be invoked via ssm-associate.`

> **Note**
>
> Se você criar várias associações de inventário para um nó gerenciado, observe o seguinte comportamento:
>
> - Cada nó pode ser atribuído a uma associação de inventário direcionada a todos os nós (— targets “Key=InstanceIds, Values=*”).

- Cada nó também pode ser atribuído a uma associação específica que usa pares de tags/chave/valor ou um grupo de recursos da AWS.
- Se um nó tiver várias associações de inventário atribuídas, o status mostrará Skipped (Ignorado) para a associação que não foi executada. A associação que foi executada mais recentemente exibe o status real da associação de inventário.
- Se um nó for atribuído a várias associações de inventário e cada uma usar um par de chave/valor de marca, essas associações de inventário não serão executadas nesse nó devido ao conflito de tags. A associação ainda é executada em nós que não apresentarem conflito entre chave e valor na tag.

Antes de começar

Antes de configurar a coleta de inventário, conclua as seguintes tarefas.

- Atualize o AWS Systems Manager SSM Agent nos nós que você deseja incluir no inventário. Ao executar a versão mais recente do SSM Agent, você garante que pode coletar metadados para todos os tipos de inventário comportados. Para obter informações sobre como atualizar o SSM Agent usando o State Manager, consulte Walkthrough: Automatically update SSM Agent with the AWS CLI.
- Verifique se você concluiu os requisitos de configuração para instâncias do Amazon Elastic Compute Cloud (Amazon EC2) e máquinas que não são do EC2 em um ambiente híbrido e multinuvem. Para mais informações, consulte Configurar nós gerenciados para o AWS Systems Manager.
- Para nós do Microsoft Windows, verifique se o nó gerenciado está configurado com o Windows PowerShell 3.0 (ou posterior). O SSM Agent usa o cmdlet do `ConvertTo-Json` no PowerShell para converter dados de inventário de atualização do Windows para o formato necessário.
- (Opcional) Crie uma sincronização de dados de recursos para armazenar de forma centralizada os dados do inventário em um bucket do Amazon S3. A sincronização de dados de recursos atualizará automaticamente os dados centralizados quando novos dados de inventário forem coletados. Para obter mais informações, consulte Demonstração: usar a sincronização de dados de recursos para agregar dados do inventário.
- (Opcional) Crie um arquivo JSON para coletar inventário personalizado. Para obter mais informações, consulte Trabalhar com inventário personalizado.

Inventário de todos os nós gerenciados na sua Conta da AWS

Você pode criar facilmente um inventário de todos os nós gerenciados na sua Conta da AWS por meio de uma associação de inventário global. Uma associação global de inventário realiza as seguintes ações:

- Aplica automaticamente a configuração (associação) do inventário global a todos os nós gerenciados existentes na sua Conta da AWS. Os nós gerenciados que já possuem uma associação de inventário são ignoradas quando a associação de inventário global é aplicada e executada. Quando um nó for ignorado, o status detalhado da mensagem mostrará `Overridden By Explicit Inventory Association`. Esses nós são ignorados pela associação global, mas ainda informam o inventário quando executam sua associação de inventário atribuída.
- Adiciona automaticamente novos nós gerenciados criados na sua conta da Conta da AWS para a associação global do inventário.

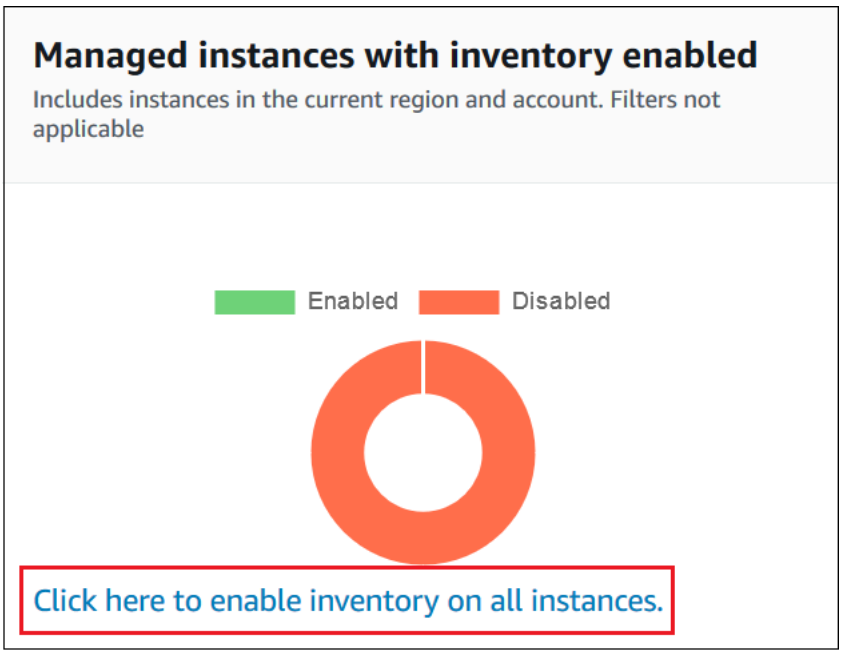
> **Note**
>
> - Se um nó gerenciado for configurado para a associação de inventário global e você atribuir uma associação específica a esse nó, o Systems Manager Inventory retirará a prioridade da associação global e aplicará a associação específica.
> - As associações globais de inventário estão disponíveis no SSM Agent versão 2.0.790.0 ou posterior. Para obter informações sobre como atualizar o SSM Agent em nós gerenciados, consulte Atualização do SSM Agent por meio de Run Command.

Configurar a coleção do inventário com um único clique (console)

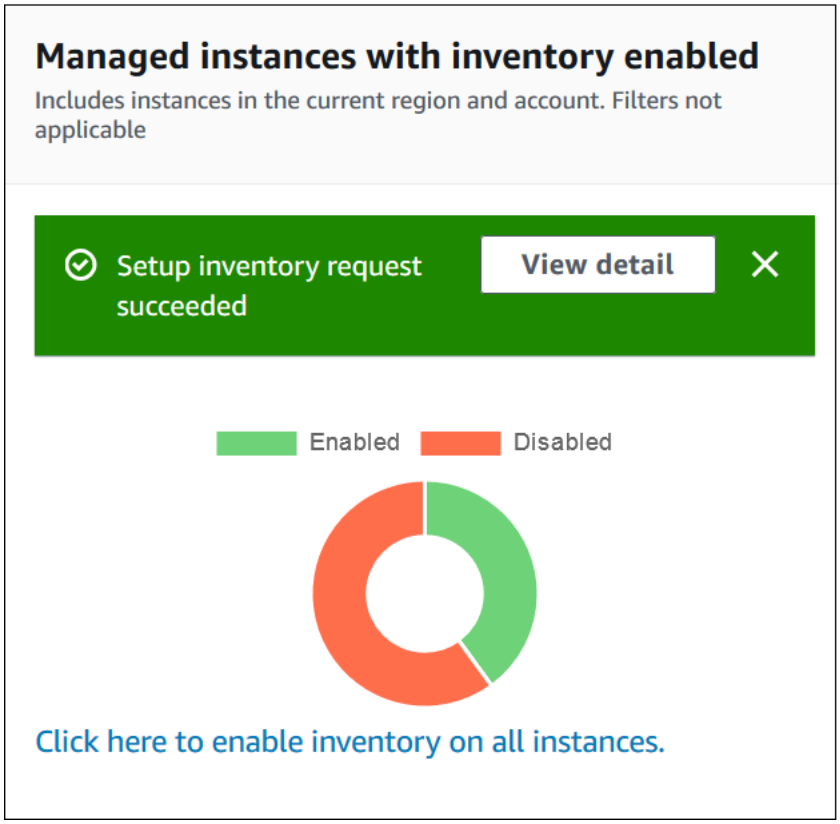
Use o procedimento a seguir para configurar o Systems Manager Inventory para todos os nós gerenciados na Conta da AWS e em uma única Região da AWS.

Para configurar todos os nós gerenciados na região atual para o Systems Manager Inventory

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Inventory.
3. No cartão Managed instances with inventory enabled (Instâncias gerenciadas com inventário habilitado), escolha Click here to enable inventory on all instances (Clique aqui para habilitar o inventário em todas as instâncias).



Se a operação for bem-sucedida, o console exibirá a seguinte mensagem.



Dependendo do número de nós gerenciados na sua conta, poderá levar vários minutos para que a associação de inventário global seja aplicada. Aguarde alguns minutos e atualize a página.

Verifique se o gráfico é alterado para refletir que o inventário está configurado em todos os nós gerenciados.

Configurar a coleta com o uso do console

Esta seção inclui informações sobre como configurar o Systems Manager Inventory para coletar metadados de seus nós gerenciados usando o console do Systems Manager. Você pode coletar rapidamente os metadados de todos os nós em uma determinada Conta da AWS (e em qualquer nó futuro que possa ser criado nessa conta) ou coletar seletivamente dados do inventário usando tags ou IDs de nós.

> **Note**
>
> Antes de concluir este procedimento, verifique se já existe uma associação de inventário global. Se já houver uma associação de inventário global, sempre que você iniciar uma nova instância, a associação será aplicada a ela, e a nova instância será inventariada.

Para configurar a coleta de inventário

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Inventory.
3. Escolha Setup Inventory.
4. Na seção Targets (Destinos), identifique os nós onde você deseja executar essa operação, escolhendo uma das opções a seguir.

   - Selecting all managed instances in this account (Selecionar todas as instâncias gerenciadas nesta conta) – Esta opção seleciona todos os nós gerenciados para os quais não há associação de inventário existente. Se você escolher essa opção, os nós que já tinham associações de inventário serão ignorados durante a coleta de inventário e mostrados com um status Skipped (Ignorado) nos resultados do inventário. Para obter mais informações, consulte Inventário de todos os nós gerenciados na sua Conta da AWS.
   - Specifying a tag (Especificar uma tag): use essa opção para especificar uma única tag para identificar os nós em sua conta, dos quais você deseja coletar o inventário. Se você usar uma tag, todos os nós criados no futuro com a mesma tag também relatarão o inventário. Se houver uma associação de inventário existente com todos os nós, o uso de uma tag para selecionar nós específicos como um destino para um inventário diferente substituirá

a associação do nó nesse grupo de destino All managed instances (Todas as instâncias gerenciadas). Os nós gerenciados com a tag especificada são ignorados em coletas de inventário futuras em All managed instances (Todas as instâncias gerenciadas).

- Manually selecting instances (Seleção manual de instâncias): use essa opção para escolher nós gerenciados específicos na sua conta. A escolha explícita de nós específicos usando essa opção substitui as associações de inventário no destino All managed instances (Todas as instâncias gerenciadas). O nó é ignorado na coleção de inventário futura de All managed instances (Todas as instâncias gerenciadas).

> **Note**
>
> Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

5. Na seção Schedule (Programação), escolha com que frequência deseja que o sistema colete metadados de inventário dos nós.
6. Na seção Parameters (Parâmetros), use as listas para ativar ou desativar diferentes tipos de coleta de inventário. Para obter mais informações sobre como coletar inventário de um arquivo e do Registro do Windows, consulte Trabalhar com o inventário de arquivos e do Registro do Windows.
7. Na seção Advanced (Avançado), escolha Sync inventory execution logs to an Amazon S3 bucket (Sincronizar logs de execução do inventário com um bucket do Amazon S3), se quiser armazenar o status de execução da associação em um bucket do Amazon S3.
8. Escolha Setup Inventory. O Systems Manager cria uma associação do State Manager e executa imediatamente o Inventory em nós.
9. No painel de navegação, escolha State Manager. Verifique se uma nova associação que usa o documento **`AWS-GatherSoftwareInventory`** foi criada A programação de associação usa uma expressão de taxa. Além disso, verifique se o campo Status mostra Success. Se você escolheu a opção Sync inventory execution logs to an Amazon S3 bucket (Sincronizar os logs de execução do inventário com um bucket do Amazon S3), poderá visualizar os dados do log no Amazon S3 depois de alguns minutos. Se você quiser visualizar os dados de inventário de um nó específico, escolha Managed Instances (Instâncias gerenciadas) no painel de navegação.
10. Escolha um nó e View details (Visualizar detalhes).
11. Na página de detalhes do nó, selecione Inventory (Inventário). Use as listas Inventory type para filtrar o inventário.

Consultar dados de inventário de várias regiões e contas

O AWS Systems Manager Inventory se integra ao Amazon Athena para ajudar você a consultar dados do inventário de várias Regiões da AWS e Contas da AWS. A integração com o Athena usa a sincronização de dados dos recursos para que você possa visualizar os dados do inventário de todas as instâncias gerenciadas na página Detailed View (Visualização detalhada) no console do AWS Systems Manager.

> **Important**
>
> Este recurso usa o AWS Glue para rastrear os dados no bucket do Amazon Simple Storage Service (Amazon S3) e no Amazon Athena para consultar os dados. Dependendo da quantidade de dados rastreados e consultados, você pode ser cobrado pelo uso desses serviços. Com o AWS Glue, você pode pagar uma taxa horária, cobrada por segundo, para crawlers (descoberta de dados) e trabalhos de ETL (processamento e carga de dados). Com o Athena, você é cobrado de acordo com a quantidade de dados verificados por cada consulta. Recomendamos que você visualize as orientações de preço para esses serviços antes de usar a integração do Amazon Athena com o Systems Manager Inventory. Para obter mais informações, consulte Preço do Amazon Athena e Preço do AWS Glue.

Você pode visualizar dados de inventário na página Detail View (Detalhes do inventário) em todas as Regiões da AWS onde o Amazon Athena estiver disponível. Para obter uma lista das regiões e dos endpoints compatíveis, consulte Amazon Athena Service Endpoints no Referência geral da Amazon Web Services.

Antes de começar

A integração com o Athena usa aos sincronização de dados d recursos. Você deve definir e configurar a sincronização de dados de recursos para usar esse recurso. Para obter mais informações, consulte Demonstração: usar a sincronização de dados de recursos para agregar dados do inventário.

Além disso, lembre-se de que a página Detail View (Visualização de detalhes) exibe dados de inventário para o proprietário do bucket central do Amazon S3 usado pela sincronização de dados dos recursos. Se você não for o proprietário do bucket central do Amazon S3, não poderá ver os dados de inventário na página Detail View (Visualização de detalhes).

Configurar o acesso

Antes de consultar e visualizar dados de diversas contas e regiões na página Visualização detalhada no console do Systems Manager, é necessário configurar a entidade do IAM com permissões para a visualização de dados.

Se os dados de inventário estiverem armazenados em um bucket do Amazon S3 que usa criptografia do AWS Key Management Service (AWS KMS), você também deve configurar a entidade do IAM e o perfil de serviço `Amazon-GlueServiceRoleForSSM` para a criptografia do AWS KMS.

Tópicos

- Como configurar a entidade do IAM para acessar a página “Visualização detalhada”
- (Opcional) Configure as permissões para exibição de dados criptografados do AWS KMS

Como configurar a entidade do IAM para acessar a página “Visualização detalhada”

A seguir, as permissões mínimas requeridas para visualizar os dados de inventário na página Visualização detalhada são descritas.

A política gerenciada **`AWSQuicksightAthenaAccess`**.

O `PassRole` a seguir e o bloco de permissões adicionais requerido

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowGlue",
            "Effect": "Allow",
            "Action": [
                "glue:GetCrawler",
                "glue:GetCrawlers",
                "glue:GetTables",
                "glue:StartCrawler",
                "glue:CreateCrawler"
            ],
            "Resource": "*"
        },
```

```
        {
            "Sid": "iamPassRole",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::123456789012:role/SSMInventoryGlueRole",
                "arn:aws:iam::123456789012:role/SSMInventoryServiceRole"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "glue.amazonaws.com"
                }
            }
        },
        {
            "Sid": "iamRoleCreation",
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:AttachRolePolicy"
            ],
            "Resource": "arn:aws:iam::111122223333:role/*"
        },
        {
            "Sid": "iamPolicyCreation",
            "Effect": "Allow",
            "Action": "iam:CreatePolicy",
            "Resource": "arn:aws:iam::111122223333:policy/*"
        }
    ]
}
```

(Opcional) Se o bucket do Amazon S3 usado para armazenar dados de inventário for criptografado usando o AWS KMS, você também deverá adicionar o seguinte bloco à política:

```
{
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt"
    ],
    "Resource": [
        "arn:aws:kms:Region:account_ID:key/key_ARN"
```

```
    ]
}
```

Para conceder acesso, adicione as permissões aos seus usuários, grupos ou perfis:

- Usuários e grupos no AWS IAM Identity Center:

  Crie um conjunto de permissões. Siga as instruções em Criação de um conjunto de permissões no Guia do usuário do AWS IAM Identity Center.
- Usuários gerenciados no IAM com provedor de identidades:

  Crie um perfil para a federação de identidades. Siga as instruções em Criando um perfil para um provedor de identidades de terceiros (federação) no Guia do Usuário do IAM.
- Usuários do IAM:
  - Crie um perfil que seu usuário possa assumir. Siga as instruções em Criação de um perfil para um usuário do IAM no Guia do usuário do IAM.
  - (Não recomendado) Vincule uma política diretamente a um usuário ou adicione um usuário a um grupo de usuários. Siga as instruções em Adição de permissões a um usuário (console) no Guia do usuário do IAM.

(Opcional) Configure as permissões para exibição de dados criptografados do AWS KMS

Se o bucket do Amazon S3 usado para armazenar os dados de inventário for criptografado usando o AWS Key Management Service (AWS KMS), você deverá configurar a entidade do IAM e o perfil Amazon-GlueServiceRoleForSSM com permissões `kms:Decrypt` para a chave do AWS KMS.

Antes de começar

Para fornecer as permissões `kms:Decrypt` para a chave do AWS KMS, adicione o bloco de política a seguir à entidade do IAM:

```
{
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt"
    ],
    "Resource": [
        "arn:aws:kms:Region:account_ID:key/key_ARN"
    ]
```

```
}
```

Caso ainda não tenha feito isso, conclua esse procedimento e adicione permissões `kms:Decrypt` para a chave do AWS KMS.

Use o procedimento a seguir para configurar a função Amazon-GlueServiceRoleForSSM com permissões `kms:Decrypt` para a chave AWS KMS.

Para configurar a função Amazon-GlueServiceRoleForSSM com permissions **`kms:Decrypt`**

1. Abra o console do IAM, em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Roles (Funções) e use o campo de pesquisa para localizar a função Amazon-GlueServiceRoleForSSM. A página Summary é aberta.
3. Use o campo de pesquisa para localizar a função Amazon-GlueServiceRoleForSSM. Selecione o nome de perfil . A página Summary é aberta.
4. Selecione o nome de perfil . A página Summary é aberta.
5. Escolha Add inline policy (Adicionar política em linha). A página Create policy (Criar política) é aberta.
6. Selecione a guia JSON.
7. Exclua o texto JSON existente no editor e, em seguida, copie e cole a seguinte política no editor JSON.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/key_ARN"
            ]
        }
    ]
}
```

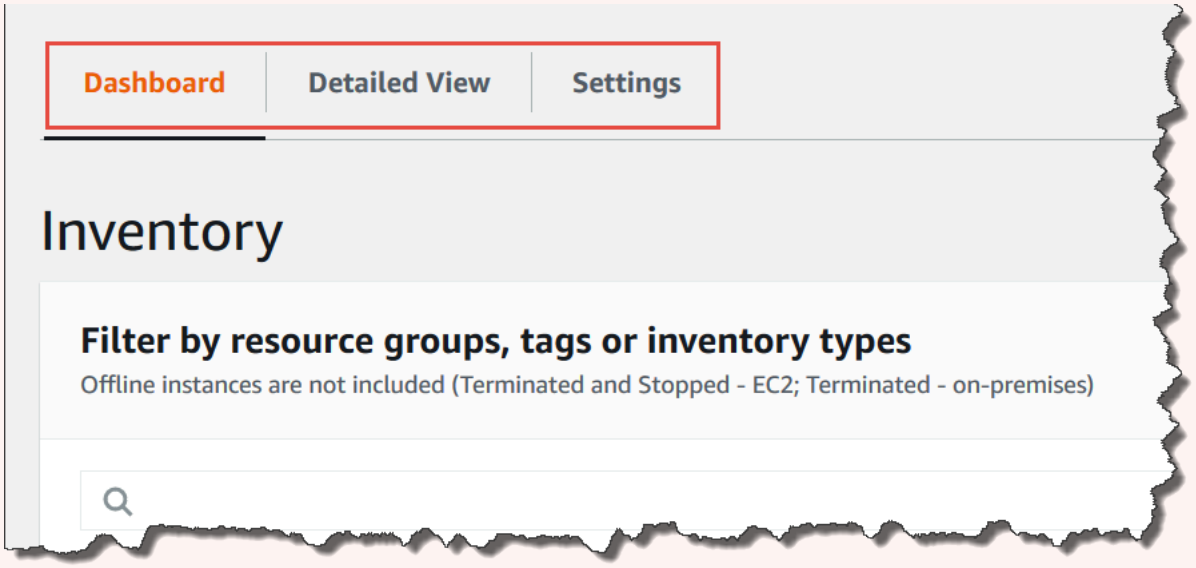
8. Escolha Review policy (Revisar política)
9. Na página Revisar política, insira um nome no campo Nome.
10. Escolha Criar política.

Consultar dados na página de visualização detalhada do inventário

Use o procedimento a seguir para visualizar os dados do inventário de várias Regiões da AWS e Contas da AWS na página Detailed Inventory View (Visualização detalhada do inventário) no Systems Manager Inventory.

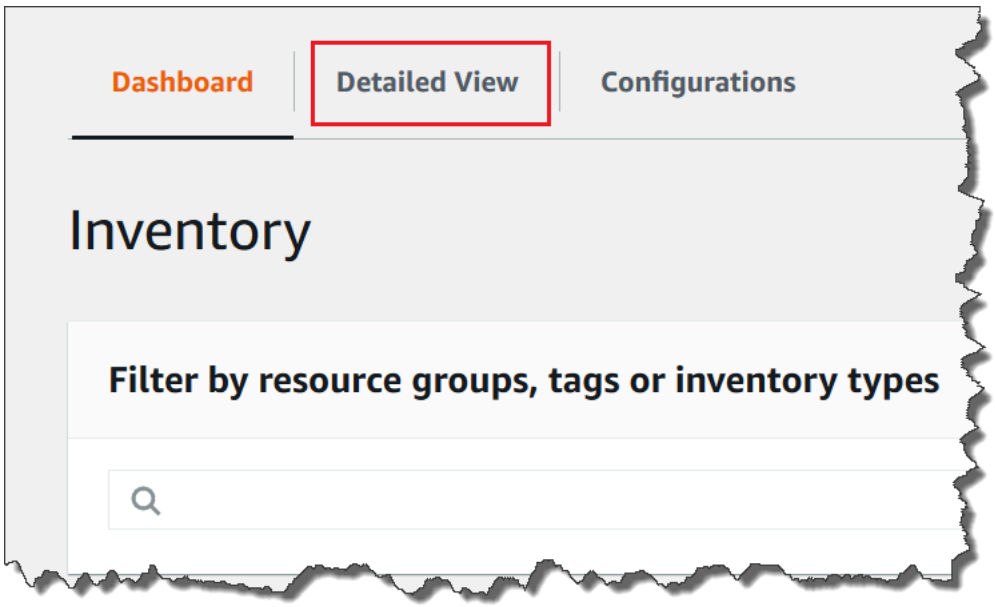
Important

A página Detailed View (Visualização detalhada) do Inventory só está disponível nas Regiões da AWS que oferecem o Amazon Athena. Se as seguintes guias não forem exibidas na página Systems Manager Inventory, isso significa que o Athena não está disponível na região e que você não pode usar a Detailed View (Visualização detalhada) para consultar os dados.

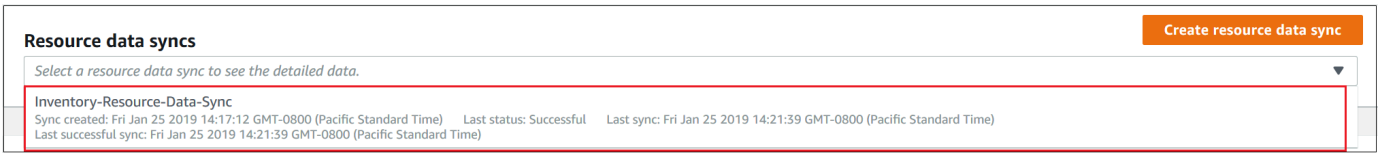


Para visualizar dados de inventário de várias Regiões e contas no console do AWS Systems Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Inventory.
3. Escolha a guia Detailed View (Visualização detalhada).



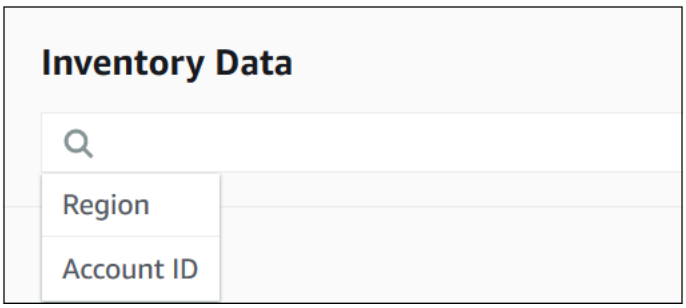
4. Escolha a sincronização de dados de recursos para a qual você deseja consultar dados.



5. Na lista Inventory Type (Tipo de inventário), escolha o tipo de dados de inventário que você deseja consultar e, em seguida, pressione Enter.



6. Para filtrar os dados, selecione a barra Filtro e escolha uma opção de filtro.



Você pode usar o botão Export to CSV (Exportar para CSV) para visualizar a consulta atual definida em um aplicativo de planilha, como o Microsoft Excel. Você também pode usar os botões Query

History (Histórico de consultas) e Run Advanced Queries (Executar consultas avançadas) para visualizar detalhes do histórico e interagir com os dados no Amazon Athena.

Editar a programação do crawler do AWS Glue

Por padrão, o AWS Glue rastreia os dados de inventário no bucket central do Amazon S3 duas vezes por dia. Se você costuma alterar os tipos de dados a serem coletados em seus nós, você pode querer rastrear os dados com mais frequência, conforme descrito no procedimento a seguir.

> **Important**
>
> O AWS Glue cobra sua Conta da AWS com base em uma taxa horária, cobrada por segundo, para crawlers (descoberta de dados) e trabalhos de ETL (processamento e carga de dados). Antes de alterar a programação do crawler, veja a página de definição de preço do AWS Glue.

Para alterar a programação do crawler de dados de inventário

1. Abra o console do AWS Glue em https://console.aws.amazon.com/glue/.
2. No painel de navegação, escolha Rastreadores.
3. Na lista de crawlers, escolha a opção ao lado do crawler de dados do Systems Manager Inventory. O nome do crawler usa o seguinte formato:

   `AWSSystemsManager-`*`s3-bucket-name`*`-`*`Region`*`-`*`account_ID`*
4. Escolha Action (Ação) e, em seguida, escolha Edit crawler (Editar crawler).
5. No painel de navegação, escolha Schedule (Programar).
6. No campo Cron expression (expressão Cron), especifique uma nova programação usando um formato cron. Para obter mais informações sobre o formato cron, consulte Programações baseadas em tempo para trabalhos e crawlers no Manual do desenvolvedor do AWS Glue.

> **Important**
>
> Você pode pausar o crawler para interromper as cobranças do AWS Glue. Se você pausar o crawler ou alterar a frequência para que os dados sejam monitorados com menos frequência, a Detailed View (Visualização detalhada) do Inventory poderá exibir dados não atualizados.

Consultar uma coleta de inventário usando filtros

Depois de coletar dados do inventário, você poderá usar os recursos de filtro no AWS Systems Manager para consultar uma lista de nós gerenciados que atendam a certos critérios de filtro.

Para consultar nós baseados em filtros de inventário

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Inventory.
3. Na seção Filter by resource groups, tags or inventory types (Filtrar por grupos de recursos, tags ou tipos de inventários), escolha a caixa de filtro. Uma lista de filtros predefinidos será exibida.
4. Escolha um atributo para filtrar. Para este exemplo, selecione **`AWS:Application`**. Se solicitado, escolha um atributo secundário para filtrar. Para este exemplo, selecione **`AWS:Application.Name`**.
5. Escolha um delimitador da lista. Por exemplo, escolha Begin with. Uma caixa de texto é exibida no filtro.
6. Insira um valor nessa caixa de texto. Por exemplo, digite Amazon (o SSM Agent é chamado de Amazon SSM Agent).
7. Pressione Enter. O sistema retornará uma lista de nós gerenciados que incluem um nome de aplicação que comece com a palavra Amazon.

> **Note**
>
> É possível combinar vários filtros para refinar sua pesquisa.

Agregar dados do inventário

Depois de configurar os nós gerenciados para o AWS Systems Manager Inventory, você poderá visualizar as contagens agregadas dos dados do inventário. Por exemplo, digamos que você configurou dezenas ou centenas de nós gerenciados para coletar o tipo de inventário da `AWS:Application`. Usando as informações desta seção, você pode ver uma contagem exata de quantos estão configurados para coletar esses dados.

Você também pode ver detalhes específicos do inventário por meio da agregação por tipo de dados. Por exemplo, o tipo de inventário `AWS:InstanceInformation` coleta informações da plataforma

do sistema operacional com o tipo de dados `Platform`. Agregando dados pelo tipo de dados da `Platform`, você pode ver rapidamente quantos nós estão executando o Windows e quantos estão executando o macOS.

Os procedimentos desta seção descrevem como visualizar as contagens agregadas de dados do inventário usando a AWS Command Line Interface (AWS CLI). Você também pode visualizar contagens agregadas pré-configuradas no console do AWS Systems Manager na página Inventory. Esses painéis pré-configurados são chamados de Inventory Insights (Visões do inventário) e oferecem correção com um clique dos problemas de configuração do inventário.

Observe os seguintes detalhes importantes sobre as contagens de agregação de dados do inventário:

- Se você encerrar um nó gerenciado configurado para coletar dados de inventário, o Systems Manager reterá os dados do inventário por 30 dias e depois os excluirá. Para nós em execução, os dados de inventário com mais de 30 dias são excluídos pelos sistemas. Se você precisar armazenar dados do inventário por mais de 30 dias, você poderá usar o AWS Config para registrar o histórico ou consultar e fazer upload dos dados periodicamente em um bucket do Amazon Simple Storage Service (Amazon S3).
- Se um nó foi configurado anteriormente para relatar um determinado tipo de dados do inventário, por exemplo, `AWS:Network`, e você alterar a configuração mais tarde para interromper a coleta desse tipo, as contagens de agregação continuará a mostrar os dados do `AWS:Network` até que o nó seja encerrado, e 30 dias tenham passado.

Para obter informações sobre como configurar rapidamente e coletar dados do inventário de todos os nós em uma determinada Conta da AWS (e de qualquer nó futuro que possa ser criado nessa conta), consulte Inventário de todos os nós gerenciados na sua Conta da AWS.

Tópicos

- Agregar dados do inventário para ver a contagem de nós gerenciados que coletam tipos de dados específicos
- Agregar dados do inventário com grupos para ver quais nós gerenciados estão e quais não estão configurados para coletar um tipo de inventário

Agregar dados do inventário para ver a contagem de nós gerenciados que coletam tipos de dados específicos

Use a ação da API GetInventory do AWS Systems Manager para visualizar as contagens agregadas dos nós que coletam um ou mais tipos de inventário e tipos de dados. Por exemplo, o tipo de inventário `AWS:InstanceInformation` permite visualizar um agregado de sistemas operacionais usando a operação da API GetInventory com o tipo de dados `AWS:InstanceInformation.PlatformType`. Este é um exemplo do comando e saída da AWS CLI:

```
aws ssm get-inventory --aggregators "Expression=AWS:InstanceInformation.PlatformType"
```

O sistema retorna informações como estas.

```
{
   "Entities":[
      {
         "Data":{
            "AWS:InstanceInformation":{
               "Content":[
                  {
                     "Count":"7",
                     "PlatformType":"windows"
                  },
                  {
                     "Count":"5",
                     "PlatformType":"linux"
                  }
               ]
            }
         }
      }
   ]
}
```

Conceitos básicos

Determine os tipos de inventário e os tipos de dados dos quais você deseja visualizar contagens. É possível visualizar uma lista de tipos de inventário e tipos de dados compatíveis com a agregação executando o comando a seguir na AWS CLI.

```
aws ssm get-inventory-schema --aggregator
```

O comando retorna uma lista JSON de tipos de inventário e tipos de dados compatíveis com a agregação. O campo TypeName mostra os tipos de inventário compatíveis. E o campo Name (Nome) mostra cada tipo de dados. Por exemplo, na lista a seguir, o tipo de inventário `AWS:Application` inclui tipos de dados para `Name` e `Version`.

```
{
    "Schemas": [
        {
            "TypeName": "AWS:Application",
            "Version": "1.1",
            "DisplayName": "Application",
            "Attributes": [
                {
                    "DataType": "STRING",
                    "Name": "Name"
                },
                {
                    "DataType": "STRING",
                    "Name": "Version"
                }
            ]
        },
        {
            "TypeName": "AWS:InstanceInformation",
            "Version": "1.0",
            "DisplayName": "Platform",
            "Attributes": [
                {
                    "DataType": "STRING",
                    "Name": "PlatformName"
                },
                {
                    "DataType": "STRING",
                    "Name": "PlatformType"
                },
                {
                    "DataType": "STRING",
                    "Name": "PlatformVersion"
                }
            ]
```

```
        },
        {
            "TypeName": "AWS:ResourceGroup",
            "Version": "1.0",
            "DisplayName": "ResourceGroup",
            "Attributes": [
                {
                    "DataType": "STRING",
                    "Name": "Name"
                }
            ]
        },
        {
            "TypeName": "AWS:Service",
            "Version": "1.0",
            "DisplayName": "Service",
            "Attributes": [
                {
                    "DataType": "STRING",
                    "Name": "Name"
                },
                {
                    "DataType": "STRING",
                    "Name": "DisplayName"
                },
                {
                    "DataType": "STRING",
                    "Name": "ServiceType"
                },
                {
                    "DataType": "STRING",
                    "Name": "Status"
                },
                {
                    "DataType": "STRING",
                    "Name": "StartType"
                }
            ]
        },
        {
            "TypeName": "AWS:WindowsRole",
            "Version": "1.0",
            "DisplayName": "WindowsRole",
            "Attributes": [
```

```
                {
                    "DataType": "STRING",
                    "Name": "Name"
                },
                {
                    "DataType": "STRING",
                    "Name": "DisplayName"
                },
                {
                    "DataType": "STRING",
                    "Name": "FeatureType"
                },
                {
                    "DataType": "STRING",
                    "Name": "Installed"
                }
            ]
        }
    ]
}
```

Você pode agregar dados para qualquer um dos tipos de inventário listados criando um comando que use a seguinte sintaxe:

```
aws ssm get-inventory --aggregators "Expression=InventoryType.DataType"
```

Aqui estão alguns exemplos.

Exemplo 1

Este exemplo agrega uma contagem das funções do Windows usadas por seus nós.

```
aws ssm get-inventory --aggregators "Expression=AWS:WindowsRole.Name"
```

Exemplo 2

Este exemplo agrega uma contagem das aplicações instaladas em seus nós.

```
aws ssm get-inventory --aggregators "Expression=AWS:Application.Name"
```

Combinar vários agregadores

Também é possível combinar vários tipos de inventário e tipos de dados em um comando para ajudar a entender melhor os dados. Aqui estão alguns exemplos.

Exemplo 1

Este exemplo agrega uma contagem dos tipos de sistema operacional usados pelos nós. Ele também retorna o nome específico dos sistemas operacionais.

```
aws ssm get-inventory --aggregators '[{"Expression":
 "AWS:InstanceInformation.PlatformType", "Aggregators":[{"Expression":
 "AWS:InstanceInformation.PlatformName"}]}]'
```

Exemplo 2

Este exemplo agrega uma contagem das aplicações em execução em seus nós e a versão específica de cada uma.

```
aws ssm get-inventory --aggregators '[{"Expression": "AWS:Application.Name",
 "Aggregators":[{"Expression": "AWS:Application.Version"}]}]'
```

Se preferir, você pode criar uma expressão de agregação com um ou mais tipos de inventário e tipos de dados em um arquivo JSON e chamar o arquivo na AWS CLI. O JSON no arquivo deve usar a seguinte sintaxe:

```
[
       {
           "Expression": "string",
           "Aggregators": [
               {
                  "Expression": "string"
               }
           ]
       }
]
```

Você deve salvar o arquivo com a extensão .json.

Veja a seguir um exemplo que usa vários tipos de inventário e tipos de dados.

```
[
```

```
        {
            "Expression": "AWS:Application.Name",
            "Aggregators": [
                {
                    "Expression": "AWS:Application.Version",
                    "Aggregators": [
                      {
                      "Expression": "AWS:InstanceInformation.PlatformType"
                      }
                    ]
                }
            ]
        }
]
```

Use o comando a seguir para chamar o arquivo na AWS CLI.

```
aws ssm get-inventory --aggregators file://file_name.json
```

O comando retorna informações como as seguintes.

```
{"Entities":
 [
   {"Data":
     {"AWS:Application":
       {"Content":
         [
           {"Count": "3",
            "PlatformType": "linux",
            "Version": "2.6.5",
            "Name": "audit-libs"},
           {"Count": "2",
            "PlatformType": "windows",
            "Version": "2.6.5",
            "Name": "audit-libs"},
           {"Count": "4",
            "PlatformType": "windows",
            "Version": "6.2.8",
            "Name": "microsoft office"},
           {"Count": "2",
            "PlatformType": "windows",
            "Version": "2.6.5",
            "Name": "chrome"},
```

```
            {"Count": "1",
             "PlatformType": "linux",
             "Version": "2.6.5",
             "Name": "chrome"},
            {"Count": "2",
             "PlatformType": "linux",
             "Version": "6.3",
             "Name": "authconfig"}
          ]
        }
     },
    "ResourceType": "ManagedInstance"}
 ]
}
```

Agregar dados do inventário com grupos para ver quais nós gerenciados estão e quais não estão configurados para coletar um tipo de inventário

Os grupos do Systems Manager Inventory permitem que você veja rapidamente uma contagem de quais nós gerenciados estão e quais não estão configurados para coletar um ou mais tipos de inventário. Com grupos, você especifica um ou mais tipos de inventário e um filtro que usa o operador `exists`.

Por exemplo, digamos que você tenha quatro nós gerenciados configurados para coletar os seguintes tipos de inventário:

- Nó 1: `AWS:Application`
- Nó 2: `AWS:File`
- Nó 3: `AWS:Application`, `AWS:File`
- Nó 4: `AWS:Network`

Você pode executar o seguinte comando na AWS CLI para ver quantos nós estão configurados para coletar os tipos de inventário `AWS:Application` e `AWS:File inventory`. A resposta também retorna uma contagem dos nós que não estão configurados para coletar esses dois tipos de inventário.

```
aws ssm get-inventory --aggregators
 'Groups=[{Name=ApplicationAndFile,Filters=[{Key=TypeName,Values=[AWS:Application],Type=Exists}
{Key=TypeName,Values=[AWS:File],Type=Exists}]}]'
```

A resposta do comando mostra que apenas um nó gerenciado está configurado para coletar os tipos de inventário `AWS:Application` e `AWS:File`.

```
{
   "Entities":[
      {
         "Data":{
            "ApplicationAndFile":{
               "Content":[
                  {
                     "notMatchingCount":"3"
                  },
                  {
                     "matchingCount":"1"
                  }
               ]
            }
         }
      }
   ]
}
```

> **Note**
>
> Os grupos não retornam contagens de tipo de dados. Além disso, você não pode detalhar os resultados para ver os IDs dos nós gerenciados que estiverem ou não configurados para coletar o tipo de inventário.

Se preferir, crie uma expressão de agregação com um ou mais tipos de inventário em um arquivo JSON e chame o arquivo na AWS CLI. O JSON no arquivo deve usar a seguinte sintaxe:

```
{
   "Aggregators":[
      {
         "Groups":[
            {
               "Name":"Name",
               "Filters":[
                  {
                     "Key":"TypeName",
```

```
                    "Values":[
                       "Inventory_type"
                    ],
                    "Type":"Exists"
                 },
                 {
                    "Key":"TypeName",
                    "Values":[
                       "Inventory_type"
                    ],
                    "Type":"Exists"
                 }
              ]
           }
        ]
     }
  ]
}
```

Você deve salvar o arquivo com a extensão .json.

Use o comando a seguir para chamar o arquivo na AWS CLI.

```
aws ssm get-inventory --cli-input-json file://file_name.json
```

Exemplos adicionais

Os exemplos a seguir mostram como agregar dados de inventário para ver quais nós gerenciados estão e não estão configurados para coletar os tipos de inventário especificados. Estes exemplos usam a AWS CLI. Cada exemplo inclui um comando completo com filtros que você pode executar na linha de comando e um arquivo input.json de exemplo se você preferir inserir as informações em um arquivo.

Exemplo 1

Este exemplo agrega uma contagem de nós que estão e que não estão configurados para coletar os tipos de inventário `AWS:Application` ou `AWS:File`.

Execute o seguinte comando na AWS CLI.

```
aws ssm get-inventory --aggregators
 'Groups=[{Name=ApplicationORFile,Filters=[{Key=TypeName,Values=[AWS:Application,
 AWS:File],Type=Exists}]}]'
```

Se preferir usar um arquivo, copie e cole o seguinte exemplo em um arquivo e salve-o como input.json.

```
{
   "Aggregators":[
      {
         "Groups":[
            {
               "Name":"ApplicationORFile",
               "Filters":[
                  {
                     "Key":"TypeName",
                     "Values":[
                        "AWS:Application",
                        "AWS:File"
                     ],
                     "Type":"Exists"
                  }
               ]
            }
         ]
      }
   ]
}
```

Execute o seguinte comando na AWS CLI.

```
aws ssm get-inventory --cli-input-json file://input.json
```

O comando retorna informações como as seguintes.

```
{
   "Entities":[
      {
         "Data":{
            "ApplicationORFile":{
               "Content":[
                  {
```

```
                    "notMatchingCount":"1"
                 },
                 {
                    "matchingCount":"3"
                 }
              ]
           }
        }
     }
  ]
}
```

Exemplo 2

Este exemplo agrega uma contagem de nós que estão e que não estão configurados para coletar os tipos de inventário `AWS:Application`, `AWS:File` e `AWS:Network`.

Execute o seguinte comando na AWS CLI.

```
aws ssm get-inventory --aggregators
 'Groups=[{Name=Application,Filters=[{Key=TypeName,Values=[AWS:Application],Type=Exists}]},
 {Name=File,Filters=[{Key=TypeName,Values=[AWS:File],Type=Exists}]},
 {Name=Network,Filters=[{Key=TypeName,Values=[AWS:Network],Type=Exists}]}]'
```

Se preferir usar um arquivo, copie e cole o seguinte exemplo em um arquivo e salve-o como input.json.

```
{
   "Aggregators":[
      {
         "Groups":[
            {
               "Name":"Application",
               "Filters":[
                  {
                     "Key":"TypeName",
                     "Values":[
                        "AWS:Application"
                     ],
                     "Type":"Exists"
                  }
               ]
            },
```

```
            {
               "Name":"File",
               "Filters":[
                  {
                     "Key":"TypeName",
                     "Values":[
                        "AWS:File"
                     ],
                     "Type":"Exists"
                  }
               ]
            },
            {
               "Name":"Network",
               "Filters":[
                  {
                     "Key":"TypeName",
                     "Values":[
                        "AWS:Network"
                     ],
                     "Type":"Exists"
                  }
               ]
            }
         ]
      }
   ]
}
```

Execute o seguinte comando na AWS CLI.

```
aws ssm get-inventory --cli-input-json file://input.json
```

O comando retorna informações como as seguintes.

```
{
   "Entities":[
      {
         "Data":{
            "Application":{
               "Content":[
                  {
                     "notMatchingCount":"2"
```

```
                  },
                  {
                     "matchingCount":"2"
                  }
               ]
            },
            "File":{
               "Content":[
                  {
                     "notMatchingCount":"2"
                  },
                  {
                     "matchingCount":"2"
                  }
               ]
            },
            "Network":{
               "Content":[
                  {
                     "notMatchingCount":"3"
                  },
                  {
                     "matchingCount":"1"
                  }
               ]
            }
         }
      }
   ]
}
```

Trabalhar com inventário personalizado

Você pode atribuir quaisquer metadados desejados aos nós criando um inventário personalizado do AWS Systems Manager Inventory. Por exemplo, digamos que você gerencie um grande número de servidores em racks no seu datacenter e esses servidores tenham sido configurados como nós gerenciados do Systems Manager. Atualmente, você armazena informações sobre a localização de racks de servidor em uma planilha. Com o inventário personalizado, você pode especificar a localização do rack de cada nó como um metadado para o nó. Quando você coleta o inventário usando o Systems Manager, os metadados são coletados com outros metadados do inventário. Você pode então fazer a portabilidade de todos os metadados do inventário para um bucket central do Amazon S3 usando a sincronização de dados de recursos e consultando os dados.

> **Note**
>
> O Systems Manager oferece suporte a no máximo 20 tipos de inventário personalizados por Conta da AWS.

Para atribuir um inventário personalizado a um nó, você pode usar a operação da API PutInventory do Systems Manager, conforme descrito em Assigning custom inventory metadata to a managed node. Como alternativa, você pode criar um arquivo JSON de inventário personalizado e enviá-lo para o nó. Esta seção descreve como criar o arquivo JSON.

O seguinte arquivo JSON de exemplo com inventário personalizado especifica informações de rack sobre um servidor on-premises. Esse exemplo especifica um tipo de dados de inventário personalizado (`"TypeName": "Custom:RackInformation"`), com várias entradas em `Content` que descrevem os dados.

```
{
    "SchemaVersion": "1.0",
    "TypeName": "Custom:RackInformation",
    "Content": {
        "Location": "US-EAST-02.CMH.RACK1",
        "InstalledTime": "2016-01-01T01:01:01Z",
        "vendor": "DELL",
        "Zone" : "BJS12",
        "TimeZone": "UTC-8"
      }
 }
```

Você também pode especificar entradas distintas na seção `Content`, conforme mostrado no exemplo a seguir.

```
{
"SchemaVersion": "1.0",
"TypeName": "Custom:PuppetModuleInfo",
    "Content": [{
        "Name": "puppetlabs/aws",
        "Version": "1.0"
      },
      {
        "Name": "puppetlabs/dsc",
```

```
        "Version": "2.0"
      }
    ]
}
```

O esquema JSON para inventário personalizado requer as seções `SchemaVersion`, `TypeName` e `Content`, mas você pode definir as informações nessas seções.

```
{
    "SchemaVersion": "user_defined",
    "TypeName": "Custom:user_defined",
    "Content": {
        "user_defined_attribute1": "user_defined_value1",
        "user_defined_attribute2": "user_defined_value2",
        "user_defined_attribute3": "user_defined_value3",
        "user_defined_attribute4": "user_defined_value4"
      }
 }
```

O valor de `TypeName` é limitado a 100 caracteres. Além disso, o valor `TypeName` deve começar com a palavra `Custom` em maiúsculas. Por exemplo, .`Custom:PuppetModuleInfo` Portanto, os exemplos a seguir resultariam em uma exceção: `CUSTOM:PuppetModuleInfo`, `custom:PuppetModuleInfo`.

A seção `Content` inclui atributos e *`dados`*. Esses itens não diferenciam entre maiúsculas e minúsculas. No entanto, se você definir um atributo (por exemplo: "`Vendor`": "DELL"), deverá sempre fazer referência a esse atributo em seus arquivos de inventário personalizados. Se você especificar "`Vendor`": "DELL" (usando um "V" maiúsculo em `vendor`) em um arquivo e, em seguida, especificar "`vendor`": "DELL" (usando um "v" minúsculo em `vendor`) em outro arquivo, o sistema retornará um erro.

> **Note**
>
> Você deve salvar o arquivo com uma extensão `.json`, e o inventário definido deve consistir somente em valores de string.

Depois de criar o arquivo, salve-o em seu nó. A tabela a seguir mostra o local em que os arquivos JSON do inventário personalizado devem ser armazenados em seu nó:

Sistema operacional	Path
Linux	/var/lib/amazon/ssm/*node-id*/inventory/custom
macOS	`/opt/aws/ssm/data/ node-id/ inventory/custom`
Windows	%SystemDrive%\ProgramData\Amazon\SSM \InstanceData*node-id*\inventory\custom

Para obter um exemplo de como usar o inventário personalizado, consulte o tópico sobre Utilização de disco da sua frota usando tipos de inventário personalizados do EC2 Systems Manager.

Excluir inventário personalizado

Você pode usar a operação da API DeleteInventory para excluir um tipo de inventário personalizado e os dados associados a esse tipo. Chame o comando delete-inventory usando a AWS Command Line Interface (AWS CLI) para excluir todos os dados de um tipo de inventário. Você chama o comando delete-inventory com `SchemaDeleteOption` para excluir um tipo de inventário personalizado.

> **Note**
>
> Um tipo de inventário também é chamado de esquema de inventário.

O parâmetro `SchemaDeleteOption` inclui as opções a seguir:

- DeleteSchema: esta opção exclui o tipo personalizado especificado e todos os dados associados a ele. Se desejar, você pode recriar o esquema mais tarde.
- DisableSchema: se você escolher essa opção, o sistema desativará a versão atual, excluirá todos os dados dela e ignorará todos os novos dados se a versão for anterior ou igual à versão desativada. Você pode permitir esse tipo de inventário novamente chamando a ação PutInventory para uma versão posterior à versão desativada.

Para excluir ou desativar o inventário personalizado usando a AWS CLI

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Execute o seguinte comando para usar a opção `dry-run` para ver quais dados serão excluídos do sistema. Esse comando não exclui nenhum dado.

```
aws ssm delete-inventory --type-name "Custom:custom_type_name" --dry-run
```

O sistema retorna informações como estas.

```
{
   "DeletionSummary":{
      "RemainingCount":3,
      "SummaryItems":[
         {
            "Count":2,
            "RemainingCount":2,
            "Version":"1.0"
         },
         {
            "Count":1,
            "RemainingCount":1,
            "Version":"2.0"
         }
      ],
      "TotalCount":3
   },
   "TypeName":"Custom:custom_type_name"
}
```

Para obter informações sobre como entender o resumo de exclusão do inventário, consulte Noções básicas sobre o resumo de exclusão do inventário.

3. Execute o seguinte comando para excluir todos os dados de um tipo de inventário personalizado.

```
aws ssm delete-inventory --type-name "Custom:custom_type_name"
```

> **Note**
>
> O resultado desse comando não mostra o andamento da exclusão. Por esse motivo, TotalCount (Total) e Remaining Count (Restante) são sempre os mesmos porque

o sistema ainda não excluiu nada. Você pode usar o comando "describe-inventory-deletions" para mostrar o andamento da exclusão, conforme descrito mais adiante neste tópico.

O sistema retorna informações como estas.

```
{
   "DeletionId":"system_generated_deletion_ID",
   "DeletionSummary":{
      "RemainingCount":3,
      "SummaryItems":[
         {
            "Count":2,
            "RemainingCount":2,
            "Version":"1.0"
         },
         {
            "Count":1,
            "RemainingCount":1,
            "Version":"2.0"
         }
      ],
      "TotalCount":3
   },
   "TypeName":"custom_type_name"
}
```

O sistema exclui todos os dados do tipo de inventário personalizado especificado no serviço do Systems Manager Inventory.

4. Execute o seguinte comando: O comando executa as seguintes ações para a versão atual do tipo de inventário: desativa a versão atual, exclui todos os dados dela e ignora todos os novos dados se a versão for anterior ou igual à versão desativada.

```
aws ssm delete-inventory --type-name "Custom:custom_type_name" --schema-delete-
option "DisableSchema"
```

O sistema retorna informações como estas.

```
{
```

```
   "DeletionId":"system_generated_deletion_ID",
   "DeletionSummary":{
      "RemainingCount":3,
      "SummaryItems":[
         {
            "Count":2,
            "RemainingCount":2,
            "Version":"1.0"
         },
         {
            "Count":1,
            "RemainingCount":1,
            "Version":"2.0"
         }
      ],
      "TotalCount":3
   },
   "TypeName":"Custom:custom_type_name"
}
```

Você pode visualizar um tipo de inventário desativado, usando o comando a seguir.

```
aws ssm get-inventory-schema --type-name Custom:custom_type_name
```

5. Execute o seguinte comando para excluir um tipo de inventário.

```
aws ssm delete-inventory --type-name "Custom:custom_type_name" --schema-delete-
option "DeleteSchema"
```

O sistema exclui o esquema e todos os dados de inventário do tipo personalizado especificado.

O sistema retorna informações como estas.

```
{
   "DeletionId":"system_generated_deletion_ID",
   "DeletionSummary":{
      "RemainingCount":3,
      "SummaryItems":[
         {
            "Count":2,
            "RemainingCount":2,
            "Version":"1.0"
```

```
         },
         {
            "Count":1,
            "RemainingCount":1,
            "Version":"2.0"
         }
      ],
      "TotalCount":3
   },
   "TypeName":"Custom:custom_type_name"
}
```

Visualizar o status da exclusão

Você pode verificar o status de uma operação de exclusão usando o comando `describe-inventory-deletions` da AWS CLI. Você pode especificar um ID de exclusão para visualizar o status de uma determinada operação de exclusão. Ou você pode omitir o ID de exclusão para visualizar uma lista de todas as exclusões executadas nos últimos 30 dias.

1. Execute o seguinte comando para visualizar o status de uma operação de exclusão. O sistema retornou o ID de exclusão no resumo de exclusão do inventário.

   ```
   aws ssm describe-inventory-deletions --deletion-id system_generated_deletion_ID
   ```

   O sistema retorna o status mais recente. A operação de exclusão pode não estar concluída. O sistema retorna informações como estas.

   ```
   {"InventoryDeletions":
     [
       {"DeletionId": "system_generated_deletion_ID",
        "DeletionStartTime": 1521744844,
        "DeletionSummary":
         {"RemainingCount": 1,
          "SummaryItems":
           [
             {"Count": 1,
              "RemainingCount": 1,
              "Version": "1.0"}
           ],
          "TotalCount": 1},
   ```

```
     "LastStatus": "InProgress",
     "LastStatusMessage": "The Delete is in progress",
     "LastStatusUpdateTime": 1521744844,
     "TypeName": "Custom:custom_type_name"}
  ]
}
```

Se a operação de exclusão for bem-sucedida, a mensagem `LastStatusMessage` indica: Exclusão bem-sucedida.

```
{"InventoryDeletions":
  [
    {"DeletionId": "system_generated_deletion_ID",
     "DeletionStartTime": 1521744844,
     "DeletionSummary":
      {"RemainingCount": 0,
       "SummaryItems":
        [
          {"Count": 1,
           "RemainingCount": 0,
           "Version": "1.0"}
        ],
       "TotalCount": 1},
     "LastStatus": "Complete",
     "LastStatusMessage": "Deletion is successful",
     "LastStatusUpdateTime": 1521745253,
     "TypeName": "Custom:custom_type_name"}
  ]
}
```

2. Execute o seguinte comando para visualizar uma lista de todas as exclusões executadas nos últimos 30 dias.

```
aws ssm describe-inventory-deletions --max-results a number
```

```
{"InventoryDeletions":
  [
    {"DeletionId": "system_generated_deletion_ID",
     "DeletionStartTime": 1521682552,
     "DeletionSummary":
      {"RemainingCount": 0,
```

```
      "SummaryItems":
       [
         {"Count": 1,
          "RemainingCount": 0,
          "Version": "1.0"}
       ],
      "TotalCount": 1},
    "LastStatus": "Complete",
    "LastStatusMessage": "Deletion is successful",
    "LastStatusUpdateTime": 1521682852,
    "TypeName": "Custom:custom_type_name"},
   {"DeletionId": "system_generated_deletion_ID",
    "DeletionStartTime": 1521744844,
    "DeletionSummary":
     {"RemainingCount": 0,
      "SummaryItems":
       [
         {"Count": 1,
          "RemainingCount": 0,
          "Version": "1.0"}
       ],
      "TotalCount": 1},
    "LastStatus": "Complete",
    "LastStatusMessage": "Deletion is successful",
    "LastStatusUpdateTime": 1521745253,
    "TypeName": "Custom:custom_type_name"},
   {"DeletionId": "system_generated_deletion_ID",
    "DeletionStartTime": 1521680145,
    "DeletionSummary":
     {"RemainingCount": 0,
      "SummaryItems":
       [
         {"Count": 1,
          "RemainingCount": 0,
          "Version": "1.0"}
       ],
      "TotalCount": 1},
    "LastStatus": "Complete",
    "LastStatusMessage": "Deletion is successful",
    "LastStatusUpdateTime": 1521680471,
    "TypeName": "Custom:custom_type_name"}
 ],
"NextToken": "next-token"
```

Noções básicas sobre o resumo de exclusão do inventário

Para ajudar você a compreender o conteúdo do resumo de exclusão do inventário, considere o exemplo a seguir. Um usuário atribuiu o inventário Custom:RackSpace a três nós. Os itens de inventário 1 e 2 usam o tipo personalizado versão 1.0 ("SchemaVersion":"1.0"). O item de inventário 3 usa o tipo personalizado versão 2.0 ("SchemaVersion":"2.0").

Inventário personalizado do RackSpace 1

```
{
   "CaptureTime":"2018-02-19T10:48:55Z",
   "TypeName":"CustomType:RackSpace",
   "InstanceId":"i-1234567890",
   "SchemaVersion":"1.0"   "Content":[
      {
         content of custom type omitted
      }
   ]
}
```

Inventário personalizado do RackSpace 2

```
{
   "CaptureTime":"2018-02-19T10:48:55Z",
   "TypeName":"CustomType:RackSpace",
   "InstanceId":"i-1234567891",
   "SchemaVersion":"1.0"   "Content":[
      {
         content of custom type omitted
      }
   ]
}
```

Inventário personalizado do RackSpace 3

```
{
   "CaptureTime":"2018-02-19T10:48:55Z",
   "TypeName":"CustomType:RackSpace",
   "InstanceId":"i-1234567892",
   "SchemaVersion":"2.0"   "Content":[
      {
```

```
         content of custom type omitted
      }
   ]
}
```

O usuário executa o comando a seguir para visualizar quais dados serão excluídos.

```
aws ssm delete-inventory --type-name "Custom:RackSpace" --dry-run
```

O sistema retorna informações como estas.

```
{
   "DeletionId":"1111-2222-333-444-66666",
   "DeletionSummary":{
      "RemainingCount":3,
      "TotalCount":3,
                TotalCount and RemainingCount are the number of items that would be
 deleted if this was not a dry run. These numbers are the same because the system
 didn't delete anything.
      "SummaryItems":[
         {
            "Count":2,             The system found two items that use SchemaVersion
 1.0. Neither item was deleted.
            "RemainingCount":2,
            "Version":"1.0"
         },
         {
            "Count":1,             The system found one item that uses SchemaVersion
 1.0. This item was not deleted.
            "RemainingCount":1,
            "Version":"2.0"
         }
      ],

   },
   "TypeName":"Custom:RackSpace"
}
```

O usuário executa o comando a seguir para excluir o inventário Custom:RackSpace.

> **Note**
>
> O resultado desse comando não mostra o andamento da exclusão. Por esse motivo, `TotalCount` e `RemainingCount` são sempre os mesmos porque o sistema ainda não excluiu nada. Você pode usar o comando `describe-inventory-deletions` para mostrar o andamento da exclusão.

```
aws ssm delete-inventory --type-name "Custom:RackSpace"
```

O sistema retorna informações como estas.

```
{
   "DeletionId":"1111-2222-333-444-7777777",
   "DeletionSummary":{
      "RemainingCount":3,           There are three items to delete
      "SummaryItems":[
         {
            "Count":2,              The system found two items that use SchemaVersion
 1.0.
            "RemainingCount":2,
            "Version":"1.0"
         },
         {
            "Count":1,              The system found one item that uses SchemaVersion
 2.0.
            "RemainingCount":1,
            "Version":"2.0"
         }
      ],
      "TotalCount":3
   },
   "TypeName":"RackSpace"
}
```

Visualizar ações de exclusão de inventário no EventBridge

Você pode configurar o Amazon EventBridge para criar um evento sempre que um usuário excluir o inventário personalizado. O EventBridge oferece três tipos de eventos para operações de exclusão do inventário personalizado:

- Excluir ação de uma instância: se o inventário personalizado de um nó gerenciado específico foi excluído com êxito ou não.
- Excluir resumo da ação: Um resumo da ação de exclusão.
- Aviso para tipo de inventário personalizado desativado: um evento de aviso se um usuário chamou a operação de API PutInventory para uma versão do tipo de inventário personalizado que foi desativada anteriormente.

Veja exemplos de cada evento:

Delete action for an instance (Excluir ação de uma instância)


```
{
   "version":"0",
   "id":"998c9cde-56c0-b38b-707f-0411b3ff9d11",
   "detail-type":"Inventory Resource State Change",
   "source":"aws.ssm",
   "account":"478678815555",
   "time":"2018-05-24T22:24:34Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ssm:us-east-1:478678815555:managed-instance/i-0a5feb270fc3f0b97"
   ],
   "detail":{
      "action-status":"succeeded",
      "action":"delete",
      "resource-type":"managed-instance",
      "resource-id":"i-0a5feb270fc3f0b97",
      "action-reason":"",
      "type-name":"Custom:MyInfo"
   }
}
```


Delete action summary (Excluir resumo da ação)


```
{
   "version":"0",
   "id":"83898300-f576-5181-7a67-fb3e45e4fad4",
   "detail-type":"Inventory Resource State Change",
   "source":"aws.ssm",
   "account":"478678815555",
   "time":"2018-05-24T22:28:25Z",
```

```
   "region":"us-east-1",
   "resources":[

   ],
   "detail":{
      "action-status":"succeeded",
      "action":"delete-summary",
      "resource-type":"managed-instance",
      "resource-id":"",
      "action-reason":"The delete for type name Custom:MyInfo was completed. The
 deletion summary is: {\"totalCount\":2,\"remainingCount\":0,\"summaryItems\":
[{\"version\":\"1.0\",\"count\":2,\"remainingCount\":0}]}",
      "type-name":"Custom:MyInfo"
   }
}
```

Aviso para tipo de inventário personalizado desativado

```
{
   "version":"0",
   "id":"49c1855c-9c57-b5d7-8518-b64aeeef5e4a",
   "detail-type":"Inventory Resource State Change",
   "source":"aws.ssm",
   "account":"478678815555",
   "time":"2018-05-24T22:46:58Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ssm:us-east-1:478678815555:managed-instance/i-0ee2d86a2cfc371f6"
   ],
   "detail":{
      "action-status":"failed",
      "action":"put",
      "resource-type":"managed-instance",
      "resource-id":"i-0ee2d86a2cfc371f6",
      "action-reason":"The inventory item with type name Custom:MyInfo was sent with a
 disabled schema version 1.0. You must send a version greater than 1.0",
      "type-name":"Custom:MyInfo"
   }
}
```

Use o procedimento a seguir para criar uma regra do EventBridge para operações de exclusão de inventário personalizadas. Este procedimento mostra como criar uma regra que envia notificações para excluir operações de inventário personalizadas em um tópico do Amazon SNS. Antes de

começar, verifique se você tem um tópico do Amazon SNS ou crie um novo. Para obter mais informações, consulte Conceitos básicos no Guia do Desenvolvedor do Amazon Simple Notification Service.

Para configurar o EventBridge para excluir operações de inventário

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Regras.
3. Escolha Create rule.
4. Insira um nome e uma descrição para a regra.

   Uma regra não pode ter o mesmo nome que outra na mesma Região e barramento de eventos.
5. Em Barramento de eventos, selecione o barramento de eventos que você deseja associar a essa regra. Se você quiser que essa regra responda a eventos correspondentes provenientes da sua Conta da AWS, selecione default (padrão). Quando um AWS service (Serviço da AWS) na sua conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta.
6. Em Rule type, escolha Rule with an event pattern.
7. Escolha Próximo.
8. Em Event source, escolha Eventos da AWS ou eventos de parceiro do EventBridge.
9. Na seção Event patter (Padrão de evento), selecione Event pattern form (Formulário de padrão de evento).
10. Em Fonte do evento, selecione Serviços da AWS.
11. Em Serviço da AWS, escolha Systems Manager.
12. Para Event type (Tipo de evento), escolha Inventory (Inventário).
13. Em Specific detail type(s) (Tipos de detalhes específicos), escolha Inventory Resource State Change (Alteração de estado de recursos de inventário).
14. Escolha Próximo.
15. Em Tipos de destino, escolha Serviço da AWS.
16. Em Select a target (Selecionar um destino), escolha SNS topic (Tópico do SNS) e depois escolha seu tópico em Topic.
17. Na seção Additional settings (Configurações adicionais), para Configure target input (Configurar entrada do destino), verifique se a opção Matched event (Evento correspondido) está selecionada.

18. Escolha Próximo.
19. (Opcional) Insira uma ou mais tags para a regra. Para obter mais informações, consulte Marcar recursos do Amazon EventBridge no Guia do usuário do Amazon EventBridge.
20. Escolha Próximo.
21. Analise os detalhes da regra e selecione Criar regra.

Visualizar o histórico do inventário e o controle de alterações

Você pode visualizar o histórico do AWS Systems Manager Inventory e alterar o rastreamento de todos os nós gerenciados usando o AWS Config. O AWS Config fornece uma visão detalhada da configuração dos recursos da AWS na Conta da AWS. Isso inclui como os recursos estão relacionados um com o outro e como eles foram configurados no passado, de modo que você possa ver como os relacionamentos e as configurações foram alterados ao longo do tempo. Para visualizar o histórico do inventário e o rastreamento de alterações, ative os seguintes recursos no AWS Config:

- SSM:ManagedInstanceInventory
- SSM:PatchCompliance
- SSM:AssociationCompliance
- SSM:FileData

> **Note**
>
> Observe os seguintes detalhes essenciais sobre o histórico do Inventory e o controle de monitoramento:
>
> - Se você usar o AWS Config para rastrear alterações no sistema, você deverá configurar o Systems Manager Inventory para coletar metadados do `AWS:File` para que você possa exibir as alterações do arquivo no AWS Config (`SSM:FileData`). Se não tiver, o AWS Config não monitorará as alterações de arquivos em seu sistema.
> - Habilitando o SSM:PatchCompliance e o SSM:AssociationCompliance, você poderá visualizar a aplicação de patches do Systems Manager Patch Manager, além do histórico de conformidade de associações e monitoramento de alterações do State Manager. Para obter mais informações sobre o gerenciamento de conformidade para esses recursos, consulte Conheça mais sobre a conformidade.

O procedimento a seguir descreve como ativar a gravação do histórico do inventário e do rastreamento de alterações no AWS Config usando a AWS Command Line Interface (AWS CLI). Para obter mais informações sobre como escolher e configurar esses recursos no AWS Config, consulte Selecionar que recursos o AWS Config registra no Manual do desenvolvedor do AWS Config. Para obter mais informações sobre a definição de preço do AWS Config, consulte Definição de preço do .

Antes de começar

O AWS Config exige permissões do AWS Identity and Access Management (IAM) para obter detalhes da configuração sobre os recursos do Systems Manager. No procedimento a seguir, você deve especificar um nome do recurso da Amazon (ARN) para uma função do IAM que conceda permissão da AWS Config para recursos do Systems Manager. Você pode anexar a política gerenciada do `AWS_ConfigRole` à função do IAM que você atribuiu ao AWS Config. Para obter mais informações sobre esse perfil, consulte Política gerenciada pela AWS:AWS_ConfigRole no Guia do desenvolvedor do AWS Config. Para obter informações sobre como criar um perfil do IAM e atribuir a política gerenciada pelo `AWS_ConfigRole`, consulte Criar uma função para delegar permissões a um AWS service (Serviço da AWS), no Guia do usuário do IAM.

Como ativar o histórico do inventário e a gravação do rastreamento de alterações no AWS Config

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Copie e cole o seguinte exemplo de JSON em um editor de texto simples e salve-o como recordingGroup.json.

```
{
   "allSupported":false,
   "includeGlobalResourceTypes":false,
   "resourceTypes":[
      "AWS::SSM::AssociationCompliance",
      "AWS::SSM::PatchCompliance",
      "AWS::SSM::ManagedInstanceInventory",
      "AWS::SSM::FileData"
   ]
}
```

3. Execute o seguinte comando para carregar o arquivo recordingGroup.json no AWS Config.

```
aws configservice put-configuration-recorder --configuration-recorder
 name=myRecorder,roleARN=arn:aws:iam::123456789012:role/myConfigRole --recording-
group file://recordingGroup.json
```

4. Execute o seguinte comando para iniciar a gravação do histórico do inventário e do rastreamento de alterações.

```
aws configservice start-configuration-recorder --configuration-recorder-
name myRecorder
```

Depois de configurar o histórico e o controle de alterações, você poderá buscar um nó gerenciado específico no histórico escolhendo o botão AWS Config no console do Systems Manager. Você pode acessar o botão AWS Config na página Managed Instances (Instâncias gerenciadas) ou na página Inventory (Inventário). Dependendo do tamanho de seu monitor, talvez seja necessário rolar para o lado direito da página para ver o botão.

interromper a coleta de dados e excluir os dados do inventário

Se você não quiser mais usar o AWS Systems Manager Inventary para exibir metadados sobre os recursos da AWS, você poderá interromper a coleta de dados e excluir os dados que já tiverem sido coletados. Esta seção inclui as seguintes informações:

Tópicos

- Interromper a coleta de dados
- Excluir uma sincronização de dados de recursos do Inventory

Interromper a coleta de dados

Quando você configura inicialmente o Systems Manager para coletar dados do inventário, o sistema cria uma associação do State Manager que define a programação e os recursos dos quais coletar os metadados. Você pode interromper a coleta de dados excluindo todas as associações do State Manager que usarem o documento `AWS-GatherSoftwareInventory`.

Para excluir uma associação do Inventory

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha State Manager.

3. Escolha uma associação que use o documento `AWS-GatherSoftwareInventory` e, em seguida, selecione Delete (Excluir).
4. Repita a etapa 3 para todas as associações restantes que usarem o documento `AWS-GatherSoftwareInventory`.

Excluir uma sincronização de dados de recursos do Inventory

Se você não quiser mais usar o AWS Systems Manager Inventory para exibir metadados sobre a AWS, também recomendamos excluir sincronizações de dados de recursos usadas para a coleta de dados do inventário.

Para excluir uma sincronização de dados de recursos do Inventory

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Inventory.
3. Selecione Create resource data sync (Criar sincronização dos dados do recurso).
4. Na lista, escolha um sincronização.

> **Important**
>
> Escolha a sincronização usada para o Inventory. O Systems Manager oferece suporte à sincronização de dados de recursos para várias ferramentas. Se você escolher a sincronização errada, a agregação de dados para o Systems Manager ou para o Systems Manager Compliance poderá ser interrompida.

5. Escolha Delete (Excluir).
6. Repita essas etapas para todas as sincronizações de dados de recursos restantes que você quiser excluir.
7. Exclua o bucket do Amazon Simple Storage Service (Amazon S3) no qual os dados foram armazenados. Para obter informações sobre como excluir um bucket do Amazon S3, consulte Excluir um bucket.

Assigning custom inventory metadata to a managed node

O procedimento a seguir demonstra o processo de usar a operação da API PutInventory do AWS Systems Manager para atribuir metadados do inventário personalizado a um nó gerenciado. Este

exemplo atribui informações de localização de rack a um nó. Para obter mais informações sobre o inventário personalizado, consulte Trabalhar com inventário personalizado.

Para atribuir metadados do inventário personalizado a um nó

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Execute o seguinte comando para atribuir informações de localização de rack a um nó:

   Linux

   ```
   aws ssm put-inventory --instance-id "ID" --items '[{"CaptureTime":
    "2016-08-22T10:01:01Z", "TypeName": "Custom:RackInfo", "Content":[{"RackLocation":
    "Bay B/Row C/Rack D/Shelf E"}], "SchemaVersion": "1.0"}]'
   ```

   Windows

   ```
   aws ssm put-inventory --instance-id "ID" --items
    "TypeName=Custom:RackInfo,SchemaVersion=1.0,CaptureTime=2021-05-22T10:01:01Z,Content=[{Rac
    B/Row C/Rack D/Shelf F'}]"
   ```

3. Execute o seguinte comando para visualizar as entradas de inventário personalizadas para esse nó:

   ```
   aws ssm list-inventory-entries --instance-id ID --type-name "Custom:RackInfo"
   ```

   O sistema responde com informações como as seguintes.

   ```
   {
       "InstanceId": "ID",
       "TypeName": "Custom:RackInfo",
       "Entries": [
           {
               "RackLocation": "Bay B/Row C/Rack D/Shelf E"
           }
       ],
       "SchemaVersion": "1.0",
       "CaptureTime": "2016-08-22T10:01:01Z"
   }
   ```

4. Execute o seguinte comando para visualizar o esquema de inventário personalizado.

```
aws ssm get-inventory-schema --type-name Custom:RackInfo
```

O sistema responde com informações como as seguintes.

```
{
    "Schemas": [
        {
            "TypeName": "Custom:RackInfo",
            "Version": "1.0",
            "Attributes": [
                {
                    "DataType": "STRING",
                    "Name": "RackLocation"
                }
            ]
        }
    ]
}
```

Using the AWS CLI to configure inventory data collection

Os procedimentos a seguir demonstram o processo de configuração do inventário do AWS Systems Manager para coletar metadados de seus nós gerenciados. Ao configurar a coleta do inventário, você começa criando uma associação do State Manager para o Systems Manager. O Systems Manager coleta os dados de inventário quando a associação é executada. Se você não criar a associação primeiro e tentar invocar o plugin `aws:softwareInventory` usando, por exemplo, o Run Command do Systems Manager, o sistema retornará o seguinte erro:

`The aws:softwareInventory plugin can only be invoked via ssm-associate.`

> **Note**
>
> Um nó gerenciado pode ter apenas uma associação de inventário configurada por vez. Se você configurar um nó com duas ou mais associações de inventário, a associação não será executada, e os dados do inventário não serão coletados.

Configure rapidamente todos os nós gerenciados para o inventário (CLI)

Você pode configurar rapidamente todos os nós gerenciados na sua Conta da AWS e na região atual para coletar dados do inventário. Esse processo é chamado de criação de uma associação global de inventário. Para criar uma associação de inventário global usando a AWS CLI, use a opção curinga para o valor `instanceIds`, conforme mostrado no procedimento a seguir.

Para configurar o inventário em todos os nós gerenciados na sua Conta da AWS e na região atual (CLI)

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Execute o seguinte comando:

   Linux & macOS

   ```
   aws ssm create-association \
   --name AWS-GatherSoftwareInventory \
   --targets Key=InstanceIds,Values=* \
   --schedule-expression "rate(1 day)" \
   --parameters
    applications=Enabled,awsComponents=Enabled,customInventory=Enabled,instanceDetailedInfo
   ```

   Windows

   ```
   aws ssm create-association ^
   --name AWS-GatherSoftwareInventory ^
   --targets Key=InstanceIds,Values=* ^
   --schedule-expression "rate(1 day)" ^
   --parameters
    applications=Enabled,awsComponents=Enabled,customInventory=Enabled,instanceDetailedInfo
   ```

> **Note**
>
> Este comando não permite que o Inventory colete metadados para o Registro ou arquivos do Windows. Para inventariar esses tipos de dados, use o próximo procedimento.

Configurar manualmente o inventário em nós gerenciados (CLI)

Use o procedimento a seguir para configurar manualmente o inventário do AWS Systems Manager em seus nós gerenciados usando IDs ou tags do nó.

Para configurar manualmente seus nós gerenciados para o inventário (CLI)

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Execute o comando a seguir para criar uma associação do State Manager que execute o Inventário do Systems Manager nesse nó. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações. Esse comando configura o serviço para executar a cada seis horas e coletar a configuração de rede, o Windows Update e os metadados de aplicações de um nó.

   Linux & macOS

   ```
   aws ssm create-association \
   --name "AWS-GatherSoftwareInventory" \
   --targets "Key=instanceids,Values=an_instance_ID" \
   --schedule-expression "rate(240 minutes)" \
   --output-location "{ \"S3Location\": { \"OutputS3Region\": \"region_ID,
    for example us-east-2\", \"OutputS3BucketName\": \"amzn-s3-demo-bucket\",
    \"OutputS3KeyPrefix\": \"Test\" } }" \
   --parameters "networkConfig=Enabled,windowsUpdates=Enabled,applications=Enabled"
   ```

   Windows

   ```
   aws ssm create-association ^
   --name "AWS-GatherSoftwareInventory" ^
   --targets "Key=instanceids,Values=an_instance_ID" ^
   --schedule-expression "rate(240 minutes)" ^
   --output-location "{ \"S3Location\": { \"OutputS3Region\": \"region_ID,
    for example us-east-2\", \"OutputS3BucketName\": \"amzn-s3-demo-bucket\",
    \"OutputS3KeyPrefix\": \"Test\" } }" ^
   --parameters "networkConfig=Enabled,windowsUpdates=Enabled,applications=Enabled"
   ```

   O sistema responde com informações como as seguintes.

```
{
    "AssociationDescription": {
        "ScheduleExpression": "rate(240 minutes)",
        "OutputLocation": {
            "S3Location": {
                "OutputS3KeyPrefix": "Test",
                "OutputS3BucketName": "Test bucket",
                "OutputS3Region": "us-east-2"
            }
        },
        "Name": "The name you specified",
        "Parameters": {
            "applications": [
                "Enabled"
            ],
            "networkConfig": [
                "Enabled"
            ],
            "windowsUpdates": [
                "Enabled"
            ]
        },
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "AssociationId": "1a2b3c4d5e6f7g-1a2b3c-1a2b3c-1a2b3c-1a2b3c4d5e6f7g",
        "DocumentVersion": "$DEFAULT",
        "LastUpdateAssociationDate": 1480544990.06,
        "Date": 1480544990.06,
        "Targets": [
            {
                "Values": [
                   "i-02573cafcfEXAMPLE"
                ],
                "Key": "InstanceIds"
            }
        ]
    }
}
```

Você pode direcionar grupos extensos de nós usando o parâmetro `Targets` com tags do EC2. Veja o exemplo a seguir.

Linux & macOS

```
aws ssm create-association \
--name "AWS-GatherSoftwareInventory" \
--targets "Key=tag:Environment,Values=Production" \
--schedule-expression "rate(240 minutes)" \
--output-location "{ \"S3Location\": { \"OutputS3Region\": \"us-east-2\",
 \"OutputS3BucketName\": \"amzn-s3-demo-bucket\", \"OutputS3KeyPrefix\": \"Test
\" } }" \
--parameters "networkConfig=Enabled,windowsUpdates=Enabled,applications=Enabled"
```

Windows

```
aws ssm create-association ^
--name "AWS-GatherSoftwareInventory" ^
--targets "Key=tag:Environment,Values=Production" ^
--schedule-expression "rate(240 minutes)" ^
--output-location "{ \"S3Location\": { \"OutputS3Region\": \"us-east-2\",
 \"OutputS3BucketName\": \"amzn-s3-demo-bucket\", \"OutputS3KeyPrefix\": \"Test
\" } }" ^
--parameters "networkConfig=Enabled,windowsUpdates=Enabled,applications=Enabled"
```

Você também pode inventariar arquivos e chaves do Registro do Windows em um nó do Windows Server usando os tipos de inventário `files` e `windowsRegistry` com expressões. Para obter mais informações sobre esses tipos de inventário, consulte Trabalhar com o inventário de arquivos e do Registro do Windows.

Linux & macOS

```
aws ssm create-association \
--name "AWS-GatherSoftwareInventory" \
--targets "Key=instanceids,Values=i-0704358e3a3da9eb1" \
--schedule-expression "rate(240 minutes)" \
--parameters '{"files":["[{\"Path\": \"C:\\Program Files\", \"Pattern\":
 [\"*.exe\"], \"Recursive\": true}]"], "windowsRegistry": ["[{\"Path\":
\"HKEY_LOCAL_MACHINE\\Software\\Amazon\", \"Recursive\":true}]"]}' \
```

```
--profile dev-pdx
```

Windows

```
aws ssm create-association ^
--name "AWS-GatherSoftwareInventory" ^
--targets "Key=instanceids,Values=i-0704358e3a3da9eb1" ^
--schedule-expression "rate(240 minutes)" ^
--parameters '{"files":["[{\"Path\": \"C:\\Program Files\", \"Pattern\":
 [\"*.exe\"], \"Recursive\": true}]"], "windowsRegistry": ["[{\"Path\":
\"HKEY_LOCAL_MACHINE\\Software\\Amazon\", \"Recursive\":true}]"]}' ^
--profile dev-pdx
```

3. Execute o seguinte comando para visualizar o status da associação.

```
aws ssm describe-instance-associations-status --instance-id an_instance_ID
```

O sistema responde com informações como as seguintes.

```
{
"InstanceAssociationStatusInfos": [
         {
            "Status": "Pending",
            "DetailedStatus": "Associated",
            "Name": "reInvent2016PolicyDocumentTest",
            "InstanceId": "i-1a2b3c4d5e6f7g",
            "AssociationId": "1a2b3c4d5e6f7g-1a2b3c-1a2b3c-1a2b3c-1a2b3c4d5e6f7g",
            "DocumentVersion": "1"
        }
]
}
```

Demonstração: usar a sincronização de dados de recursos para agregar dados do inventário

A demonstração a seguir descreve como criar uma configuração de sincronização de dados de recursos para o AWS Systems Manager Inventory, usando a AWS Command Line Interface (AWS CLI). A sincronização de dados de recursos sincroniza automaticamente os dados de inventário de portas de todos os nós gerenciados em um bucket central do Amazon Simple Storage Service

(Amazon S3). A sincronização atualiza automaticamente os dados no bucket central do Amazon S3 quando novos dados do inventário são descobertos.

Essa demonstração também descreve como usar o Amazon Athena e o Amazon QuickSight para consultar e analisar os dados agregados. Para obter informações sobre como criar uma sincronização de dados de recursos usando o Systems Manager no AWS Management Console, consulte Demonstração: usar a sincronização de dados de recursos para agregar dados do inventário. Para obter informações sobre como consultar o inventário de várias Regiões da AWS e contas usando o Systems Manager no AWS Management Console, consulte Consultar dados de inventário de várias regiões e contas.

> **Note**
>
> Essa demonstração inclui informações sobre como criptografar a sincronização usando o AWS Key Management Service (AWS KMS). O Inventory não coleta dados específicos do usuário, dados patenteados ou confidenciais e, portanto, a criptografia é opcional. Para obter mais informações sobre o AWS KMS, consulte o Manual do desenvolvedor do AWS Key Management Service.

Antes de começar

Avalie ou conclua as tarefas a seguir, antes de começar a demonstração nesta seção:

- Colete o inventário de dados dos nós gerenciados. Para fins das seções Amazon Athena e Amazon QuickSight nesta demonstração, recomendamos que você colete os dados da aplicação. Para obter mais informações sobre como coletar dados do inventário, consulte Configurar a coleta de inventário ou Using the AWS CLI to configure inventory data collection.
- (Opcional) Se os dados de inventário forem armazenados em um bucket do Amazon Simple Storage Service (Amazon S3) que use a criptografia do AWS Key Management Service (AWS KMS), você também deve configurar sua conta do IAM e a função de serviço do `Amazon-GlueServiceRoleForSSM` para a criptografia do AWS KMS. Se você não configurar a conta do IAM e essa função, o Systems Manager exibirá `Cannot load Glue tables` quando você escolher a guia Detailed View (Visualização detalhada) no console. Para obter mais informações, consulte (Opcional) Configure as permissões para exibição de dados criptografados do AWS KMS.
- (Opcional) Se Você quiser criptografar a sincronização dos dados do recurso, usando o AWS KMS, crie uma nova chave que inclua a política a seguir, ou atualize uma chave existente e adicione essa política a ela.

JSON

```
{
    "Version": "2012-10-17",
    "Id": "ssm-access-policy",
    "Statement": [
        {
            "Sid": "ssm-access-policy-statement",
            "Action": [
                "kms:GenerateDataKey"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Resource": "arn:aws:kms:us-east-2:123456789012:key/KMS_key_id",
            "Condition": {
                "StringLike": {
                    "aws:SourceAccount": "123456789012"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:ssm:*:123456789012:resource-data-
sync/*"
                }
            }
        }
    ]
}
```

Como criar uma sincronização de dados de recurso para inventário

1. Abra o console do Amazon S3, em https://console.aws.amazon.com/s3/.
2. Crie um bucket para armazenar os dados de inventário agregados. Para obter mais informações, consulte Criar um bucket no Guia do usuário do Amazon Simple Storage Service. Anote o nome do bucket e a Região da AWS em que você o criou.
3. Depois de criar o bucket, escolha a guia Permissions e depois escolha Bucket Policy.
4. Copie e cole a seguinte política de bucket no editor de políticas. Substitua amzn-s3-demo-bucket e *account-id* pelo nome do bucket do Amazon S3 que você criou e um ID de Conta da AWS válido. Ao adicionar múltiplas contas, adicione mais uma sequência de caracteres de condição e

um ARN para cada conta. Ao adicionar uma conta, remova os espaços reservados adicionais do exemplo. Opcionalmente, substitua *bucket-prefix* pelo nome de um prefixo do Amazon S3 (subdiretório). Se você não criou um prefixo, remova *bucket-prefix/* do ARN na política.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": " SSMBucketDelivery",
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/bucket-prefix/*/
accountid=account-id/*"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": [
                        "account-id1",
                        "account-id2",
                        "account-id3",
                        "account-id4"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:ssm:*:111122223333:resource-data-sync/*",
                        "arn:aws:ssm:*:111122223333:resource-data-sync/*",
                        "arn:aws:ssm:*:111122223333:resource-data-sync/*",
                        "arn:aws:ssm:*:111122223333:resource-data-sync/*"
                    ]
                }
            }
        }
    ]
}
```

5. (Opcional) Se deseja criptografar a sincronização, você deve adicionar as seguintes condições à política listada na etapa anterior. Adicione esses na seção `StringEquals`.

```
"s3:x-amz-server-side-encryption":"aws:kms",
"s3:x-amz-server-side-encryption-aws-kms-key-
id":"arn:aws:kms:region:account_ID:key/KMS_key_ID"
```

   Exemplo:

```
"StringEquals": {
          "s3:x-amz-acl": "bucket-owner-full-control",
          "aws:SourceAccount": "account-id",
          "s3:x-amz-server-side-encryption":"aws:kms",
          "s3:x-amz-server-side-encryption-aws-kms-key-
id":"arn:aws:kms:region:account_ID:key/KMS_key_ID"
        }
```

6. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

7. (Opcional) Se você quiser criptografar a sincronização, execute o seguinte comando para verificar se a política do bucket está aplicando o requisito de chave do AWS KMS. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   Linux & macOS

```
aws s3 cp ./A_file_in_the_bucket s3://amzn-s3-demo-bucket/prefix/ \
--sse aws:kms \
--sse-kms-key-id "arn:aws:kms:region:account_ID:key/KMS_key_id" \
--region region, for example, us-east-2
```

   Windows

```
aws s3 cp ./A_file_in_the_bucket s3://amzn-s3-demo-bucket/prefix/ ^
    --sse aws:kms ^
    --sse-kms-key-id "arn:aws:kms:region:account_ID:key/KMS_key_id" ^
    --region region, for example, us-east-2
```

8. Execute o comando a seguir para criar uma configuração de sincronização de dados de recursos com o bucket do Amazon S3 que você criou no início deste procedimento. Esse comando cria uma sincronização da Região da AWS à qual você está conectado.

> **Note**
>
> Se a sincronização e o bucket do Amazon S3 de destino estiverem localizados em diferentes regiões, você poderá estar sujeito ao preço da transferência de dados. Para obter mais informações, consulte Preços do Amazon S3.

Linux & macOS

```
aws ssm create-resource-data-sync \
--sync-name a_name \
--s3-destination "BucketName=amzn-s3-demo-bucket,Prefix=prefix_name,
 if_specified,SyncFormat=JsonSerDe,Region=bucket_region"
```

Windows

```
aws ssm create-resource-data-sync ^
--sync-name a_name ^
--s3-destination "BucketName=amzn-s3-demo-bucket,Prefix=prefix_name,
 if_specified,SyncFormat=JsonSerDe,Region=bucket_region"
```

Você pode usar o parâmetro `region` para especificar onde a configuração de sincronização deve ser criada. No exemplo a seguir, os dados de inventário da região us-west-1 serão sincronizados no bucket do Amazon S3 na região us-west-2.

Linux & macOS

```
aws ssm create-resource-data-sync \
    --sync-name InventoryDataWest \
    --s3-destination "BucketName=amzn-s3-demo-
bucket,Prefix=HybridEnv,SyncFormat=JsonSerDe,Region=us-west-2"
    --region us-west-1
```

Windows

```
aws ssm create-resource-data-sync ^
--sync-name InventoryDataWest ^
--s3-destination "BucketName=amzn-s3-demo-
bucket,Prefix=HybridEnv,SyncFormat=JsonSerDe,Region=us-west-2" ^ --region us-
west-1
```

(Opcional) Se quiser criptografar a sincronização usando o AWS KMS, execute o seguinte comando para criar a sincronização. Se você criptografar a sincronização, a chave AWS KMS e o bucket do Amazon S3 deverão estar na mesma região.

Linux & macOS

```
aws ssm create-resource-data-sync \
--sync-name sync_name \
--s3-destination "BucketName=amzn-s3-demo-bucket,Prefix=prefix_name,
 if_specified,SyncFormat=JsonSerDe,AWSKMSKeyARN=arn:aws:kms:region:account_ID:key/
KMS_key_ID,Region=bucket_region" \
--region region
```

Windows

```
aws ssm create-resource-data-sync ^
--sync-name sync_name ^
--s3-destination "BucketName=amzn-s3-demo-bucket,Prefix=prefix_name,
 if_specified,SyncFormat=JsonSerDe,AWSKMSKeyARN=arn:aws:kms:region:account_ID:key/
KMS_key_ID,Region=bucket_region" ^
--region region
```

9. Execute o seguinte comando para visualizar o status da configuração de sincronização.

```
aws ssm list-resource-data-sync
```

Se você criou a configuração de sincronização em uma Região diferente, deverá especificar o parâmetro `region`, como mostra o exemplo a seguir.

```
aws ssm list-resource-data-sync --region us-west-1
```

10. Depois que a configuração de sincronização tiver sido criada com sucesso, navegue pelo bucket de destino no Amazon S3. Os dados do Inventory deverão ser exibidos em poucos minutos.

Trabalhar com os dados no Amazon Athena

A seção a seguir descreve como visualizar e consultar os dados no Amazon Athena. Antes de começar, recomendamos que você saiba mais sobre o Athena. Para obter mais informações, consulte O que é o Amazon Athena? e Trabalhar com dados no Manual do usuário do Amazon Athena.

Para visualizar e consultar dados no Amazon Athena

1. Abra o console do Athena em https://console.aws.amazon.com/athena/.
2. Copie e cole a seguinte instrução no editor de consultas e escolha Run Query.

   ```
   CREATE DATABASE ssminventory
   ```

   O sistema cria um banco de dados chamado ssminventory.
3. Copie e cole a seguinte instrução no editor de consultas e escolha Run Query. Substitua amzn-s3-demo-bucket e *bucket_prefix* pelo nome e prefixo do destino do Amazon S3.

   ```
   CREATE EXTERNAL TABLE IF NOT EXISTS ssminventory.AWS_Application (
   Name string,
   ResourceId string,
   ApplicationType string,
   Publisher string,
   Version string,
   InstalledTime string,
   Architecture string,
   URL string,
   Summary string,
   PackageId string
   )
   PARTITIONED BY (AccountId string, Region string, ResourceType string)
   ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
   WITH SERDEPROPERTIES (
     'serialization.format' = '1'
   ) LOCATION 's3://amzn-s3-demo-bucket/bucket_prefix/AWS:Application/'
   ```
4. Copie e cole a seguinte instrução no editor de consultas e escolha Run Query.

```
MSCK REPAIR TABLE ssminventory.AWS_Application
```

O sistema particiona a tabela.

> **Note**
>
> Se você criar sincronizações de dados de recursos em outras Regiões da AWS ou Contas da AWS, você deverá executar esse comando novamente para atualizar as partições. Talvez você também precise atualizar a política de bucket do Amazon S3.

5. Para visualizar seus dados, escolha o ícone de exibição ao lado da tabela `AWS_Application`.



6. Copie e cole a seguinte instrução no editor de consultas e escolha Run Query.

```
SELECT a.name, a.version, count( a.version) frequency
from aws_application a where
a.name = 'aws-cfn-bootstrap'
group by a.name, a.version
order  by frequency desc
```

A consulta retorna uma contagem de versões diferentes do `aws-cfn-bootstrap`, que é uma aplicação da AWS presente nas instâncias do Amazon Elastic Compute Cloud (Amazon EC2) para Linux, macOS e Windows Server.

7. Copie e cole individualmente as seguintes instruções no editor de consultas, substitua amzn-s3-demo-bucket e *`bucket-prefix`* por informações para o Amazon S3 e escolha Executar consulta. Essas instruções configuram as tabelas de inventário adicionais no Athena.

```
CREATE EXTERNAL TABLE IF NOT EXISTS ssminventory.AWS_AWSComponent (
 `ResourceId` string,
  `Name` string,
  `ApplicationType` string,
  `Publisher` string,
  `Version` string,
  `InstalledTime` string,
  `Architecture` string,
  `URL` string
```

```
)
PARTITIONED BY (AccountId string, Region string, ResourceType string)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = '1'
) LOCATION 's3://amzn-s3-demo-bucket/bucket-prefix/AWS:AWSComponent/'
```

```
MSCK REPAIR TABLE ssminventory.AWS_AWSComponent
```

```
CREATE EXTERNAL TABLE IF NOT EXISTS ssminventory.AWS_WindowsUpdate (
  `ResourceId` string,
  `HotFixId` string,
  `Description` string,
  `InstalledTime` string,
  `InstalledBy` string
)
PARTITIONED BY (AccountId string, Region string, ResourceType string)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = '1'
) LOCATION 's3://amzn-s3-demo-bucket/bucket-prefix/AWS:WindowsUpdate/'
```

```
MSCK REPAIR TABLE ssminventory.AWS_WindowsUpdate
```

```
CREATE EXTERNAL TABLE IF NOT EXISTS ssminventory.AWS_InstanceInformation (
  `AgentType` string,
  `AgentVersion` string,
  `ComputerName` string,
  `IamRole` string,
  `InstanceId` string,
  `IpAddress` string,
  `PlatformName` string,
  `PlatformType` string,
  `PlatformVersion` string
)
PARTITIONED BY (AccountId string, Region string, ResourceType string)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = '1'
) LOCATION 's3://amzn-s3-demo-bucket/bucket-prefix/AWS:InstanceInformation/'
```

```
MSCK REPAIR TABLE ssminventory.AWS_InstanceInformation
```

```
CREATE EXTERNAL TABLE IF NOT EXISTS ssminventory.AWS_Network (
  `ResourceId` string,
  `Name` string,
  `SubnetMask` string,
  `Gateway` string,
  `DHCPServer` string,
  `DNSServer` string,
  `MacAddress` string,
  `IPV4` string,
  `IPV6` string
)
PARTITIONED BY (AccountId string, Region string, ResourceType string)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = '1'
) LOCATION 's3://amzn-s3-demo-bucket/bucket-prefix/AWS:Network/'
```

```
MSCK REPAIR TABLE ssminventory.AWS_Network
```

```
CREATE EXTERNAL TABLE IF NOT EXISTS ssminventory.AWS_PatchSummary (
  `ResourceId` string,
  `PatchGroup` string,
  `BaselineId` string,
  `SnapshotId` string,
  `OwnerInformation` string,
  `InstalledCount` int,
  `InstalledOtherCount` int,
  `NotApplicableCount` int,
  `MissingCount` int,
  `FailedCount` int,
  `OperationType` string,
  `OperationStartTime` string,
  `OperationEndTime` string
)
PARTITIONED BY (AccountId string, Region string, ResourceType string)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = '1'
```

```
) LOCATION 's3://amzn-s3-demo-bucket/bucket-prefix/AWS:PatchSummary/'
```

```
MSCK REPAIR TABLE ssminventory.AWS_PatchSummary
```

Trabalhar com os dados no Amazon QuickSight

A seção a seguir fornece uma visão geral com links para construir uma visualização no Amazon QuickSight.

Para construir uma visualização no Amazon QuickSight

1. Cadastre-se no Amazon QuickSight e faça login no console do QuickSight.
2. Crie um conjunto de dados na tabela `AWS_Application` e em quaisquer outras tabelas que você tenha criado. Para obter mais informações, consulte Criar um conjunto de dados usando o Amazon Athena.
3. Una tabelas. Por exemplo, você pode unir a coluna `instanceid` de `AWS_InstanceInformation` porque ela corresponde à coluna `resourceid` em outras tabelas de inventário. Para obter mais informações sobre como unir tabelas, consulte o tópico sobre como Unir tabelas.
4. Crie uma visualização. Para obter mais informações, consulte o tópico sobre como Trabalhar com o Amazon QuickSight Visuals.

Solucionar problemas com o Systems Manager Inventory

Este tópico inclui informações sobre como resolver erros comuns ou problemas com o inventário do AWS Systems Manager. Se você estiver tendo problemas para visualizar os nós no Systems Manager, consulte Solução de problemas de disponibilidade do nó gerenciado.

Tópicos

- Os múltiplos que aplicam todas as associações com o documento 'AWS-GatherSoftwareInventory' não são compatíveis
- O status de execução do Inventory nunca sai de pendente
- Falha na execução do documento AWS-ListWindowsInventory
- O console não exibe as guias Dashboard (Painel) | Detailed View (Visualização detalhada) | Settings (Configurações) do inventário

- UnsupportedAgent
- Ignorado
- Falha
- Falha na conformidade do inventário para uma instância do Amazon EC2
- O objeto do bucket do S3 contém dados antigos

Os múltiplos que aplicam todas as associações com o documento '**AWS-GatherSoftwareInventory**' não são compatíveis

Um erro `Multiple apply all associations with document 'AWS-GatherSoftwareInventory' are not supported` significa que um ou mais Regiões da AWS nas quais você está tentando configurar uma associação do Inventory para todas os nós gerenciados já estão definidas com uma associação de inventário para todas as instâncias. Se necessário, você pode excluir a associação de inventário existente para todas os nós e criar uma nova. Para visualizar as associações de inventário existentes, escolha State Manager no console do Systems Manager e, em seguida, localize associações que usam o Documento do SSM `AWS-GatherSoftwareInventory`. Se a associação de inventário existente para todos os nós tiver sido criada em várias regiões e você quiser criar uma nova, exclua a associação existente de cada região onde ela existir.

O status de execução do Inventory nunca sai de pendente

Há duas razões pelas quais a coleta do inventário nunca sai do status `Pending`:

- Nenhum nó presente na Região da AWS selecionada:

  Se você criar uma associação de inventário global usando o Systems Manager Quick Setup, o status da associação de inventário (documento `AWS-GatherSoftwareInventory`) será `Pending`, caso nenhum nó esteja disponível na região selecionada.
- Permissões insuficientes:

  Uma associação de inventário mostrará `Pending` se um ou mais nós gerenciados não tiverem permissão para executar o Systems Manager Inventory. Verifique se o perfil de instância do AWS Identity and Access Management (IAM) inclui a política gerenciada AmazonSSMManagedInstanceCore Para obter informações sobre como adicionar essa política a um perfil de instância, consulte Configuração alternativa para permissões de instâncias do EC2.

  O perfil de instância deve ter, no mínimo, as seguintes permissões do IAM:

JSON

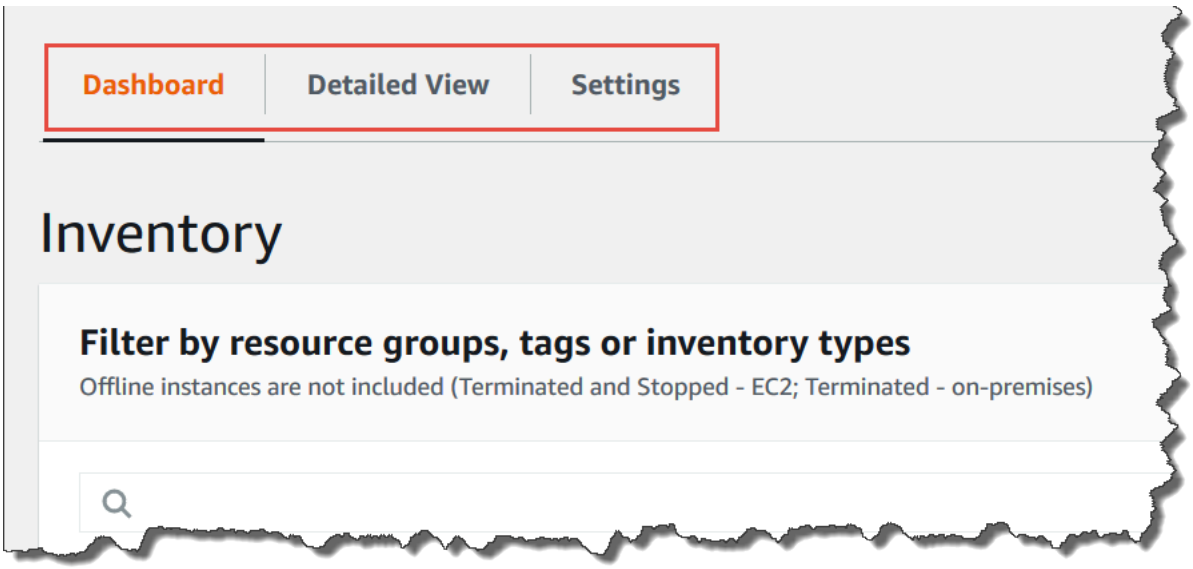
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeAssociation",
                "ssm:ListAssociations",
                "ssm:ListInstanceAssociations",
                "ssm:PutInventory",
                "ssm:PutComplianceItems",
                "ssm:UpdateAssociationStatus",
                "ssm:UpdateInstanceAssociationStatus",
                "ssm:UpdateInstanceInformation",
                "ssm:GetDocument",
                "ssm:DescribeDocument"
            ],
            "Resource": "*"
        }
    ]
}
```

Falha na execução do documento **`AWS-ListWindowsInventory`**

O documento `AWS-ListWindowsInventory` está obsoleto. Não use esse documento para coletar inventário. Em vez disso, use um dos processos descritos em Configurar a coleta de inventário.

O console não exibe as guias Dashboard (Painel) | Detailed View (Visualização detalhada) | Settings (Configurações) do inventário

A página Detailed View (Visualização detalhada) do Inventory só está disponível nas Regiões da AWS que oferecem o Amazon Athena. Se as seguintes guias não forem exibidas na página do Inventory, isso significa que o Athena não está disponível na região e que você não pode usar a Detailed View (Visualização detalhada) para consultar os dados.



UnsupportedAgent

Se o status detalhado de uma associação de inventário mostrar UnsupportedAgent e o Association status (status da associação) mostrar Failed (Falhou), a versão do AWS Systems Manager SSM Agent no nó gerenciado não está correta. Para criar uma associação de inventário global (para fazer inventário de todos os nós em sua Conta da AWS), por exemplo, você precisa usar o SSM Agent versão 2.0.790.0 ou posterior. Você pode ver a versão do agente em execução em cada um de seus nós gerenciados na página Managed Instances (Instâncias gerenciadas) na coluna Agent version (Versão do agente). Para obter informações sobre como atualizar o SSM Agent em nós gerenciados, consulte Atualização do SSM Agent por meio de Run Command.

Ignorado

Se o status da associação de inventário de um nó mostrar Skipped, isso significa que uma associação de inventário de prioridade mais alta já está em execução nesse nó. O Systems Manager segue uma ordem de prioridade específica quando várias associações de inventário podem ser aplicadas ao mesmo nó gerenciado.

Ordem de prioridade da associação de inventário

O Systems Manager aplica associações de inventário na seguinte ordem de prioridade:

1. Associações de inventário de Quick Setup: associações criadas usando a Quick Setup e o console unificado. Essas associações têm nomes que começam com `AWS-QuickSetup-SSM-CollectInventory-` e têm como destino todos os nós gerenciados.

2. Associações explícitas de inventário: associações que têm como destino nós gerenciados específicos usando:
   - IDs de instância
   - Pares de chave/valor de tags
   - Grupos de recursos AWS
3. Associações globais de inventário: associações que têm como destino todos os nós gerenciados (usando `--targets "Key=InstanceIds,Values=*"`), mas que não foram criadas por meio da Quick Setup.

Cenários comuns

Cenário 1: a associação de Quick Setup substitui a associação explícita

- Você tem uma associação de inventário de Quick Setup direcionada a todas as instâncias
- Você cria uma associação manual direcionada para nós gerenciados específicos por tag
- Resultado: a associação manual mostra `Skipped` com o status detalhado `OverriddenByExplicitInventoryAssociation`
- A associação de Quick Setup continua a coletar inventário de todas as instâncias

Cenário 2: a associação explícita substitui a associação global

- Você tem uma associação global de inventário direcionada a todas as instâncias (não criada pela Quick Setup)
- Você cria uma associação direcionada para uma instância específica
- Resultado: a associação global mostra `Skipped` para as instâncias especificamente direcionadas
- A associação explícita é executada nas instâncias de destino

Etapas de resolução

Se você quiser usar sua própria associação de inventário em vez da Quick Setup:

1. Identificar associações de Quick Setup: no console do Systems Manager, acesse State Manager e procure associações com nomes começando com `AWS-QuickSetup-SSM-CollectInventory-`.
2. Remova a Quick Setup:

- Acesse a Quick Setup no console do Systems Manager.
- Descubra sua configuração de coleta de inventário.
- Exclua a Quick Setup (isso removerá a associação de inventário associada).

> **Note**
>
> Não é necessário excluir manualmente a associação criada pela Quick Setup.

3. Verifique se sua associação é executada: depois de remover a Quick Setup, sua associação explícita de inventário deverá começar a ser executada com êxito.

Se você quiser modificar o comportamento existente:

- Para visualizar as associações de inventário existentes, escolha State Manager no console do Systems Manager e localize associações que usem o documento `AWS-GatherSoftwareInventory` do SSM.
- Lembre-se de que cada nó gerenciado só pode ter uma associação de inventário ativa por vez.

> **Important**
>
> - Os dados de inventário ainda são coletados dos nós ignorados quando a associação de inventário atribuída (de maior prioridade) é executada.
> - As associações de inventário de Quick Setup têm precedência sobre todos os outros tipos, mesmo aquelas com segmentação explícita.
> - A mensagem de status detalhada `OverriddenByExplicitInventoryAssociation` aparece quando qualquer associação é substituída por uma de maior prioridade, independentemente do tipo de associação.

Falha

Se o status da associação de inventário de um nó exibir Failed (Falhou), isso pode significar que o nó tem várias associações de inventário atribuídas a ele. Um nó só pode ter uma associação de inventário atribuída por vez. Uma associação de inventário usa o documento `AWS-GatherSoftwareInventory` AWS Systems Manager (documento SSM). Você pode executar

o seguinte comando usando a AWS Command Line Interface (AWS CLI) para exibir uma lista de associações para um nó.

```
aws ssm describe-instance-associations-status --instance-id instance-ID
```

Falha na conformidade do inventário para uma instância do Amazon EC2

A conformidade do inventário para uma instância do Amazon Elastic Compute Cloud (Amazon EC2) pode falhar se você atribuir várias associações de inventário à instância.

Para resolver esse problema, exclua uma ou mais associações de inventário atribuídas à instância. Para obter mais informações, consulte Excluir uma associação.

> **Note**
>
> Se você criar várias associações de inventário para um nó gerenciado, observe o seguinte comportamento:
>
> - Cada nó pode ser atribuído a uma associação de inventário direcionada a todos os nós (—targets “Key=InstanceIds, Values=*”).
> - Também é possível atribuir cada nó a uma associação específica que usa pares de chave/valor de tag ou um grupo de recursos da AWS.
> - Se um nó tiver várias associações de inventário atribuídas, o status mostrará Skipped (Ignorado) para a associação que não foi executada. A associação que foi executada mais recentemente exibe o status real da associação de inventário.
> - Se um nó for atribuído a várias associações de inventário e cada uma usar um par de chave/valor de tag, essas associações de inventário não serão executadas nesse nó devido ao conflito de tags. A associação ainda será executada em nós que não apresentarem o conflito de chave/valor da tag.

O objeto do bucket do S3 contém dados antigos

Os dados que estão no objeto do bucket do Amazon S3 são atualizados quando a associação do inventário tem êxito e novos dados são descobertos. O objeto do bucket do Amazon S3 é atualizado para cada nó quando a associação é executada e falha, mas os dados que estão no objeto não são atualizados nesse caso. Os dados que estão no objeto do bucket do Amazon S3 serão atualizados

somente quando a associação for executada com êxito. Quando a associação de inventário falhar, você verá dados antigos no objeto do bucket do Amazon S3.

AWS Systems Manager Patch Manager

O Patch Manager, uma ferramenta do AWS Systems Manager, automatiza o processo de aplicação de patches aos nós gerenciados com atualizações de relativas a segurança e outros tipos de atualizações.

> **Note**
>
> O Systems Manager fornece suporte a políticas de patches no Quick Setup, uma ferramenta do AWS Systems Manager. O uso de políticas de patches é o método recomendado para configurar suas operações de patches. Usando uma única configuração de política de patch, é possível definir a aplicação de patches para todas as contas em todas as regiões da sua organização, somente para as contas e regiões que você escolher, ou para um único par de conta-região. Para obter mais informações, consulte Configurações de políticas de patches em Quick Setup.

Você pode usar o Patch Manager para aplicar patches em sistemas operacionais e aplicações. (No Windows Server, o suporte a aplicações é limitado a atualizações de aplicações da Microsoft.) É possível usar o Patch Manager para instalar os Service Packs em nós do Windows e executar atualizações de versões secundárias em nós do Linux. Você pode aplicar patches a frotas de instâncias do Amazon Elastic Compute Cloud (Amazon EC2), a dispositivos de borda, a servidores on-premises e a máquinas virtuais (VMs) por tipo de sistema operacional. Isso inclui versões com suporte de vários sistemas operacionais, conforme listado em Pré-requisitos da Patch Manager. Você pode verificar instâncias para visualizar somente um relatório de patches ausentes ou verificar e instalar automaticamente todos os patches ausentes. Para começar a usar o Patch Manager, abra o Systems Manager console (Console do gerenciador de sistemas). No painel de navegação, escolha Patch Manager.

A AWS não faz testes de patches antes de disponibilizá-los no Patch Manager. Além disso, o Patch Manager não oferece suporte à atualização das principais versões de sistemas operacionais, como Windows Server 2016 a Windows Server 2019 ou SUSE Linux Enterprise Server (SLES) 12.0 para SLES 15.0.

Para tipos de sistema operacional baseados em Linux que relatem um nível de gravidade para patches, o Patch Manager usa o nível de gravidade relatado pelo editor do software para o aviso

de atualização ou patch individual. O Patch Manager não deriva níveis de gravidade de fontes de terceiros, como o Sistema comum de pontuação de vulnerabilidades (CVSS), ou a partir de métricas lançadas pelo Banco de dados nacional de vulnerabilidades (NVD).

Como o Patch Manager beneficia minha organização?

O Patch Manager automatiza o processo de aplicação de patches aos nós gerenciados com atualizações de relativas a segurança e outros tipos de atualizações. Ele oferece vários benefícios principais:

- Controle centralizado de patches: com políticas de patch, você pode configurar operações recorrentes de aplicação de patches para todas as contas em todas as Regiões da sua organização, em contas e Regiões específicas ou em um único par de conta e Região.
- Operações flexíveis de aplicação de patches: é possível optar por verificar as instâncias para visualizar apenas um relatório dos patches ausentes ou verificar e instalar automaticamente todos os patches ausentes.
- Relatórios de conformidade abrangentes: depois de realizar as operações de verificação, é possível visualizar informações detalhadas sobre quais nós gerenciados não estão em conformidade com patches e quais patches estão ausentes.
- Suporte entre iplataformas: o Patch Manager é compatível com vários sistemas operacionais, incluindo diversas distribuições do Linux, macOS e Windows Server.
- Listas de referência de patches personalizadas: é possível definir o que constitui conformidade de patches para a sua organização por meio de listas de referência de patches personalizadas que especificam quais patches estão aprovados para instalação.
- Integração com outros serviços da AWS: o Patch Manager pode ser integrado com o AWS Organizations, o AWS Security Hub, o AWS CloudTrail e o AWS Config para melhorar o gerenciamento e reforçar a segurança.
- Upgrades determinísticos: suporte para upgrades determinísticos por meio de repositórios com versionamento para sistemas operacionais, como o Amazon Linux 2023.

Quem deve usar o Patch Manager?

O Patch Manager foi concebido para os seguintes casos de uso:

- Administradores de TI que têm a necessidade de manter a conformidade dos patches em toda a frota de nós gerenciados.

- Gerentes de operações que exigem visibilidade sobre o status de conformidade dos patches em toda a sua infraestrutura
- Arquitetos de nuvem que pretendem implementar soluções automatizadas para aplicação de patches em grande escala
- Engenheiros de DevOps que precisam integrar a aplicação de patches aos seus fluxos de trabalho de operações
- Organizações que possuem implantações em várias contas/regiões e que necessitam de gerenciamento centralizado de patches
- Qualquer pessoa responsável pela mantenção da postura de segurança e da integridade operacional de nós gerenciados da AWS, dispositivos de borda, servidores on-premises e máquinas virtuais

Quais são os principais recursos do Patch Manager?

O Patch Manager fornece vários recursos principais:

- Políticas de patches: configure operações de aplicação de patches em várias Regiões e Contas da AWS utilizando uma única política por meio da integração com o AWS Organizations.
- Listas de referência de patches personalizadas: defina regras de aprovação automática de patches alguns dias após o lançamento dos mesmos, bem como listas de patches aprovados e rejeitados.
- Diversos métodos de aplicação de patches: escolha entre políticas de patches, janelas de manutenção ou operações sob demanda para aplicação de patches a fim de atender às suas necessidades específicas.
- Relatórios de conformidade: gere relatórios detalhados sobre o status de conformidade dos patches, que podem ser enviados a um bucket do Amazon S3 no formato CSV.
- Suporte entre plataformas: aplique patches em sistemas operacionais e aplicações no Windows Server, várias distribuições do Linux e no macOS.
- Flexibilidade de programação: defina programações diferentes para verificar e instalar patches usando expressões CRON ou Rate personalizadas.
- Ganchos do ciclo de vida: execute scripts personalizados antes e depois de operações de aplicação de patches usando documentos do Systems Manager.
- Foco na segurança: por padrão, o foco do Patch Manager está em atualizações de segurança em vez de na instalação de todos os patches disponíveis.

- Controle de taxa: configure limites de simultaneidade e erro para operações de aplicação de patches a fim de minimizar o impacto sobre as operações.

O que é conformidade no Patch Manager?

A referência para o que constitui a conformidade de patches para os nós gerenciados em suas frotas do Systems Manager não é definida pela AWS, pelos fornecedores de sistemas operacionais (SO) ou por terceiros, como empresas de consultoria em segurança.

Em vez disso, você define o que significa conformidade de patches para nós gerenciados em sua organização ou conta em uma lista de referência de patches. Uma lista de referência de patches é uma configuração que especifica regras para a instalação de patches em um nó gerenciado. Um nó gerenciado é compatível com patches quando está atualizado com todos os patches que atendem aos critérios de aprovação que você especifica na lista de referência de patches.

Observe que estar em conformidade com uma lista de referência de patches não significa que um nó gerenciado seja necessariamente seguro. Em conformidade significa que os patches definidos pela lista de referência de patches que estão disponíveis e aprovados foram instalados no nó. A segurança geral de um nó gerenciado é determinada por muitos fatores fora do escopo do Patch Manager. Para obter mais informações, consulte Segurança.

Cada lista de referência de patches é uma configuração para um tipo específico de sistema operacional (SO) compatível, como Red Hat Enterprise Linux (RHEL), macOS ou Windows Server. Uma lista de referência de patches pode definir regras de patches para todas as versões compatíveis de um sistema operacional ou ser limitada somente às que você especificar, como RHEL 6.10, RHEL 7.8. e RHEL 9.3.

Em uma lista de referência de patches, você pode especificar que todos os patches de determinadas classificações e níveis de severidade sejam aprovados para instalação. Por exemplo, você pode incluir todos os patches classificados como `Security`, mas excluir outras classificações, como `Bugfix` ou `Enhancement`. E você pode incluir todos os patches com uma severidade de `Critical` e excluir outros, como `Important` e `Moderate`.

Você também pode definir patches explicitamente em uma lista de referência de patches adicionando suas IDs às listas de patches específicos a serem aprovados ou rejeitados, como `KB2736693` para Windows Server ou `dbus.x86_64:1:1.12.28-1.amzn2023.0.1` para Amazon Linux 2023 (AL2023). Opcionalmente, você pode especificar um determinado número de dias de espera para a aplicação de patches após a disponibilização de um patch. Para Linux e macOS, você tem a opção

de especificar uma lista externa de patches para fins de conformidade (uma lista de substituição de instalação) em vez daquelas definidas pelas regras da lista de referência de patches.

Quando uma operação de aplicação de patches é executada, o Patch Manager compara os patches atualmente aplicados a um nó gerenciado com aqueles que devem ser aplicados de acordo com as regras definidas na lista de referência de patches ou na lista de substituição de instalação. É possível optar para que o Patch Manager mostre somente um relatório de patches que estejam faltando (uma operação `Scan`) ou que o Patch Manager instale automaticamente todos os patches que descobrir estarem faltando em um nó gerenciado (uma operação `Scan and install`).

O Patch Manager fornece listas de referência de patches predefinidas que você pode usar para suas operações de aplicação de patches; no entanto, essas configurações predefinidas são fornecidas como exemplos e não como práticas recomendadas. Recomendamos que você crie suas próprias listas de referência de patches personalizadas para exercer maior controle sobre o que constitui a conformidade de patches para sua frota.

Para obter mais informações sobre a listas de referência de patches personalizadas, consulte os seguintes tópicos:

- Listas de referência de patches predefinidas e personalizadas
- Sobre formatos de nomes de pacotes para listas de patches aprovados e rejeitados
- Visualizar as listas de referência de patches predefinidas da AWS
- Trabalhando com linhas de base de patch personalizadas
- Trabalhando com relatórios de conformidade de patch

Componentes primários

Antes de começar a trabalhar com a ferramenta Patch Manager, você deve se familiarizar com alguns dos principais componentes e atributos das operações de aplicação de patches da ferramenta.

Linhas de base de patch

O Patch Manager usa listas de referência de patches, que incluem regras para a aprovação automática de patches poucos dias após seus lançamentos, bem como listas opcionais de patches aprovados e rejeitados. Quando uma operação de aplicação de patches é executada, o Patch Manager compara os patches atualmente aplicados a um nó gerenciado com aqueles que devem ser

aplicados de acordo com as regras definidas na lista de referência de patches. É possível optar para que o Patch Manager mostre somente um relatório de patches que estejam faltando (uma operação `Scan`) ou que o Patch Manager instale automaticamente todos os patches que descobrir estarem faltando em um nó gerenciado (uma operação `Scan and install`).

Métodos de operação de aplicação de patches

O Patch Manager atualmente oferece quatro métodos para a execução das operações `Scan` e `Scan and install`:

- (Recomendado) Uma política de patch configurada em Quick Setup: com base na integração com o AWS Organizations, uma única política de patch pode definir programações de aplicação de patches e listas de referência de patches para uma organização inteira, incluindo várias Contas da AWS e todas as Regiões da AWS em que essas contas operem. Uma política de patch também pode ter como destino somente algumas unidades organizacionais (UOs) em uma organização. É possível usar uma única política de patch para verificar e instalar em horários diferentes. Para obter mais informações, consulte Configurar a aplicação de patches para instâncias em uma organização usando uma política de patch do Quick Setup e Configurações de políticas de patches em Quick Setup.
- Uma opção do Host Management configurada em Quick Setup: também há suporte para as configurações do Host Management na integração com o AWS Organizations, possibilitando a execução de uma operação de aplicação de patches até para uma organização inteira. No entanto, essa opção se limita à verificação de patches ausentes usando a lista de referência de patches atual padrão e ao fornecimento de resultados em relatórios de conformidade. Esse método de operação não pode instalar patches. Para obter mais informações, consulte Configurar o gerenciamento de host do Amazon EC2 usando a Quick Setup.
- Uma janela de manutenção para executar uma tarefa de **`Scan`** ou **`Install`** patches: uma janela de manutenção, que você configura na ferramenta do Systems Manager chamada Maintenance Windows, pode ser configurada para executar diferentes tipos de tarefas em um agendamento definido por você. Uma tarefa do tipo Run Command pode ser usada para executar tarefas `Scan` ou `Scan and install` em um conjunto de nós gerenciados que você escolher. Cada tarefa da janela de manutenção pode ter como destino os nós gerenciados em apenas um único par Conta da AWS-Região da AWS. Para obter mais informações, consulte Tutorial: Create a maintenance window for patching using the console.
- Uma operação Aplicar patch agora sob demanda em Patch Manager: a opção Aplicar patch agora permite que você ignore as configurações de programação quando você precisar corrigir os nós gerenciados o mais rápido possível. Usando Patch now (Aplicar patch agora), você especifica

se deseja executar a operação `Scan` ou `Scan and install` e em quais nós gerenciados executar a operação. Você também pode optar por executar documentos do Systems Manager (documentos SSM) como ganchos do ciclo de vida durante a operação de aplicação de patches. Cada operação Patch Now (Aplicar patch agora) pode atingir nós gerenciados em apenas um único par Conta da AWS-Região da AWS. Para obter mais informações, consulte Aplicação de patches em nós gerenciados sob demanda.

Relatórios de conformidade

Depois de uma operação `Scan`, é possível usar o console do Systems Manager para visualizar informações sobre quais dos seus nós gerenciados estão fora de conformidade com o patch e quais patches estão faltando em cada um desses nós. Você também pode gerar relatórios de conformidade de patches no formato .csv que serão enviados a um bucket do Amazon Simple Storage Service (Amazon S3) de sua escolha. Você pode gerar relatórios únicos ou gerar relatórios em uma programação regular. Para um único nó gerenciado, os relatórios incluem detalhes de todos os patches para o nó. Para obter um relatório sobre todas os nós gerenciados, apenas um resumo de quantos patches estão ausentes é fornecido. Depois que um relatório é gerado, você pode usar uma ferramenta, como o Amazon QuickSight, para importar e analisar os dados. Para obter mais informações, consulte Trabalhando com relatórios de conformidade de patch.

> **Note**
>
> Um item de conformidade gerado por meio do uso de uma política de patch tem um tipo de execução de `PatchPolicy`. Um item de conformidade não gerado em uma operação de política de patch tem um tipo de execução de `Command`.

Integrações

O Patch Manager se integra com os outros seguintes Serviços da AWS:

- AWS Identity and Access Management (IAM): use o IAM para controlar quais usuários, grupos e funções têm acesso às operações do Patch Manager. Para obter mais informações, consulte Como o AWS Systems Manager funciona com o IAM e Configurar permissões de instância obrigatórias para o Systems Manager.
- AWS CloudTrail: use o CloudTrail para registrar um histórico auditável de eventos de operação de aplicação de patches iniciados por usuários, perfis ou grupos. Para obter mais informações, consulte Registrar em log chamadas de API do AWS Systems Manager com o AWS CloudTrail.

- AWS Security Hub: dados de conformidade de patches do Patch Manager pode ser enviado para o AWS Security Hub. O Security Hub oferece uma visão abrangente dos alertas de segurança de alta prioridade e do status de conformidade. Também monitora o estado de aplicação de patches da sua frota. Para obter mais informações, consulte Integrar o Patch Manager ao AWS Security Hub.
- AWS Config: configure a gravação no AWS Config para visualizar os dados de gerenciamento de instâncias do Amazon EC2 no painel do Patch Manager. Para obter mais informações, consulte Visualizar resumos do painel de patches.

Tópicos

- Configurações de políticas de patches em Quick Setup
- Pré-requisitos da Patch Manager
- Como operações do Patch Manager funcionam
- Documentos do comando do SSM para aplicação de patches em nós gerenciados
- Linhas de base de patch
- Usar o Kernel Live Patching em nós gerenciados do Amazon Linux 2
- Trabalhar com recursos e conformidade do Patch Manager usando o console
- Trabalhar com recursos do Patch Manager usando a AWS CLI
- Tutoriais do AWS Systems Manager Patch Manager
- Solução de problemas do Patch Manager

Configurações de políticas de patches em Quick Setup

A AWS recomenda o uso de políticas de patch para configurar a aplicação de patches para sua organização e Contas da AWS. As políticas de patch foram introduzidas no Patch Manager em dezembro de 2022.

Uma política de patch é uma configuração que você define usando o Quick Setup, uma ferramenta do AWS Systems Manager. As políticas de patch fornecem um controle mais amplo e centralizado sobre suas operações de aplicação de patch do que o disponível com os métodos anteriores de configuração de patches. As políticas de patch podem ser usadas com todos os sistemas operacionais com suporte pelo Patch Manager, incluindo versões do Linux, macOS, e Windows Server com suporte. Para obter instruções sobre como criar uma política de patches, consulte

Configurar a aplicação de patches para instâncias em uma organização usando uma política de patch do Quick Setup.

Principais características das políticas de patch

Em vez de usar outros métodos para corrigir seus nós, use uma política de patch para aproveitar esses recursos principais:

- Configuração única: a configuração de operações de aplicação de patches usando uma janela de manutenção ou uma associação do State Manager pode exigir várias tarefas em diferentes partes do console do Systems Manager. Usando uma política de patch, todas as suas operações de aplicação de patch podem ser configuradas em um único assistente.
- Suporte para várias contas/várias regiões: usando uma janela de manutenção, uma associação do State Manager ou o recurso Patch Now do Patch Manager, você está limitado a visar nós gerenciados em um único par Conta da AWS-Região da AWS. Se você usa várias contas e várias regiões, suas tarefas de configuração e manutenção podem exigir muito tempo, pois você precisa executar tarefas de configuração em cada par conta-região. No entanto, se você usar o AWS Organizations, poderá configurar uma política de patch que se aplique a todos os seus nós gerenciados em todas as Regiões da AWS, em todas as suas Contas da AWS. Ou, se você preferir, uma política de patch pode ser aplicada somente a algumas unidades organizacionais (UOs) nas contas e regiões que você escolher. Uma política de patch também pode ser aplicada a uma única conta local, se você preferir.
- Suporte de instalação no nível organizacional: a opção de configuração existente do Host Management em Quick Setup fornece suporte a uma verificação diária de seus nós gerenciados para verificar a conformidade dos patches. No entanto, essa verificação é feita em um horário predeterminado e resulta somente em informações de conformidade dos patches. Nenhuma instalação de patch é executada. Com uma política de patch, é possível especificar diferentes programações de verificação e instalação. Você também pode escolher a frequência e a hora dessas operações usando expressões personalizadas do CRON ou Rate. Por exemplo, é possível verificar todos os dias se há patches faltando, para fornecer informações de conformidade atualizadas regularmente. Porém, sua programação de instalação pode ser de apenas uma vez por semana, para evitar períodos de inatividade indesejados.
- Seleção simplificada da lista de referência de patches: as políticas de patch ainda incorporam listas de referência de patches, e não há alterações na forma como as lista de referência de patches são configuradas. No entanto, ao criar ou atualizar uma política de patch, é possível selecionar a lista de referência gerenciada da AWS ou a personalizada que desejar usar para

cada tipo de sistema operacional (SO) em uma única lista. Não é necessário especificar a lista de referência padrão para cada tipo de sistema operacional em tarefas separadas.

> **Note**
>
> Quando as operações de aplicação de patch baseadas em uma política de patch são executadas, elas usam o documento SSM da `AWS-RunPatchBaseline`. Para ter mais informações, consulte Documento de comando do SSM para a aplicação de patches: `AWS-RunPatchBaseline`.

Informações relacionadas

Implante centralmente as operações de aplicação de patches em toda a sua organização da AWS usando o Quick Setup do Systems Manager (blog de operações e migrações para a nuvem da AWS)

Outras diferenças com as políticas de patches

Aqui estão algumas outras diferenças a serem observadas ao usar as políticas de patch em vez de os métodos anteriores de configuração de patches:

- Não são necessários grupos de patches: em operações de aplicação de patches anteriores, você podia marcar vários nós para pertencerem a um grupo de patches e, em seguida, especificar a lista de referência de patches a ser usada para esse grupo de patches. Se nenhum grupo de patches tivesse sido definido, o Patch Manager aplicaria patch nas instâncias com a lista de referência de patches padrão atual do tipo de sistema operacional. Usando políticas de patch, não é mais necessário configurar e manter grupos de patches.

  > **Note**
  >
  > Não há suporte no console à funcionalidade de grupos de patches para pares de conta-região que ainda não usavam grupos de patches antes do lançamento do suporte à política de patches em 22 de dezembro de 2022. A funcionalidade de grupo de patches ainda está disponível em pares de conta-região que começaram a usar grupos de patches antes dessa data.

- Página “Configure patching” (Configurar a aplicação de patches) removida: antes do lançamento das políticas de patch, você podia especificar padrões para quais nós aplicar patches, uma

programação de patches e uma operação de aplicação de patches em uma página Configure patching (Configurar a aplicação de patches). Esta página foi removida do Patch Manager. Essas opções agora estão especificadas nas políticas de patch.

- Sem suporte para “Patch Now”: a capacidade de aplicar patches em nós sob demanda ainda está limitada a um único par Conta da AWS-Região da AWS por vez. Para ter mais informações, consulte Aplicação de patches em nós gerenciados sob demanda.
- Políticas de patch e informações de conformidade: quando seus nós gerenciados são varridos quanto à conformidade de acordo com uma configuração de política de aplicação de patches, os dados de conformidade são disponibilizados para você. É possível visualizar e trabalhar com os dados da mesma forma que com outros métodos de verificação de conformidade. Embora você possa configurar uma política de patch para uma organização inteira ou várias unidades organizacionais, as informações de conformidade são relatadas individualmente para cada par Conta da AWS-Região da AWS. Para ter mais informações, consulte Trabalhando com relatórios de conformidade de patch.
- Status de conformidade da associação e políticas de patch: o status de patch de um nó gerenciado que está sob uma política de patch de Quick Setup corresponde ao status de execução da associação do State Manager para esse nó. Se o status de execução da associação for `Compliant`, o status de patch do nó gerenciado também será marcado como `Compliant`. Se o status de execução da associação for `Non-Compliant`, o status de patch do nó gerenciado também será marcado como `Non-Compliant`.

Regiões da AWS com suporte para políticas de patch

No momento, as configurações de política de patch do Quick Setup são compatíveis nas seguintes regiões:

- Leste dos EUA (Ohio) (us-east-2)
- Leste dos EUA (Norte da Virgínia) (us-east-1)
- Oeste dos EUA (Norte da Califórnia) (us-west-1)
- Oeste dos EUA (Oregon) (us-west-2)
- Ásia-Pacífico (Mumbai) (ap-south-1)
- Ásia-Pacífico (Seul) (ap-northeast-2)
- Ásia-Pacífico (Singapura) (ap-southeast-1)
- Ásia-Pacífico (Sydney) (ap-southeast-2)
- Ásia Pacific (Tóquio) (ap-northeast-1)

- Canadá (Central) (ca-central-1)
- Europa (Frankfurt) (eu-central-1)
- Europa (Irlanda) (eu-west-1)
- Europa (Londres) (eu-west-2)
- Europa (Paris) (eu-west-3)
- UE (Estocolmo) (eu-north-1)
- América do Sul (São Paulo) (sa-east-1)

Pré-requisitos da Patch Manager

Certifique-se de ter atendido aos pré-requisitos necessários antes de usar o Patch Manager, uma ferramenta do AWS Systems Manager.

Tópicos

- Versão do SSM Agent
- Versão do Python
- Conectividade com a origem do patch
- Acesso do endpoint do S3
- Permissões para instalar patches localmente
- Sistemas operacionais compatíveis com o Patch Manager

Versão do SSM Agent

A versão 2.0.834.0 ou posterior do SSM Agent deve estar em execução nos nós gerenciados que você quiser gerenciar com o Patch Manager.

> **Note**
>
> Uma versão atualizada do SSM Agent é lançada sempre que novas ferramentas são adicionadas ao Systems Manager ou sempre que atualizações são feitas nas ferramentas existentes. Deixar de usar a versão mais recente do agente pode impedir que seu nó gerenciado use várias ferramentas do Systems Manager. Por isso, recomendamos automatizar o processo de manter o SSM Agent atualizado em suas máquinas. Para mais informações, consulte Automatizar atualizações do SSM Agent. Inscreva-se na página Notas

de versão do SSM Agent no GitHub para receber notificações sobre atualizações do SSM Agent.

Versão do Python

Para o macOS e a maioria dos sistemas operacionais (SOs) Linux, o Patch Manager é atualmente compatível com o Python versões 2.6-3.10. O SOs AlmaLinux, Debian Server, Raspberry Pi OS e Ubuntu Server exigem uma versão compatível do Python 3 (3.0-3.10).

Conectividade com a origem do patch

Se os nós gerenciados não tiverem uma conexão direta com a Internet e você estiver usando uma Amazon Virtual Private Cloud (Amazon VPC) com um endpoint da VPC, verifique se os nós têm acesso aos repositórios de patch de origem (repositórios). Em nós do Linux, as atualizações de patches normalmente são baixadas dos repositórios remotos configurados em seu nó. Portanto, o nó deve conseguir se conectar aos repositórios para que a aplicação do patch seja realizada. Para obter mais informações, consulte Como os patches de segurança são selecionados.

CentOS e CentOS Stream: habilite a sinalização **`EnableNonSecurity`**

Os nós gerenciados do CentOS 6 e 7 usam o Yum como gerenciador de pacotes. Os nós do CentOS 8 e CentOS Stream usam o DNF como gerenciador de pacotes. Ambos os gerenciadores de pacotes usam o conceito de um aviso de atualização. Uma notificação de atualização é um conjunto de pacotes que corrige problemas específicos.

No entanto, os repositórios padrão do CentOS e CentOS Stream não são configurados com um aviso de atualização. Isso significa que o Patch Manager não detecta pacotes em repositórios padrão do CentOS e CentOS Stream. Para permitir que o Patch Manager processe pacotes que não estão contidos em um aviso de atualização, você deve ativar o sinalizador `EnableNonSecurity` nas regras da lista de referência de patches.

Windows Server: garanta a conectividade ao Windows Update Catalog ou ao Windows Server Update Services (WSUS)

Os nós gerenciados do Windows Server devem se conectar ao Windows Update Catalog ou ao Windows Server Update Services (WSUS). Confirme se os nós têm conectividade com o Catálogo de atualizações da Microsoft por meio de um gateway da Internet, um gateway de NAT ou uma instância NAT. Se você estiver usando o WSUS, confirme se o nó tem conectividade com o servidor WSUS

em seu ambiente. Para obter mais informações, consulte Problema: o nó gerenciado não tem acesso ao Catálogo do Windows Update ou ao WSUS.

Acesso do endpoint do S3

Se os nós gerenciados operam em uma rede privada ou pública, sem acesso aos buckets do Amazon Simple Storage Service (Amazon S3) gerenciados pela AWS, as operações de aplicação de patches falham. Para obter informações sobre os buckets do S3 que os nós gerenciados devem poder acessar, consulte Comunicações do SSM Agent com os buckets do S3 gerenciados pela AWS e Melhorar a segurança das instâncias do EC2 usando endpoints da VPC para o Systems Manager.

Permissões para instalar patches localmente

Nos sistemas operacionais Windows Server e Linux, o Patch Manager assume as contas de administrador e usuário-raiz, respectivamente, para instalar os patches.

No entanto, para Brew e Brew Cask no macOS, o Homebrew não oferece suporte à execução de seus comandos sob o usuário-raiz. Como resultado, o Patch Manager consulta e executa comandos Homebrew como o proprietário do diretório Homebrew ou como um usuário válido pertencente ao grupo de proprietários do diretório Homebrew. Portanto, para instalar os patches, o proprietário do diretório `homebrew` também precisa de permissões recursivas para o diretório `/usr/local`.

> **Tip**
>
> O comando a seguir fornece essa permissão para o usuário especificado:
>
> ```
> sudo chown -R $USER:admin /usr/local
> ```

Sistemas operacionais compatíveis com o Patch Manager

A ferramenta Patch Manager não é compatível com todas as versões de sistemas operacionais que são compatíveis com outras ferramentas do Systems Manager. Por exemplo, o Patch Manager não oferece suporte para CentOS 6.3 ou Raspberry Pi OS 8 (Jessie). (Para obter a lista completa de sistemas operacionais compatíveis com o Systems Manager, consulte Sistemas operacionais compatíveis com o Systems Manager.) Portanto, verifique se os nós gerenciados utilizados com o Patch Manager estão executando um dos sistemas operacionais listados na tabela a seguir.

> **Note**
>
> O Patch Manager depende dos repositórios de patches configurados em um nó gerenciado, como o Catálogo do Windows Update e o Windows Server Update Services para Windows, para recuperar os patches disponíveis para instalação. Portanto, para versões de sistema operacional de fim de vida útil (EOL), se nenhuma nova atualização estiver disponível, o Patch Manager talvez não consiga relatar as novas atualizações. Isso pode ocorrer porque nenhuma nova atualização foi lançada pelo mantenedor da distribuição Linux, pela Microsoft ou pela Apple, ou porque o nó gerenciado não tem a licença adequada para acessar as novas atualizações.
>
> O Patch Manager relata o status de conformidade em relação aos patches disponíveis no nó gerenciado. Portanto, se uma instância estiver executando um sistema operacional em EOL e nenhuma atualização estiver disponível, o Patch Manager poderá relatar o nó como Em conformidade, dependendo das linhas de referência do patch configuradas para a operação de patch.

Sistema operacional	Detalhes
Linux	• AlmaLinux 8.x, 9.x • Amazon Linux 2012.03-2018.03 • Amazon Linux 2 versão 2.0 e todas as versões posteriores • Amazon Linux 2022 • Amazon Linux 2023 • CentOS 6.5 – 7.9, 8.x • CentOS Stream 8, 9 • Debian Server 8.x, 9.x, 10.x, 11.x e 12.x • Oracle Linux 7.5 – 8.x, 9.x • Raspberry Pi OS (antigo Raspbian) 9 (Stretch) • Red Hat Enterprise Linux (RHEL) 6.5 – 8.x, 9.x, 10x • Rocky Linux 8.x, 9.x

Sistema operacional	Detalhes
	• SUSE Linux Enterprise Server (SLES) 12.0 e versões 12.x posteriores; 15.x • Ubuntu Server 14.04 LTS, 16.04 LTS, 18.04 LTS, 20.04 LTS, 20.10 STR, 22.04 LTS, 23.04, 23.10, 24.04, 24.10 e 25.04

Sistema operacional	Detalhes
macOS	O macOS é compatível somente com instâncias do Amazon EC2. 11.3.1; 11.4-11.7 (Big Sur) 12.0 – 12.7 (Monterey) 13.0 – 13.7 (Ventura) 14.x (Sonoma) 15.x (Sequoia) macOS atualizações do sistema operacional O Patch Manager não oferece suporte a atualizações ou upgrades do sistema operacional (SO) macOS, como do 12.x para 13.x ou 13.1 para 13.2. Para realizar atualizações de versão do sistema operacional macOS, recomendamos usar os mecanismos integrados de atualização do sistema operacional da Apple. Para obter mais informações, consulte Gerenciamento de dispositivos no site da Apple Developer Documentation. Suporte do Homebrew O Patch Manager requer que o Homebrew, o sistema de gerenciamento de pacotes de software de código aberto, seja instalado em um dos seguintes locais de instalação padrão: • `/usr/local/bin` • `/opt/homebrew/bin`

Sistema operacional	Detalhes
	As operações de aplicação de patches usando o Patch Manager não funcionam corretamente quando o Homebrew não está instalado. O Homebrew descontinuou o suporte para macOS 10.14.x (Mojave) e 10.15.x (Catalina). Como resultado, as operações de aplicação de patches usando o Patch Manager não são compatíveis nessas versões. Suporte de região Não há suporte para macOS em todas as Regiões da AWS. Para obter mais informaçõ es sobre o suporte a instâncias do EC2 para macOS, consulte Instâncias Mac do Amazon EC2 no Guia do usuário do Amazon EC2. macOS dispositivos de borda O SSM Agent para dispositivos principais do AWS IoT Greengrass não é compatível com o macOS. Você não pode usar o Patch Manager para aplicar patches aos dispositivos de borda do macOS.

Sistema operacional	Detalhes
Windows	Windows Server 2008 a Windows Server 2025, incluindo versões R2. **Note** O SSM Agent para dispositivos principais do AWS IoT Greengrass não é compatível com o Windows 10. Você não pode usar o Patch Manager para aplicar patches aos dispositivos de borda do Windows 10. Suporte de 2008 do Windows Server A partir de 14 de janeiro de 2020, o Windows Server 2008 não é mais compatível para obter recursos ou atualizações de segurança da Microsoft. As Amazon Machine Images (AMIs) herdadas para Windows Server 2008 e 2008 R2 ainda incluem a versão 2 do SSM Agent pré-instalada, mas o Systems Manager não é oficialmente compatível com as versões 2008 e não atualiza mais o agente para essas versões do Windows Server. Além disso, o SSM Agent versão 3 pode não ser compatível com todas as operações no Windows Server 2008 e 2008 R2. A versão final do SSM Agent oficialmente compatível com as versões 2008 do Windows Server é a 2.3.1644.0. Suporte ao Windows Server 2012 e 2012 R2 O Windows Server 2012 e 2012 R2 atingiram o fim do suporte em 10 de outubro de 2023. Para usar o Patch Manager com essas versões,

Sistema operacional	Detalhes
	recomendamos também usar as Extended Security Updates (ESUs) da Microsoft. Para obter mais informações, consulte Fim do suporte ao Windows Server 2012 e 2012 R2 no site da Microsoft.

Como operações do Patch Manager funcionam

Esta seção fornece detalhes técnicos que explicam como o Patch Manager, uma ferramenta do AWS Systems Manager, define quais patches devem ser instalados e como ele os instala em cada sistema operacional compatível. Para sistemas operacionais Linux, ele também fornece informações sobre como especificar um repositório de origem, em uma lista personalizada de referência de patches, para patches diferentes do padrão configurado em um nó gerenciado. Esta seção também fornece detalhes sobre como as regras de linha de base de patch funcionam em diferentes distribuições do sistema operacional Linux.

> **Note**
>
> As informações nos tópicos a seguir se aplicam independentemente do método ou tipo de configuração que você estiver usando para suas operações de aplicação de patch:
>
> - Uma política de patch configurada no Quick Setup
> - Uma opção do Host Management configurada no Quick Setup
> - Uma janela de manutenção para executar um patch `Scan` ou tarefa `Install`
> - Uma operação Patch now (Aplicar patch agora) sob demanda

Tópicos

- Como as datas de lançamento e atualização de pacotes são calculadas
- Como os patches de segurança são selecionados
- Como especificar um repositório de origem de patches alternativo (Linux)
- Como os patches são instalados
- Como funcionam as regras de linha de base de patch em sistemas baseados no Linux
- Principais diferenças entre a aplicação de patches em nós do Linux e em nós do Windows Server

Como as datas de lançamento e atualização de pacotes são calculadas

> **Important**
>
> As informações nesta página se aplicam aos sistemas operacionais (SOs) Amazon Linux 1, Amazon Linux 2, Amazon Linux 2022 e Amazon Linux 2023 para instâncias do Amazon Elastic Compute Cloud (Amazon EC2). Os pacotes para esses tipos de SOs são criados e mantidos pela Amazon Web Services. A forma como os fabricantes de outros SOs gerenciam seus pacotes e repositórios afeta a forma como as datas de lançamento e de atualização deles são calculadas. Para sistemas operacionais além de Amazon Linux, Amazon Linux 2, Amazon Linux 2022 e Amazon Linux 2023, como Red Hat Enterprise Linux (RHEL) e SUSE Linux Enterprise Server (SLES), consulte a documentação do fabricante para obter informações sobre como seus pacotes são atualizados e mantidos.

Nas configurações da lista de referência de patches personalizada que você cria, para a maioria dos tipos de SO, é possível especificar que os patches sejam aprovados automaticamente para instalação após um determinado número de dias. O AWS fornece várias listas de referência de patches predefinidas que incluem datas de aprovação automática de sete dias.

Um atraso de aprovação automática é o número de dias para aguardar após o lançamento do patch, antes que a aplicação do patch seja aprovada automaticamente. Por exemplo, você cria uma regra usando a classificação `CriticalUpdates` e a configura para um atraso de aprovação automática de 7 dias. Como resultado, um novo patch crítico com data de lançamento ou data da última atualização em 7 de julho será aprovado automaticamente em 14 de julho.

Para evitar resultados inesperados com atrasos na aprovação automática no Amazon Linux 1, no Amazon Linux 2, no Amazon Linux 2022 e no Amazon Linux 2023, é importante entender como as datas de lançamento e de atualização desses serviços são calculadas.

Na maioria dos casos, o tempo de espera de aprovação automática antes da instalação dos patches é calculado com base em um valor `Updated Date` em `updateinfo.xml`, e não um valor `Release Date`. Veja a seguir detalhes importantes sobre esses cálculos de data:

- `Release Date` é a data de lançamento de uma notificação. Isso não significa que o pacote já esteja necessariamente disponível nos repositórios associados.
- `Update Date` é a última data em que a notificação foi atualizada. Uma atualização de uma notificação pode representar algo tão pequeno quanto uma atualização de texto ou descrição. Isso

não significa que o pacote tenha sido liberado nessa data ou já esteja necessariamente disponível nos repositórios associados.

Isso significa que um pacote pode ter um valor `Update Date` de 7 de julho, mas não estará disponível para instalação até (p. ex.) 13 de julho. Nesse caso, suponha que uma lista de referência de patches que especifique um atraso de aprovação automática de 7 dias seja executada em uma operação `Install` em 14 de julho. Como o valor `Update Date` é de 7 dias antes da data de execução, os patches e atualizações no pacote serão instalados em 14 de julho. A instalação acontece mesmo que apenas 1 dia tenha passado desde que o pacote foi disponibilizado para instalação efetiva.

- Um pacote contendo patches de sistema operacional ou aplicativo pode ser atualizado mais de uma vez após o lançamento inicial.
- Um pacote pode ser lançado nos repositórios gerenciados da AWS e ser revertido se houver a descoberta de problemas posteriormente.

Em algumas operações de aplicação de patches, talvez esses fatores não sejam importantes. Por exemplo, se uma lista de referência de patches estiver configurada para instalar um patch com valores de severidade de `Low` e `Medium`, e uma classificação de `Recommended`, qualquer atraso na aprovação automática pode ter pouco impacto em suas operações.

No entanto, em casos nos quais o tempo de patches críticos ou de alta severidade é mais importante, pode ser necessário que você exerça mais controle sobre quando os patches são instalados. O método recomendado para fazer isso é usar repositórios alternativos de origem de patch em vez dos repositórios padrão para operações de patch em um nó gerenciado.

Você pode especificar os repositórios de origem de patches alternativos ao criar uma linha de base de patch personalizada. Em cada linha de base de patch personalizada, você pode especificar as configurações de origem de patches para até 20 versões de um sistema operacional Linux compatível. Para ter mais informações, consulte Como especificar um repositório de origem de patches alternativo (Linux).

Como os patches de segurança são selecionados

O foco principal do Patch Manager, uma ferramenta do AWS Systems Manager, é instalar as atualizações relacionadas à segurança de sistemas operacionais nos nós gerenciados. Por padrão, o Patch Manager não instala todos os patches disponíveis, mas sim um conjunto menor de patches relacionados à segurança.

Por padrão, o Patch Manager não substitui um pacote que foi marcado como obsoleto em um repositório de pacotes por pacotes de substituição de nome diferente, a menos que essa substituição seja necessário pela instalação de outra atualização de pacote. Em vez disso, para comandos que atualizam um pacote, o Patch Manager somente informa e instala atualizações ausentes do pacote instalado, mas obsoleto. Isso porque a substituição de um pacote obsoleto normalmente exige a desinstalação de um pacote existente e a instalação do substituto. A substituição do pacote obsoleto pode introduzir alterações de impacto ou funcionalidades adicionais não pretendidas.

Esse comportamento é consistente com o comando `update-minimal` do YUM e DNF, cujo foco está em atualizações de segurança em vez de em atualizações de recursos. Para obter mais informações, consulte the section called “Como os patches são instalados”.

Para tipos de sistema operacional baseados em Linux que relatem um nível de gravidade para patches, o Patch Manager usa o nível de gravidade relatado pelo editor do software para o aviso de atualização ou patch individual. O Patch Manager não deriva níveis de gravidade de fontes de terceiros, como o Sistema comum de pontuação de vulnerabilidades (CVSS), ou a partir de métricas lançadas pelo Banco de dados nacional de vulnerabilidades (NVD).

> **Note**
>
> Em todos os sistemas baseados em Linux com os quais o Patch Manager é compatível, você pode escolher um outro repositório de origem configurado para o nó gerenciado, normalmente para instalar atualizações não relacionadas a segurança. Para mais informações, consulte Como especificar um repositório de origem de patches alternativo (Linux).

O restante desta seção explica como o Patch Manager seleciona patches de segurança para os diferentes sistemas operacionais compatíveis.

Amazon Linux 1, Amazon Linux 2, Amazon Linux 2022, and Amazon Linux 2023

O Amazon Linux 1 e o Amazon Linux 2 lidam com repositórios pré-configurados de modo diferente do Amazon Linux 2022 e do Amazon Linux 2023.

No Amazon Linux 1 e no Amazon Linux 2, o serviço de lista de referência de patches do Systems Manager usa repositórios pré-configurados em seu nó gerenciado. Há dois repositórios pré-configurados (repos) em um nó:

No Amazon Linux 1

- ID do repositório: `amzn-main/latest`

  Nome do repositório: `amzn-main-Base`
- ID do repositório: `amzn-updates/latest`

  Nome do repositório: `amzn-updates-Base`

No Amazon Linux 2

- ID do repositório: `amzn2-core/2/`*`architecture`*

  Nome do repositório: `Amazon Linux 2 core repository`
- ID do repositório: `amzn2extra-docker/2/`*`architecture`*

  Nome do repositório: `Amazon Extras repo for docker`

> **Note**
>
> *`arquitetura`* pode ser x86_64 ou aarch64.

Quando você cria uma instância do Amazon Linux 2023 (AL2023), ela contém as atualizações que estavam disponíveis na versão do AL2023 e na AMI específica que você selecionou. A instância do AL2023 não recebe automaticamente outras atualizações de segurança críticas e importantes durante a inicialização. Em vez disso, com o recurso de atualizações determinísticas por meio de repositórios versionados compatível com o AL2023, o qual está ativado por padrão, é possível aplicar atualizações com base em uma programação que atenda às suas necessidades específicas. Para obter mais informações, consulte Deterministic upgrades through versioned repositories no Amazon Linux 2023 User Guide.

No Amazon Linux 2022, os repositórios pré-configurados estão vinculados a versões vinculadas de atualizações de pacotes. Quando novas Amazon Machine Images (AMIs) para o Amazon Linux 2022 são lançadas, elas são vinculadas a uma versão específica. Para atualizações de patches, o Patch Manager recupera a versão vinculada mais recente do repositório de atualizações de patches e, em seguida, atualiza os pacotes no nó gerenciado com base no conteúdo dessa versão vinculada.

No AL2023, o repositório pré-configurado é o seguinte:

- ID do repositório: `amazonlinux`

  Nome do repositório: repositório do Amazon Linux 2023

No Amazon Linux 2022 (versão de pré-visualização), os repositórios pré-configurados estão vinculados a versões vinculadas de atualizações de pacotes. Quando novas Amazon Machine Images (AMIs) para o Amazon Linux 2022 são lançadas, elas são vinculadas a uma versão específica. Para atualizações de patches, o Patch Manager recupera a versão vinculada mais recente do repositório de atualizações de patches e, em seguida, atualiza os pacotes no nó gerenciado com base no conteúdo dessa versão vinculada.

No Amazon Linux 2022, o repositório pré-configurado é o seguinte:

- ID do repositório: `amazonlinux`

  Nome do repositório: repositório do Amazon Linux 2022

> **Note**
>
> Todas as atualizações são obtidas por download dos repositórios remotos configurados em seu nó gerenciado. Portanto, o nó deve ter acesso de saída à Internet para se conectar aos repositórios para que a aplicação do patch seja realizada.

Os nós gerenciados do Amazon Linux 1 e do Amazon Linux 2 usam o Yum como gerenciador de pacotes. Os nós gerenciados do Amazon Linux 2022 e do Amazon Linux 2023 usam o DNF como gerenciador de pacotes.

Os gerenciadores de pacotes usam o conceito de um aviso de atualização como um arquivo denominado `updateinfo.xml`. Uma notificação de atualização é um conjunto de pacotes que corrige problemas específicos. Todos os pacotes que estão em um aviso de atualização são considerados de segurança pelo Patch Manager. Pacotes individuais não recebem classificações ou níveis de gravidade. Por esse motivo, o Patch Manager designa os atributos de um aviso de atualização aos pacotes relacionados.

> **Note**
>
> Se você marcar a caixa de seleção Incluir atualizações não relacionadas a segurança na página Criar lista de referência de patch, os pacotes não classificados em um arquivo `updateinfo.xml` (ou um pacote que contenha um arquivo sem valores de classificação, gravidade e data devidamente formatados) poderão ser incluídos na lista de patches pré-filtrada. No entanto, para que um patch seja aplicado, ele ainda deve atender às regras de linha de base de patch especificadas pelo usuário.
> Para obter mais informações sobre a opção Incluir atualizações não relacionadas à segurança, consulte Como os patches são instalados e Como funcionam as regras de linha de base de patch em sistemas baseados no Linux.

CentOS and CentOS Stream

No CentOS e CentOS Stream, o serviço de lista de referência de patches do Systems Manager usa repositórios (repos) pré-configurados em seu nó gerenciado. A lista a seguir fornece exemplos para uma Amazon Machine Image (AMI) do CentOS 8.2 fictícia:

- ID do repositório: `example-centos-8.2-base`

  Nome do repositório: `Example CentOS-8.2 - Base`
- ID do repositório: `example-centos-8.2-extras`

  Nome do repositório: `Example CentOS-8.2 - Extras`
- ID do repositório: `example-centos-8.2-updates`

  Nome do repositório: `Example CentOS-8.2 - Updates`
- ID do repositório: `example-centos-8.x-examplerepo`

  Nome do repositório: `Example CentOS-8.x – Example Repo Packages`

> **Note**
>
> Todas as atualizações são obtidas por download dos repositórios remotos configurados em seu nó gerenciado. Portanto, o nó deve ter acesso de saída à Internet para se conectar aos repositórios para que a aplicação do patch seja realizada.

Os nós gerenciados do CentOS 6 e 7 usam o Yum como gerenciador de pacotes. Os nós do CentOS 8 e CentOS Stream usam o DNF como gerenciador de pacotes. Ambos os gerenciadores de pacotes usam o conceito de um aviso de atualização. Uma notificação de atualização é um conjunto de pacotes que corrige problemas específicos.

No entanto, os repositórios padrão do CentOS e CentOS Stream não são configurados com um aviso de atualização. Isso significa que o Patch Manager não detecta pacotes em repositórios padrão do CentOS e CentOS Stream. Para permitir que o Patch Manager processe pacotes que não estão contidos em um aviso de atualização, você deve ativar o sinalizador `EnableNonSecurity` nas regras da lista de referência de patches.

> **Note**
>
> Há compatibilidade para as notificações de atualização do CentOS e CentOS Stream. Os repositórios com notificações de atualização podem ser obtidos por download após a inicialização.

Debian Server and Raspberry Pi OS

No Debian Server e no Raspberry Pi OS (anteriormente chamado de Raspbian), o serviço de lista de referência de patches do Systems Manager usa repositórios (repos) pré-configurados na instância. Esses repos pré-configurados são usados para obter uma lista das atualizações de pacote disponíveis. Para isso, o Systems Manager executa um comando equivalente a `sudo apt-get update`.

Os pacotes são então filtrados dos repositórios `debian-security` *`codename`*. Isso significa que, em cada versão do Debian Server, o Patch Manager identifica apenas as atualizações que fazem parte do repositório associado a essa versão, da seguinte forma:

- Debian Server 8: `debian-security jessie`
- Debian Server 9: `debian-security stretch`
- Debian Server 10: `debian-security buster`
- Debian Server 11: `debian-security bullseye`
- Debian Server 12: `debian-security bookworm`

> **Note**
>
> Somente no Debian Server 8: como alguns nós gerenciados do Debian Server 8.* se referem a um repositório de pacotes obsoleto (`jessie-backports`), o Patch Manager executa etapas adicionais para garantir que as operações de patch sejam bem-sucedidas. Para obter mais informações, consulte Como os patches são instalados.

Oracle Linux

No Oracle Linux, o serviço de lista de referência de patches do Systems Manager usa repositórios (repos) pré-configurados em seu nó gerenciado. Há normalmente dois repositórios pré-configurados (repositórios) em um nó.

Oracle Linux 7:

- ID do repositório: `ol7_UEKR5/x86_64`

  Nome do repositório: `Latest Unbreakable Enterprise Kernel Release 5 for Oracle Linux 7Server (x86_64)`
- ID do repositório: `ol7_latest/x86_64`

  Nome do repositório: `Oracle Linux 7Server Latest (x86_64)`

Oracle Linux 8:

- ID do repositório: `ol8_baseos_latest`

  Nome do repositório: `Oracle Linux 8 BaseOS Latest (x86_64)`
- ID do repositório: `ol8_appstream`

  Nome do repositório: `Oracle Linux 8 Application Stream (x86_64)`
- ID do repositório: `ol8_UEKR6`

  Nome do repositório: `Latest Unbreakable Enterprise Kernel Release 6 for Oracle Linux 8 (x86_64)`

Oracle Linux 9:

- ID do repositório: `ol9_baseos_latest`

  Nome do repositório: `Oracle Linux 9 BaseOS Latest (x86_64)`

- ID do repositório: `ol9_appstream`

  Nome do repositório: `Oracle Linux 9 Application Stream Packages(x86_64)`

- ID do repositório: `ol9_UEKR7`

  Nome do repositório: `Oracle Linux UEK Release 7 (x86_64)`

> **Note**
>
> Todas as atualizações são obtidas por download dos repositórios remotos configurados em seu nó gerenciado. Portanto, o nó deve ter acesso de saída à Internet para se conectar aos repositórios para que a aplicação do patch seja realizada.

Os nós gerenciados do Oracle Linux usam o Yum como gerenciador de pacotes, e o Yum usa o conceito de uma notificação de atualização como um arquivo chamado `updateinfo.xml`. Uma notificação de atualização é um conjunto de pacotes que corrige problemas específicos. Pacotes individuais não recebem classificações ou níveis de gravidade. Por essa razão, o Patch Manager designa os atributos de um aviso de atualização aos pacotes relacionados e instala pacotes com base nos filtros de classificação especificados na lista de referência do patch.

> **Note**
>
> Se você marcar a caixa de seleção Incluir atualizações não relacionadas a segurança na página Criar lista de referência de patch, os pacotes não classificados em um arquivo `updateinfo.xml` (ou um pacote que contenha um arquivo sem valores de classificação, gravidade e data devidamente formatados) poderão ser incluídos na lista de patches pré-filtrada. No entanto, para que um patch seja aplicado, ele ainda deve atender às regras de linha de base de patch especificadas pelo usuário.

AlmaLinux, RHEL, and Rocky Linux

No AlmaLinux, no Red Hat Enterprise Linux e no Rocky Linux, o serviço de lista de referência de patches do Systems Manager usa repositórios (repos) pré-configurados em seu nó gerenciado. Há normalmente três repositórios pré-configurados (repos) em um nó.

Todas as atualizações são obtidas por download dos repositórios remotos configurados em seu nó gerenciado. Portanto, o nó deve ter acesso de saída à Internet para se conectar aos repositórios para que a aplicação do patch seja realizada.

> **Note**
>
> Se você marcar a caixa de seleção Incluir atualizações não relacionadas a segurança na página Criar lista de referência de patch, os pacotes não classificados em um arquivo `updateinfo.xml` (ou um pacote que contenha um arquivo sem valores de classificação, gravidade e data devidamente formatados) poderão ser incluídos na lista de patches pré-filtrada. No entanto, para que um patch seja aplicado, ele ainda deve atender às regras de linha de base de patch especificadas pelo usuário.

Os nós gerenciados do Red Hat Enterprise Linux 7 usam o Yum como gerenciador de pacotes. O AlmaLinux, o Red Hat Enterprise Linux 8 e os nós gerenciados do Rocky Linux usam o DNF como o gerenciador de pacotes. Os gerenciadores de pacotes usam o conceito de um aviso de atualização como um arquivo chamado `updateinfo.xml`. Uma notificação de atualização é um conjunto de pacotes que corrige problemas específicos. Pacotes individuais não recebem classificações ou níveis de gravidade. Por essa razão, o Patch Manager designa os atributos de um aviso de atualização aos pacotes relacionados e instala pacotes com base nos filtros de classificação especificados na lista de referência do patch.

RHEL 7

> **Note**
>
> Os IDs de repo a seguir estão associados ao RHUI 2. O RHUI 3 foi lançado em dezembro de 2019 e apresentou um esquema de nomenclatura diferente para IDs do repositório Yum. Dependendo da AMI do RHEL-7 na qual você cria os nós gerenciados, talvez seja necessário atualizar os comandos. Para obter mais

informações, consulte Repository IDs for RHEL 7 in AWS Have Changed no Red Hat Customer Portal.

- ID do repositório: `rhui-REGION-client-config-server-7/x86_64`

  Nome do repositório: `Red Hat Update Infrastructure 2.0 Client Configuration Server 7`
- ID do repositório: `rhui-REGION-rhel-server-releases/7Server/x86_64`

  Nome do repositório: `Red Hat Enterprise Linux Server 7 (RPMs)`
- ID do repositório: `rhui-REGION-rhel-server-rh-common/7Server/x86_64`

  Nome do repositório: `Red Hat Enterprise Linux Server 7 RH Common (RPMs)`

AlmaLinux, 8 RHEL 8 e Rocky Linux 8

- ID do repositório: `rhel-8-appstream-rhui-rpms`

  Nome do repositório: `Red Hat Enterprise Linux 8 for x86_64 - AppStream from RHUI (RPMs)`
- ID do repositório: `rhel-8-baseos-rhui-rpms`

  Nome do repositório: `Red Hat Enterprise Linux 8 for x86_64 - BaseOS from RHUI (RPMs)`
- ID do repositório: `rhui-client-config-server-8`

  Nome do repositório: `Red Hat Update Infrastructure 3 Client Configuration Server 8`

AlmaLinux 9, RHEL 9 e Rocky Linux 9

- ID do repositório: `rhel-9-appstream-rhui-rpms`

  Nome do repositório: `Red Hat Enterprise Linux 9 for x86_64 - AppStream from RHUI (RPMs)`
- ID do repositório: `rhel-9-baseos-rhui-rpms`

  Nome do repositório: `Red Hat Enterprise Linux 9 for x86_64 - BaseOS from RHUI (RPMs)`
- ID do repositório: `rhui-client-config-server-9`

Nome do repositório: `Red Hat Enterprise Linux 9 Client Configuration`

SLES

Nas instâncias do SUSE Linux Enterprise Server (SLES), a biblioteca do ZYPP obtém a lista de patches disponíveis (um conjunto de pacotes) dos seguintes locais:

- Lista de repositórios: `etc/zypp/repos.d/*`
- Informações do pacote: `/var/cache/zypp/raw/*`

Os nós gerenciados do SLES usam o Zypper como gerenciador de pacotes, e o Zypper usa o conceito de um patch. Um patch é um conjunto de pacotes que corrige um problema específico. O Patch Manager lida com todos os pacotes referenciados em um patch como relacionados à segurança. Como pacotes individuais não recebem classificações ou níveis de gravidade, o Patch Manager atribui aos pacotes os atributos do patch ao qual eles pertencem.

Ubuntu Server

No Ubuntu Server, o serviço de lista de referência de patches do Systems Manager usa repositórios (repos) pré-configurados em seu nó gerenciado. Esses repos pré-configurados são usados para obter uma lista das atualizações de pacote disponíveis. Para isso, o Systems Manager executa um comando equivalente a `sudo apt-get update`.

Os pacotes são então filtrados dos repositórios *`codename`*`-security`, onde codename é exclusivo da versão do lançamento, como `trusty` para Ubuntu Server 14. O Patch Manager identifica apenas upgrades que são parte desses repositórios:

- Ubuntu Server 14.04 LTS: `trusty-security`
- Ubuntu Server 16.04 LTS: `xenial-security`
- Ubuntu Server 18.04 LTS: `bionic-security`
- Ubuntu Server 20.04 LTS: `focal-security`
- Ubuntu Server 20.10 STR: `groovy-security`
- Ubuntu Server 22.04 LTS (`jammy-security`)
- Ubuntu Server 23.04 (`lunar-security`)
- Ubuntu Server 23.10 (`mantic-security`)
- Ubuntu Server 24.04 LTS (`noble-security`)

- Ubuntu Server 24.10 (`oracular-security`)
- Ubuntu Server 25.04 (`plucky-security`)

Windows Server

Em sistemas operacionais Microsoft Windows, o Patch Manager recupera uma lista de atualizações disponíveis que a Microsoft publica no Microsoft Update e são disponibilizadas automaticamente para o Windows Server Update Services (WSUS).

> **Note**
>
> O Patch Manager apenas disponibiliza patches para versões de sistemas operacionais Windows Server compatíveis com o Patch Manager. Por exemplo, o Patch Manager não pode ser usado para aplicar patches ao Windows RT.

O Patch Manager monitora continuamente as novas atualizações em cada Região da AWS. A lista de atualizações do disponíveis em cada região é atualizada pelo menos uma vez por dia. Quando as informações de patch da Microsoft são processadas, o Patch Manager remove da sua lista de patches as atualizações que foram substituídas por atualizações posteriores. Portanto, apenas a atualização mais recente é exibida e disponibilizada para instalação. Por exemplo, se `KB4012214` substituir `KB3135456`, somente o `KB4012214` será disponibilizado como atualização no Patch Manager.

Da mesma forma, o Patch Manager pode instalar somente os patches que estão disponíveis no nó gerenciado durante o período da operação de patch. Por padrão, o Windows Server 2019 e o Windows Server 2022 removem atualizações que são substituídas por atualizações posteriores. Como resultado, se você usar o parâmetro `ApproveUntilDate` em uma lista de referência de patches Windows Server, mas a data selecionada no parâmetro `ApproveUntilDate` for anterior à data do patch mais recente, ocorrerá o seguinte cenário:

- O patch substituído é removido do nó e, portanto, não pode ser instalado usando o Patch Manager.
- O patch de substituição mais recente está presente no nó, mas ainda não foi aprovado para instalação na data especificada pelo parâmetro `ApproveUntilDate`.

Isso significa que o nó gerenciado é compatível em termos de operações do Systems Manager, mesmo que um patch crítico do mês anterior possa não ter sido instalado. Esse mesmo cenário pode ocorrer ao usar o parâmetro `ApproveAfterDays`. Devido ao comportamento do patch substituído pela Microsoft, é possível definir um número (geralmente maior que 30 dias) para que os patches do Windows Server nunca sejam instalados se o patch mais recente disponível da Microsoft for lançado antes de o número de dias em `ApproveAfterDays` tiver passado. Observe que esse comportamento do sistema não se aplica se você tiver modificado as configurações do Objeto de Política de Grupo (GPO) do Windows para disponibilizar o patch substituído nos nós gerenciados.

> **Note**
>
> Em alguns casos, a Microsoft lança patches para aplicações que não especificam data e hora atualizadas. Nesses casos, uma data e hora atualizadas de `01/01/1970` são fornecidas por padrão.

Como especificar um repositório de origem de patches alternativo (Linux)

Ao usar os repositórios padrão configurados em um nó gerenciado para operações de aplicação de patch, o Patch Manager, uma ferramenta do AWS Systems Manager, verifica ou instala patches relacionados à segurança. Esse é o comportamento padrão do Patch Manager. Para obter informações completas sobre como o Patch Manager seleciona e instala patches de segurança, consulte Como os patches de segurança são selecionados.

Em sistemas Linux, no entanto, você também pode usar o Patch Manager para instalar patches que não são relacionados à segurança ou que estão em um repositório de origem diferente do padrão configurado em seu nó gerenciado. Você pode especificar os repositórios de origem de patches alternativos ao criar uma linha de base de patch personalizada. Em cada linha de base de patch personalizada, você pode especificar as configurações de origem de patches para até 20 versões de um sistema operacional Linux compatível.

Por exemplo, suponha que a sua frota Ubuntu Server inclua nós gerenciados do Ubuntu Server 14.04 e Ubuntu Server 16.04. Nesse caso, você pode especificar os repositórios alternativo para cada versão na mesma linha de base de patch personalizada. Para cada versão, insira um nome, especifique o tipo de versão do sistema operacional (produto) e forneça uma configuração do repositório. Você também pode especificar um único repositório de origem alternativo que se aplica a todas as versões de um sistema operacional compatível.

> **Note**
>
> Executar uma linha personalizada de referência de patches, que especifica repositórios de patches alternativos para um nó gerenciado, não torna esses repositórios o novo padrão no sistema operacional. Após a conclusão da operação de patch, os repositórios previamente configurados como padrão para o sistema operacional do nó permanecerão como padrão.

Para obter uma lista de exemplos de situações para usar essa opção, consulte Exemplos de uso de repositórios de origem de patches alternativos mais adiante neste tópico.

Para obter informações sobre linhas de base de patch padrão e personalizadas, consulte Listas de referência de patches predefinidas e personalizadas.

Exemplo: usar o console

Para especificar repositórios de origem de patches alternativos quando você está trabalhando no console do Systems Manager, use a seção Patch sources (Origens de patches) na página Create patch baseline (Criar linha de base de patch). Para obter informações sobre como usar as opções de Fontes de patch, consulte Criar uma lista de referência de patches personalizada para o Linux.

Exemplos: usar a AWS CLI

Para obter um exemplo do uso da opção `--sources` com a AWS Command Line Interface (AWS CLI), consulte Criar uma linha de base de patch com repositórios personalizados para diferentes versões do SO.

Tópicos

- Considerações importantes para repositórios alternativos
- Exemplos de uso de repositórios de origem de patches alternativos

Considerações importantes para repositórios alternativos

Tenha em mente os seguintes pontos ao planejar a estratégia de patches usando alternativas de repositórios de patch.

Somente repositórios especificados são usados para a aplicação de patches

Especificar repositórios alternativos não significa especificar repositórios adicionais. Você pode optar por especificar repositórios que não sejam os configurados como padrão em um nó gerenciado. No

entanto, você também deve especificar os repositórios padrão como parte da configuração de origem de patch alternativo se você deseja que as atualizações sejam aplicadas.

Por exemplo, em nós gerenciados do Amazon Linux 2, os repositórios padrão são `amzn2-core` e `amzn2extra-docker`. Se você deseja incluir o repositório Extra Packages for Enterprise Linux (EPEL) em suas operações de patch, deve especificar os três repositórios como repositórios alternativos.

> **Note**
>
> Executar uma linha personalizada de referência de patches, que especifica repositórios de patches alternativos para um nó gerenciado, não torna esses repositórios o novo padrão no sistema operacional. Após a conclusão da operação de patch, os repositórios previamente configurados como padrão para o sistema operacional do nó permanecerão como padrão.

O comportamento de patches para distribuições com base em YUM depende do manifesto updateinfo.xml

Quando você especifica repositórios alternativos de patch para distribuições com base em YUM, como Amazon Linux 1 ou Amazon Linux 2, Red Hat Enterprise Linux, ou CentOS, o comportamento de patches depende de o repositório incluir ou não uma atualização manifesto na forma de um arquivo `updateinfo.xml` completo e formatado corretamente. Esse arquivo especifica a data de lançamento, classificações e gravidades dos vários pacotes. Qualquer um dos seguintes afetará o comportamento de patch:

- Se você filtrar Classificação e Gravidade, mas eles não estiverem especificados em `updateinfo.xml`, o pacote não será incluído pelo filtro. Isso também significa que os pacotes sem um arquivo `updateinfo.xml` não serão incluídos no patch.
- Se você filtrar ApprovalAfterDays, mas a data de liberação do pacote não estiver no formato Unix Epoch (ou não tiver data de liberação especificada), o pacote não será incluído pelo filtro.
- Há uma exceção se você selecionar a caixa de seleção Incluir atualizações não relacionadas a segurança na página Criar linha de base de patch. Nesse caso, os pacotes sem um arquivo `updateinfo.xml` (ou que contenham esse arquivo sem valores propriamente formatados de Classificação, Gravidade e Data) serão incluídos na lista de patches pré-filtrados. (Eles ainda deverão corresponder aos outros requisitos de regras de linha de base de patch para serem instalados.)

Exemplos de uso de repositórios de origem de patches alternativos

Exemplo 1 - Atualizações não de segurança para o Ubuntu Server

Você já está usando o Patch Manager para instalar patches de segurança em uma frota de nós gerenciados do Ubuntu Server usando a lista de referência de patches `AWS-UbuntuDefaultPatchBaseline` predefinida fornecida pela AWS. Você pode criar uma nova linha de base de patch baseada nesse padrão, mas especifique nas regras de aprovação que você deseja que as atualizações não relacionadas à segurança que fazem parte da distribuição padrão sejam instaladas também. Quando essa lista de referência de patches é executada em nós, os patches para problemas relacionados ou não a segurança são aplicados. Você também pode optar por aprovar patches não relacionados a segurança nas exceções de patches que você especifica para uma linha de base.

Exemplo 2 – Personal Package Archives (PPA) para o Ubuntu Server

As instâncias do Ubuntu Server estão executando o software que é distribuído por meio de um Personal Package Archives (PPA) para Ubuntu. Nesse caso, crie uma lista de referência de patches que especifique um repositório PPA que você configurou em seu nó gerenciado como repositório de origem para a operação de aplicação de patch. Em seguida, use o Run Command para executar o documento da lista de referência de patches em seus nós.

Exemplo 3: aplicações corporativas internas no Amazon Linux

Você deve executar algumas aplicações necessárias para a conformidade regulatória do setor nos nós gerenciados do Amazon Linux. Você pode configurar um repositório para essas aplicações em nós, usar o YUM inicialmente para instalar as aplicações e, em seguida, atualizar ou criar uma nova lista de referência de patches para incluir esse novo repositório corporativo. Depois disso, você pode usar o Run Command para executar o documento `AWS-RunPatchBaseline` com a opção `Scan` para conferir se o pacote corporativo está listado entre os pacotes instalados e está atualizado dentro do nó gerenciado. Se ele não estiver atualizado, você pode executar o documento novamente usando a opção `Install` para atualizar os aplicativos.

Como os patches são instalados

O Patch Manager, uma ferramenta do AWS Systems Manager, usa o mecanismo interno para um tipo de sistema operacional para instalar atualizações em um nó gerenciado. Por exemplo, a API do Windows Update é usada no Windows Server e o gerenciador de pacotes `yum` é usado no Amazon Linux 2.

O Patch Manager não instala um novo pacote que substitui um pacote obsoleto atualmente instalado. (Exceções: o novo pacote depende da instalação de outro pacote que está sendo atualizado, ou o novo pacote possui o mesmo nome do pacote obsoleto.) Em vez disso, o Patch Manager informa e instala as atualizações disponíveis para os pacotes que estão instalados. Essa abordagem ajuda a impedir alterações inesperadas na funcionalidade do sistema que podem ocorrer ao substituir um pacote por outro.

Se for necessário desinstalar um pacote que se tornou obsoleto e instalar um substituto, pode ser necessário que você utilize um script personalizado ou comandos do gerenciador de pacotes fora das operações padrão do Patch Manager.

O lembrete desta seção explica como o Patch Manager instala patches em um sistema operacional.

Amazon Linux 1, Amazon Linux 2, Amazon Linux 2022, and Amazon Linux 2023

Nos nós gerenciados do Amazon Linux 1, do Amazon Linux 2, do Amazon Linux 2022 e do Amazon Linux 2023, o fluxo de trabalho de instalação de patches ocorre da seguinte maneira:

1. Se uma lista de patches for especificada usando um URL https ou um URL de estilo caminho do Amazon Simple Storage Service (Amazon S3), que usa o parâmetro do `InstallOverrideList` para os documentos `AWS-RunPatchBaseline` ou `AWS-RunPatchBaselineAssociation`, os patches listados serão instalados e as de etapas de 2 a 7 serão ignoradas.
2. Aplique o GlobalFilters conforme especificado na linha de base de patch, mantendo apenas os pacotes qualificados para processamento adicional.
3. Aplique o ApprovalRules conforme especificado na lista de referência de patches. Cada regra de aprovação pode definir um pacote como aprovado.

   As regras de aprovação, no entanto, também dependem da seleção da opção Include nonsecurity updates (Incluir atualizações não relacionadas à segurança) ao criar ou atualizar pela última vez a lista de referência de patches.

   Se nenhuma atualização que não seja de segurança for excluída, uma regra implícita será aplicada para selecionar apenas os pacotes com atualizações nos repositórios de segurança. Para cada pacote, a versão candidata do pacote (que geralmente é a versão mais recente) deve fazer parte de um repo de segurança.

   Se atualizações não relacionadas à segurança forem incluídas, os patches de outros repositórios também serão considerados.

4. Aplique o ApprovedPatches conforme especificado na lista de referência de patches. Os patches aprovados têm aprovação para atualizar, mesmo que sejam descartados por GlobalFilters ou mesmo que nenhuma regra de aprovação especificada em ApprovalRules conceda a aprovação.
5. Aplique o RejectedPatches conforme especificado na lista de referência de patches. Os patches rejeitados são removidos da lista de patches aprovados e não serão aplicados.
6. Se várias versões de um patch forem aprovadas, a versão mais recente será aplicada.
7. A API de atualização do YUM (Amazon Linux 1, Amazon Linux 2) ou a API de atualização do DNF (Amazon Linux 2022, API (Amazon Linux 2023) é aplicada aos patches aprovados da seguinte maneira:
   - Para linhas de base de patch padrão predefinidas fornecidas pela AWS, somente os patches especificados em `updateinfo.xml` são aplicados (apenas atualizações de segurança). Isso acontece porque a caixa de seleção Incluir atualizações não relacionadas a segurança não está marcada. As linhas de base predefinidas são equivalentes a uma linha de base personalizada com o seguinte:
     - A caixa de seleção Incluir atualizações não relacionadas a segurança não está marcada
     - Uma lista de GRAVIDADE de `[Critical, Important]`
     - Uma lista de CLASSIFICAÇÃO de `[Security, Bugfix]`

     No Amazon Linux 1 e no Amazon Linux 2, o comando yum equivalente para esse fluxo de trabalho é:

     ```
     sudo yum update-minimal --sec-severity=Critical,Important --bugfix -y
     ```

     No Amazon Linux 2022 e no Amazon Linux 2023, o comando dnf equivalente para esse fluxo de trabalho é:

     ```
     sudo dnf upgrade-minimal --sec-severity=Critical --sec-severity=Important --
     bugfix -y
     ```

     Se a caixa de seleção Incluir atualizações não relacionadas a segurança estiver marcada, os patches em `updateinfo.xml` e fora de `updateinfo.xml` serão todos aplicados (atualizações de segurança e não relacionadas a segurança).

     Para o Amazon Linux 1 e o Amazon Linux 2, se uma linha de base com Incluir atualizações não relacionadas a segurança for selecionada, tiver uma lista de GRAVIDADE de

`[Critical, Important]` e uma lista de CLASSIFICAÇÃO de `[Security, Bugfix]`, o comando yum equivalente será:

```
sudo yum update --security --sec-severity=Critical,Important --bugfix -y
```

No Amazon Linux 2022 e no Amazon Linux 2023, o comando dnf equivalente é:

```
sudo dnf upgrade --security --sec-severity=Critical --sec-severity=Important --
bugfix -y
```

> **Note**
>
> Novos pacotes que substituem pacotes que se tornaram obsoletos com nomes diferentes serão instalados se os seguintes comandos `yum` ou `dnf` forem executados fora do Patch Manager. Porém, eles não são instalados por meio de operações do Patch Manager equivalentes.

Detalhes adicionais de patches para Amazon Linux 2022 e Amazon Linux 2023

Suporte ao nível de gravidade "Nenhuma"

O Amazon Linux 2022 e o Amazon Linux 2023 também são compatíveis com o nível de gravidade de patch `None`, que é reconhecido pelo gerenciador de pacotes DNF.

Suporte ao nível de gravidade "Média"

No Amazon Linux 2022 e no Amazon Linux 2023, um nível de gravidade de patch `Medium` é equivalente a uma gravidade `Moderate` que pode ser definida em alguns repositórios externos. Se você incluir patches de gravidade `Medium` na linha lista de referência de patches, os patches de gravidade `Moderate` dos patches externos também serão instalados nas instâncias.

Quando você consulta dados de conformidade usando a ação de API DescribeInstancePatches, a filtragem para o nível de gravidade `Medium` informa patches com ambos os níveis de gravidade `Medium` e `Moderate`.

Tratamento de dependências transitivas para Amazon Linux 2022 e Amazon Linux 2023

No Amazon Linux 2022 e no Amazon Linux 2023, o Patch Manager pode instalar versões diferentes de dependências transitivas do que os comandos `dnf` equivalentes instalam. Dependências transitivas são pacotes que são instalados automaticamente para satisfazer os requisitos de outros pacotes (dependências de dependências).

Por exemplo, o `dnf upgrade-minimal --security` instala as versões mínimas das dependências transitivas necessárias para resolver problemas de segurança conhecidos, enquanto o Patch Manager instala as versões mais recentes disponíveis das mesmas dependências transitivas.

8. O nó gerenciado será reinicializado se todas as atualizações forem instaladas. (Exceção: Se o parâmetro `RebootOption` estiver definido como `NoReboot` no documento `AWS-RunPatchBaseline`, o nó gerenciado não será reinicializado depois que o Patch Manager for executado. Para obter mais informações, consulte Nome do parâmetro: `RebootOption`).

CentOS and CentOS Stream

Em nós gerenciados do CentOS e CentOS Stream, o fluxo de trabalho da instalação de patches é o seguinte:

1. Se uma lista de patches for especificada usando um URL https ou um URL de estilo caminho do Amazon Simple Storage Service (Amazon S3), que usa o parâmetro do `InstallOverrideList` para os documentos `AWS-RunPatchBaseline` ou `AWS-RunPatchBaselineAssociation`, os patches listados serão instalados e as de etapas de 2 a 7 serão ignoradas.

   Aplique o GlobalFilters conforme especificado na linha de base de patch, mantendo apenas os pacotes qualificados para processamento adicional.

2. Aplique o ApprovalRules conforme especificado na lista de referência de patches. Cada regra de aprovação pode definir um pacote como aprovado.

   As regras de aprovação, no entanto, também dependem da seleção da opção Include nonsecurity updates (Incluir atualizações não relacionadas à segurança) ao criar ou atualizar pela última vez a lista de referência de patches.

   Se nenhuma atualização que não seja de segurança for excluída, uma regra implícita será aplicada para selecionar apenas os pacotes com atualizações nos repositórios de segurança.

Para cada pacote, a versão candidata do pacote (que geralmente é a versão mais recente) deve fazer parte de um repo de segurança.

Se atualizações não relacionadas à segurança forem incluídas, os patches de outros repositórios também serão considerados.

3. Aplique o ApprovedPatches conforme especificado na lista de referência de patches. Os patches aprovados têm aprovação para atualizar, mesmo que sejam descartados por GlobalFilters ou mesmo que nenhuma regra de aprovação especificada em ApprovalRules conceda a aprovação.
4. Aplique o RejectedPatches conforme especificado na lista de referência de patches. Os patches rejeitados são removidos da lista de patches aprovados e não serão aplicados.
5. Se várias versões de um patch forem aprovadas, a versão mais recente será aplicada.
6. A API de atualização do YUM (nas versões 6.x e 7.x) ou a atualização do DNF (no CentOS 8 e CentOS Stream) é aplicada a patches aprovados.

> **Note**
>
> No CentOS, o Patch Manager pode instalar versões de dependências transitivas das instaladas pelos comandos `dnf` equivalentes. Dependências transitivas são pacotes que são instalados automaticamente para satisfazer os requisitos de outros pacotes (dependências de dependências).
> Por exemplo, o `dnf upgrade-minimal --security` instala as versões mínimas das dependências transitivas necessárias para resolver problemas de segurança conhecidos, enquanto o Patch Manager instala as versões mais recentes disponíveis das mesmas dependências transitivas.

7. O nó gerenciado será reinicializado se todas as atualizações forem instaladas. (Exceção: Se o parâmetro `RebootOption` estiver definido como `NoReboot` no documento `AWS-RunPatchBaseline`, o nó gerenciado não será reinicializado depois que o Patch Manager for executado. Para obter mais informações, consulte Nome do parâmetro: `RebootOption`).

Debian Server and Raspberry Pi OS

Em instâncias do Debian Server e do Raspberry Pi OS (anteriormente Raspbian), o fluxo de trabalho da instalação de patches é o seguinte:

1. Se uma lista de patches for especificada usando um URL https ou um URL de estilo caminho do Amazon Simple Storage Service (Amazon S3), que usa o parâmetro do `InstallOverrideList` para os documentos `AWS-RunPatchBaseline` ou `AWS-RunPatchBaselineAssociation`, os patches listados serão instalados e as de etapas de 2 a 7 serão ignoradas.
2. Se uma atualização estiver disponível para o `python3-apt` (uma interface de biblioteca Python para `libapt`), ela será atualizada para a versão mais recente. (Este pacote não relacionado à segurança é atualizado mesmo se você não selecionou a opção Include nonsecurity updates (Incluir atualizações não relacionadas à segurança)).

   > **Important**
   >
   > Somente no Debian Server 8: como alguns nós gerenciados do Debian Server 8.* se referem a um repositório de pacotes obsoleto (`jessie-backports`), o Patch Manager executa etapas adicionais para garantir que as operações de patch tenham êxito:
   >
   > a. Em seu nó gerenciado, a referência ao repositório `jessie-backports` é comentada na lista de locais de origem (`/etc/apt/sources.list.d/jessie-backports`). Como resultado, nenhuma tentativa é feita para baixar patches desse local.
   >
   > b. Uma chave de assinatura da atualização de segurança do Stretch é importada. Essa chave fornece as permissões necessárias para as operações de atualização e instalação nas distribuições do Debian Server 8.*.
   >
   > c. Uma operação `apt-get` é executada para garantir que a versão mais recente de `python3-apt` seja instalada antes do início do processo de aplicação de patch.
   >
   > d. Após a conclusão do processo de instalação, a referência ao repositório `jessie-backports` é restaurada e a chave de assinatura é removida das chaves de origem apt. Isso é feito para deixar a configuração do sistema como estava antes da operação de aplicação de patch.
   >
   > Na próxima vez que o Patch Manager atualizar o sistema, o mesmo processo será repetido.

3. Aplique o GlobalFilters conforme especificado na linha de base de patch, mantendo apenas os pacotes qualificados para processamento adicional.
4. Aplique o ApprovalRules conforme especificado na lista de referência de patches. Cada regra de aprovação pode definir um pacote como aprovado.

> **Note**
>
> Como não é possível determinar de forma confiável as datas de lançamento dos pacotes de atualização do Debian Server, as opções de aprovação automática não são compatíveis com esse sistema operacional.

As regras de aprovação, no entanto, também dependem da seleção da opção Include nonsecurity updates (Incluir atualizações não relacionadas à segurança) ao criar ou atualizar pela última vez a lista de referência de patches.

Se nenhuma atualização que não seja de segurança for excluída, uma regra implícita será aplicada para selecionar apenas os pacotes com atualizações nos repositórios de segurança. Para cada pacote, a versão candidata do pacote (que geralmente é a versão mais recente) deve fazer parte de um repo de segurança.

Se atualizações não relacionadas à segurança forem incluídas, os patches de outros repositórios também serão considerados.

> **Note**
>
> Para o Debian Server e o Raspberry Pi OS, as versões candidatas de patch são limitadas a patches incluídos no `debian-security`.

5. Aplique o ApprovedPatches conforme especificado na lista de referência de patches. Os patches aprovados têm aprovação para atualizar, mesmo que sejam descartados por GlobalFilters ou mesmo que nenhuma regra de aprovação especificada em ApprovalRules conceda a aprovação.
6. Aplique o RejectedPatches conforme especificado na lista de referência de patches. Os patches rejeitados são removidos da lista de patches aprovados e não serão aplicados.
7. A biblioteca de APT é usada para atualizar pacotes.

> **Note**
>
> O Patch Manager não oferece suporte ao uso da opção `Pin-Priority` de APT para atribuir prioridades aos pacotes. O Patch Manager agrega as atualizações disponíveis

de todos os repositórios habilitados e seleciona a atualização mais recente que corresponde à lista de referência de cada pacote instalado.

8. O nó gerenciado será reinicializado se todas as atualizações forem instaladas. (Exceção: Se o parâmetro `RebootOption` estiver definido como `NoReboot` no documento `AWS-RunPatchBaseline`, o nó gerenciado não será reinicializado depois que o Patch Manager for executado. Para obter mais informações, consulte Nome do parâmetro: `RebootOption`).

macOS

Em nós gerenciados do macOS, o fluxo de trabalho da instalação de patches é o seguinte:

1. O`/Library/Receipts/InstallHistory.plist`é um registro de software que foi instalado e atualizado usando o`softwareupdate`e`installer`Gerenciadores de pacotes. Usar o`pkgutil`ferramenta de linha de comando (para`installer`) e o`softwareupdate`, os comandos da CLI são executados para analisar esta lista.

   Para o `installer`, a resposta aos comandos da CLI inclui `package name`, `version`, `volume`, `location` e `install-time`, mas apenas o `package name` e a `version` serão usados pelo Patch Manager.

   Para o `softwareupdate`, a resposta aos comandos da CLI inclui o nome do pacote (`display name`), `version` e `date`, mas apenas o nome e a versão do pacote são usados pelo Patch Manager.

   Para Brew e Brew Cask, o Homebrew não suporta seus comandos rodando sob o usuário raiz. Como resultado, o Patch Manager consulta e executa comandos Homebrew como o proprietário do diretório Homebrew ou como um usuário válido pertencente ao grupo de proprietários do diretório Homebrew. Os comandos são semelhantes a`softwareupdate`e`installer`são executados por meio de um subprocesso Python para coletar dados de pacotes, e a saída é analisada para identificar nomes e versões de pacotes.
2. Aplique o GlobalFilters conforme especificado na linha de base de patch, mantendo apenas os pacotes qualificados para processamento adicional.
3. Aplique o ApprovalRules conforme especificado na lista de referência de patches. Cada regra de aprovação pode definir um pacote como aprovado.
4. Aplique o ApprovedPatches conforme especificado na lista de referência de patches. Os patches aprovados têm aprovação para atualizar, mesmo que sejam descartados por

GlobalFilters ou mesmo que nenhuma regra de aprovação especificada em ApprovalRules conceda a aprovação.

5. Aplique o RejectedPatches conforme especificado na lista de referência de patches. Os patches rejeitados são removidos da lista de patches aprovados e não serão aplicados.
6. Se várias versões de um patch forem aprovadas, a versão mais recente será aplicada.
7. Invoca a CLI de pacote apropriada em seu nó gerenciado para processar patches aprovados da seguinte maneira:

> **Note**
>
> O `installer` não tem a funcionalidade para verificar e instalar atualizações. Portanto, para o `installer`, o Patch Manager somente informa quais pacotes estão instalados. Como resultado, os pacotes do `installer` nunca são relatados como `Missing`.

   - Para listas de referência de patches padrão predefinidas fornecidas pela AWS e para linhas de base de patch personalizadas em que a caixa de seleção Incluir atualizações não relacionadas a segurança não está marcada, somente atualizações de segurança são aplicadas.
   - Para listas de referência de patches personalizadas em que a caixa de seleção Incluir atualizações não relacionadas a segurança estiver selecionada, tanto as atualizações de segurança quanto as não relacionadas a segurança se aplicarão.
8. O nó gerenciado será reinicializado se todas as atualizações forem instaladas. (Exceção: Se o parâmetro `RebootOption` estiver definido como `NoReboot` no documento `AWS-RunPatchBaseline`, o nó gerenciado não será reinicializado depois que o Patch Manager for executado. Para obter mais informações, consulte Nome do parâmetro: `RebootOption`).

Oracle Linux

Em nós gerenciados do Oracle Linux, o fluxo de trabalho da instalação de patches é o seguinte:

1. Se uma lista de patches for especificada usando um URL https ou um URL de estilo caminho do Amazon Simple Storage Service (Amazon S3), que usa o parâmetro do `InstallOverrideList` para os documentos `AWS-RunPatchBaseline` ou `AWS-RunPatchBaselineAssociation`, os patches listados serão instalados e as de etapas de 2 a 7 serão ignoradas.

2. Aplique o GlobalFilters conforme especificado na linha de base de patch, mantendo apenas os pacotes qualificados para processamento adicional.
3. Aplique o ApprovalRules conforme especificado na lista de referência de patches. Cada regra de aprovação pode definir um pacote como aprovado.

   As regras de aprovação, no entanto, também dependem da seleção da opção Include nonsecurity updates (Incluir atualizações não relacionadas à segurança) ao criar ou atualizar pela última vez a lista de referência de patches.

   Se nenhuma atualização que não seja de segurança for excluída, uma regra implícita será aplicada para selecionar apenas os pacotes com atualizações nos repositórios de segurança. Para cada pacote, a versão candidata do pacote (que geralmente é a versão mais recente) deve fazer parte de um repo de segurança.

   Se atualizações não relacionadas à segurança forem incluídas, os patches de outros repositórios também serão considerados.
4. Aplique o ApprovedPatches conforme especificado na lista de referência de patches. Os patches aprovados têm aprovação para atualizar, mesmo que sejam descartados por GlobalFilters ou mesmo que nenhuma regra de aprovação especificada em ApprovalRules conceda a aprovação.
5. Aplique o RejectedPatches conforme especificado na lista de referência de patches. Os patches rejeitados são removidos da lista de patches aprovados e não serão aplicados.
6. Se várias versões de um patch forem aprovadas, a versão mais recente será aplicada.
7. Em nós gerenciados da versão 7, a API da atualização do YUM é aplicada a patches aprovados, da seguinte maneira:
   - Para linhas de base de patch padrão predefinidas fornecidas pela AWS e para linhas de base de patch personalizadas em que a caixa de seleção Incluir atualizações não relacionadas a segurança não está marcada, somente os patches especificados em `updateinfo.xml` são aplicados (somente atualizações de segurança).

     O comando equivalente do yum para esse fluxo de trabalho é:

     ```
     sudo yum update-minimal --sec-severity=Important,Moderate --bugfix -y
     ```
   - Para linhas de base de patch personalizadas em que a opção Incluir atualizações não relacionadas a segurança está selecionada, ambos os patches no `updateinfo.xml` e

os que não estão no `updateinfo.xml` são aplicados (atualizações de segurança e não relacionadas a segurança).

O comando equivalente do yum para esse fluxo de trabalho é:

```
sudo yum update --security --bugfix -y
```

Em nós gerenciados da versão 8 e 9, a API da atualização do DNF é aplicada a patches aprovados, da seguinte maneira:

- Para linhas de base de patch padrão predefinidas fornecidas pela AWS e para linhas de base de patch personalizadas em que a caixa de seleção Incluir atualizações não relacionadas a segurança não está marcada, somente os patches especificados em `updateinfo.xml` são aplicados (somente atualizações de segurança).

  O comando equivalente do yum para esse fluxo de trabalho é:

  ```
  sudo dnf upgrade-minimal --security --sec-severity=Moderate --sec-
  severity=Important
  ```

  > **Note**
  >
  > No Oracle Linux, o Patch Manager pode instalar versões de dependências transitivas das instaladas pelos comandos `dnf` equivalentes. Dependências transitivas são pacotes que são instalados automaticamente para satisfazer os requisitos de outros pacotes (dependências de dependências).
  > Por exemplo, o `dnf upgrade-minimal --security` instala as versões mínimas das dependências transitivas necessárias para resolver problemas de segurança conhecidos, enquanto o Patch Manager instala as versões mais recentes disponíveis das mesmas dependências transitivas.

- Para linhas de base de patch personalizadas em que a opção Incluir atualizações não relacionadas a segurança está selecionada, ambos os patches no `updateinfo.xml` e os que não estão no `updateinfo.xml` são aplicados (atualizações de segurança e não relacionadas a segurança).

  O comando equivalente do yum para esse fluxo de trabalho é:

```
sudo dnf upgrade --security --bugfix
```

> **Note**
>
> Novos pacotes que substituem pacotes que se tornaram obsoletos com nomes diferentes serão instalados se os seguintes comandos `yum` ou `dnf` forem executados fora do Patch Manager. Porém, eles não são instalados por meio de operações do Patch Manager equivalentes.

8. O nó gerenciado será reinicializado se todas as atualizações forem instaladas. (Exceção: Se o parâmetro `RebootOption` estiver definido como `NoReboot` no documento `AWS-RunPatchBaseline`, o nó gerenciado não será reinicializado depois que o Patch Manager for executado. Para obter mais informações, consulte Nome do parâmetro: `RebootOption`).

AlmaLinux, RHEL, and Rocky Linux

Em nós gerenciados do AlmaLinux, do Red Hat Enterprise Linux e do Rocky Linux, o fluxo de trabalho da instalação de patches é este:

1. Se uma lista de patches for especificada usando um URL https ou um URL de estilo caminho do Amazon Simple Storage Service (Amazon S3), que usa o parâmetro do `InstallOverrideList` para os documentos `AWS-RunPatchBaseline` ou `AWS-RunPatchBaselineAssociation`, os patches listados serão instalados e as de etapas de 2 a 7 serão ignoradas.
2. Aplique o GlobalFilters conforme especificado na linha de base de patch, mantendo apenas os pacotes qualificados para processamento adicional.
3. Aplique o ApprovalRules conforme especificado na lista de referência de patches. Cada regra de aprovação pode definir um pacote como aprovado.

   As regras de aprovação, no entanto, também dependem da seleção da opção Include nonsecurity updates (Incluir atualizações não relacionadas à segurança) ao criar ou atualizar pela última vez a lista de referência de patches.

   Se nenhuma atualização que não seja de segurança for excluída, uma regra implícita será aplicada para selecionar apenas os pacotes com atualizações nos repositórios de segurança.

Para cada pacote, a versão candidata do pacote (que geralmente é a versão mais recente) deve fazer parte de um repo de segurança.

Se atualizações não relacionadas à segurança forem incluídas, os patches de outros repositórios também serão considerados.

4. Aplique o ApprovedPatches conforme especificado na lista de referência de patches. Os patches aprovados têm aprovação para atualizar, mesmo que sejam descartados por GlobalFilters ou mesmo que nenhuma regra de aprovação especificada em ApprovalRules conceda a aprovação.
5. Aplique o RejectedPatches conforme especificado na lista de referência de patches. Os patches rejeitados são removidos da lista de patches aprovados e não serão aplicados.
6. Se várias versões de um patch forem aprovadas, a versão mais recente será aplicada.
7. A API de atualização do YUM (no RHEL 7) ou a API de atualização do DNF (no AlmaLinux 8 e 9, no RHEL 8, 9 e 10 e no Rocky Linux 8 e 9) é aplicada a patches aprovados da seguinte forma:

   Cenário 1: atualizações não relacionadas à segurança excluídas

   - Aplica-se a: listas de referência de patches padrão predefinidas fornecidas pela AWS e listas de referência de patches personalizadas.
   - Caixa de seleção Incluir atualizações não relacionadas à segurança: não está marcada.
   - Patches aplicados: os patches especificados em `updateinfo.xml` (somente atualizações de segurança) serão aplicados apenas se ambos corresponderem à configuração da lista de referência de patches e forem encontrados nos repositórios configurados.

     Em alguns casos, um patch especificado em `updateinfo.xml` pode não estar mais disponível em um repositório configurado. Os repositórios configurados geralmente têm apenas a versão mais recente de um patch, que é um acúmulo de todas as atualizações anteriores, mas a versão mais recente pode não corresponder às regras da lista de referência de patches e ser omitida da operação de aplicação de patches.
   - Comandos: no RHEL 7, o comando yum equivalente para esse fluxo de trabalho é:

     ```
     sudo yum update-minimal --sec-severity=Critical,Important --bugfix -y
     ```

Para o AlmaLinux, o RHEL 8 e o Rocky Linux, os comandos dnf equivalentes para o fluxo de trabalho são:

```
sudo dnf update-minimal --sec-severity=Critical --bugfix -y ; \
sudo dnf update-minimal --sec-severity=Important --bugfix -y
```

> **Note**
>
> No Alma Linux, RHEL e Rocky LinuxRocky Linux, o Patch Manager pode instalar versões de dependências transitivas das instaladas pelos comandos `dnf` equivalentes. Dependências transitivas são pacotes que são instalados automaticamente para satisfazer os requisitos de outros pacotes (dependências de dependências).
> Por exemplo, o `dnf upgrade-minimal --security` instala as versões mínimas das dependências transitivas necessárias para resolver problemas de segurança conhecidos, enquanto o Patch Manager instala as versões mais recentes disponíveis das mesmas dependências transitivas.

Cenário 2: atualizações não relacionadas à segurança incluídas

- Aplica-se a: listas de referência de patches personalizadas.
- Caixa de seleção Incluir atualizações não relacionadas à segurança: marcada.
- Patches aplicados: os patches em `updateinfo.xml` e os que não estão em `updateinfo.xml` são aplicados (atualizações de segurança e não relacionadas à segurança).
- Comandos: no RHEL 7, o comando yum equivalente para esse fluxo de trabalho é:

```
sudo yum update --security --bugfix -y
```

No AlmaLinux 8 e 9, no RHEL 8 e 9 e no Rocky Linux 8 e 9, o comando dnf equivalente para esse fluxo de trabalho é:

```
sudo dnf update --security --bugfix -y
```

> **Note**
>
> Novos pacotes que substituem pacotes que se tornaram obsoletos com nomes diferentes serão instalados se os seguintes comandos `yum` ou `dnf` forem executados fora do Patch Manager. Porém, eles não são instalados por meio de operações do Patch Manager equivalentes.

8. O nó gerenciado será reinicializado se todas as atualizações forem instaladas. (Exceção: Se o parâmetro `RebootOption` estiver definido como `NoReboot` no documento `AWS-RunPatchBaseline`, o nó gerenciado não será reinicializado depois que o Patch Manager for executado. Para obter mais informações, consulte Nome do parâmetro: `RebootOption`).

SLES

Em nós gerenciados do SUSE Linux Enterprise Server (SLES), o fluxo de trabalho da instalação de patches é o seguinte:

1. Se uma lista de patches for especificada usando um URL https ou um URL de estilo caminho do Amazon Simple Storage Service (Amazon S3), que usa o parâmetro do `InstallOverrideList` para os documentos `AWS-RunPatchBaseline` ou `AWS-RunPatchBaselineAssociation`, os patches listados serão instalados e as de etapas de 2 a 7 serão ignoradas.
2. Aplique o GlobalFilters conforme especificado na linha de base de patch, mantendo apenas os pacotes qualificados para processamento adicional.
3. Aplique o ApprovalRules conforme especificado na lista de referência de patches. Cada regra de aprovação pode definir um pacote como aprovado.

   As regras de aprovação, no entanto, também dependem da seleção da opção Include nonsecurity updates (Incluir atualizações não relacionadas à segurança) ao criar ou atualizar pela última vez a lista de referência de patches.

   Se nenhuma atualização que não seja de segurança for excluída, uma regra implícita será aplicada para selecionar apenas os pacotes com atualizações nos repositórios de segurança. Para cada pacote, a versão candidata do pacote (que geralmente é a versão mais recente) deve fazer parte de um repo de segurança.

Se atualizações não relacionadas à segurança forem incluídas, os patches de outros repositórios também serão considerados.

4. Aplique o ApprovedPatches conforme especificado na lista de referência de patches. Os patches aprovados têm aprovação para atualizar, mesmo que sejam descartados por GlobalFilters ou mesmo que nenhuma regra de aprovação especificada em ApprovalRules conceda a aprovação.
5. Aplique o RejectedPatches conforme especificado na lista de referência de patches. Os patches rejeitados são removidos da lista de patches aprovados e não serão aplicados.
6. Se várias versões de um patch forem aprovadas, a versão mais recente será aplicada.
7. A API de atualização do Zypper é aplicada a patches aprovados.
8. O nó gerenciado será reinicializado se todas as atualizações forem instaladas. (Exceção: Se o parâmetro `RebootOption` estiver definido como `NoReboot` no documento `AWS-RunPatchBaseline`, o nó gerenciado não será reinicializado depois que o Patch Manager for executado. Para obter mais informações, consulte Nome do parâmetro: `RebootOption`).

Ubuntu Server

Em nós gerenciados do Ubuntu Server, o fluxo de trabalho da instalação de patches é o seguinte:

1. Se uma lista de patches for especificada usando um URL https ou um URL de estilo caminho do Amazon Simple Storage Service (Amazon S3), que usa o parâmetro do `InstallOverrideList` para os documentos `AWS-RunPatchBaseline` ou `AWS-RunPatchBaselineAssociation`, os patches listados serão instalados e as de etapas de 2 a 7 serão ignoradas.
2. Se uma atualização estiver disponível para o `python3-apt` (uma interface de biblioteca Python para `libapt`), ela será atualizada para a versão mais recente. (Este pacote não relacionado à segurança é atualizado mesmo se você não selecionou a opção Include nonsecurity updates (Incluir atualizações não relacionadas à segurança)).
3. Aplique o GlobalFilters conforme especificado na linha de base de patch, mantendo apenas os pacotes qualificados para processamento adicional.
4. Aplique o ApprovalRules conforme especificado na lista de referência de patches. Cada regra de aprovação pode definir um pacote como aprovado.

> **Note**
>
> Como não é possível determinar de forma confiável as datas de lançamento dos pacotes de atualização do Ubuntu Server, as opções de aprovação automática não são compatíveis com esse sistema operacional.

As regras de aprovação, no entanto, também dependem da seleção da opção Include nonsecurity updates (Incluir atualizações não relacionadas à segurança) ao criar ou atualizar pela última vez a lista de referência de patches.

Se nenhuma atualização que não seja de segurança for excluída, uma regra implícita será aplicada para selecionar apenas os pacotes com atualizações nos repositórios de segurança. Para cada pacote, a versão candidata do pacote (que geralmente é a versão mais recente) deve fazer parte de um repo de segurança.

Se atualizações não relacionadas à segurança forem incluídas, os patches de outros repositórios também serão considerados.

As regras de aprovação, no entanto, também dependem da seleção da opção Include nonsecurity updates (Incluir atualizações não relacionadas à segurança) ao criar ou atualizar pela última vez a lista de referência de patches.

> **Note**
>
> Para cada versão do Ubuntu Server, as versões candidatas a patches são limitadas aos patches que fazem parte do repositório associado a essa versão, da seguinte forma:
>
> - Ubuntu Server 14.04 LTS: `trusty-security`
> - Ubuntu Server 16.04 LTS: `xenial-security`
> - Ubuntu Server 18.04 LTS: `bionic-security`
> - Ubuntu Server 20.04 LTS): `focal-security`
> - Ubuntu Server 20.10 STR: `groovy-security`
> - Ubuntu Server 22.04 LTS: `jammy-security`
> - Ubuntu Server 23.04 (`lunar-security`)
> - Ubuntu Server 23.10 (`mantic-security`)

- Ubuntu Server 24.04 LTS (`noble-security`)
- Ubuntu Server 24.10 (`oracular-security`)
- Ubuntu Server 25.04 (`plucky-security`)

5. Aplique o ApprovedPatches conforme especificado na lista de referência de patches. Os patches aprovados têm aprovação para atualizar, mesmo que sejam descartados por GlobalFilters ou mesmo que nenhuma regra de aprovação especificada em ApprovalRules conceda a aprovação.
6. Aplique o RejectedPatches conforme especificado na lista de referência de patches. Os patches rejeitados são removidos da lista de patches aprovados e não serão aplicados.
7. A biblioteca de APT é usada para atualizar pacotes.

> **Note**
>
> O Patch Manager não oferece suporte ao uso da opção `Pin-Priority` de APT para atribuir prioridades aos pacotes. O Patch Manager agrega as atualizações disponíveis de todos os repositórios habilitados e seleciona a atualização mais recente que corresponde à lista de referência de cada pacote instalado.

8. O nó gerenciado será reinicializado se todas as atualizações forem instaladas. (Exceção: Se o parâmetro `RebootOption` estiver definido como `NoReboot` no documento `AWS-RunPatchBaseline`, o nó gerenciado não será reinicializado depois que o Patch Manager for executado. Para obter mais informações, consulte Nome do parâmetro: `RebootOption`).

Windows Server

Quando uma operação de aplicação de patch é executada em um nó gerenciado do Windows Server, o nó solicita um snapshot da lista de referência de patches apropriada no Systems Manager. Esse snapshot contém a lista de todas as atualizações disponíveis na linha de base de patch que foram aprovadas para implantação. Essa lista de atualizações é enviada à API do Windows Update, que determina quais das atualizações são aplicáveis ao nó gerenciado e os instala conforme necessário. O Windows permite que somente a versão mais recente disponível de um KB seja instalada. O Patch Manager instala a versão mais recente de um KB quando ele, ou qualquer versão anterior do KB, corresponde à lista de referência de patches aplicada. Se todas as atualizações forem instaladas, o nó gerenciado será reinicializado posteriormente quantas vezes for preciso para concluir todas as correções necessárias.

(Exceção: Se o parâmetro `RebootOption` estiver definido como `NoReboot` no documento `AWS-RunPatchBaseline`, o nó gerenciado não será reinicializado depois que o Patch Manager for executado. Para obter mais informações, consulte Nome do parâmetro: `RebootOption`). O resumo da operação de patch pode ser encontrado na saída da solicitação do Run Command. Os logs adicionais podem ser encontrados em seu nó gerenciado da pasta `%PROGRAMDATA%\Amazon\PatchBaselineOperations\Logs`.

Como a API do Windows Update é usada para fazer download e instalar KBs, todas as configurações de política de grupo para Windows Update são respeitadas. Nenhuma configuração de política de grupo é necessária para usar o Patch Manager, mas todas as configurações definidas serão aplicadas, como para direcionar nós gerenciados para um servidor WSUS (Windows Server Update Services).

> **Note**
>
> Por padrão, o Windows baixa todas as KBs do site do Microsoft Windows Update porque o Patch Manager usa a API do Windows Update para conduzir o download e a instalação de KBs. Por isso, o nó gerenciado precisa acessar o site de atualização do Microsoft Windows Update, caso contrário, a aplicação de patches falhará. Uma alternativa é configurar um servidor WSUS para servir como repositório de KBs e configurar os nós gerenciados para direcionar esse WSUS, em vez de usar políticas de grupo.
> O Patch Manager pode fazer referência a IDs de KB ao criar listas de referência de patches personalizadas para Windows Server, por exemplo, quando uma lista de Patches aprovados ou uma lista de Patches rejeitados é incluída na configuração da lista de referência. Somente as atualizações que recebem um ID de KB no Microsoft Windows Update ou em um servidor WSUS são instaladas pelo Patch Manager. Atualizações sem ID de KB não são incluídas nas operações de aplicação de patches.
> Para obter mais informações sobre a criação de listas de referência de patches personalizadas, consulte os seguintes tópicos:
>
> - Criar uma lista de referência de patches personalizada para o Windows Server
> - Criar uma lista de referência de patches (CLI)
> - Formatos de nomes de pacotes para sistemas operacionais Windows

Como funcionam as regras de linha de base de patch em sistemas baseados no Linux

As regras na linha de base de patch para distribuições do Linux funcionam de forma diferente dependendo do tipo de distribuição. Ao contrário das atualizações de patch em nós gerenciados do Windows Server, as regras são avaliadas em cada nó para levar em conta os repositórios configurados na instância. O Patch Manager, uma ferramenta do AWS Systems Manager, usa o gerenciador de pacotes nativo para conduzir a instalação de patches aprovados pela lista de referência de patches.

Para tipos de sistema operacional baseados em Linux que relatem um nível de gravidade para patches, o Patch Manager usa o nível de gravidade relatado pelo editor do software para o aviso de atualização ou patch individual. O Patch Manager não deriva níveis de gravidade de fontes de terceiros, como o Sistema comum de pontuação de vulnerabilidades (CVSS), ou a partir de métricas lançadas pelo Banco de dados nacional de vulnerabilidades (NVD).

Tópicos

- Como as regras de lista de referência de patches funcionam no Amazon Linux 1, Amazon Linux 2, Amazon Linux 2022 e Amazon Linux 2023
- Como as regras de lista de referência de patches funcionam no CentOS e CentOS Stream
- Como as regras de linha de base de patch funcionam no Debian Server e no Raspberry Pi OS
- Como as regras de linha de base de patch funcionam no macOS
- Como as regras de linha de base de patch funcionam no Oracle Linux
- Como as regras de lista de referência de patches funcionam no AlmaLinux, no RHEL e no Rocky Linux
- Como as regras de linha de base de patch funcionam no SUSE Linux Enterprise Server
- Como as regras de linha de base de patch funcionam no Ubuntu Server

Como as regras de lista de referência de patches funcionam no Amazon Linux 1, Amazon Linux 2, Amazon Linux 2022 e Amazon Linux 2023

> **Note**
>
> O Amazon Linux 2023 (AL2023) usa repositórios versionados que podem ser bloqueados em uma versão específica por meio de uma ou mais configurações do sistema. Para todas as operações de aplicação de patches nas instâncias do EC2 AL2023, o Patch Manager usa as

versões mais recentes do repositório, independentemente da configuração do sistema. Para obter mais informações, consulte Deterministic upgrades through versioned repositories no Amazon Linux 2023 User Guide.

Em instâncias do Amazon Linux 1, do Amazon Linux 2, do Amazon Linux 2022 e do Amazon Linux 2023, o processo de seleção de patches é o seguinte:

1. No nó gerenciado, a biblioteca do YUM (Amazon Linux 1, Amazon Linux 2) ou a biblioteca do DNF (Amazon Linux 2022 e Amazon Linux 2023) acessa o arquivo `updateinfo.xml` para cada repositório configurado.

   Se nenhum arquivo `updateinfo.xml` for encontrado, a instalação dos patches depende das configurações de Incluir atualizações não relacionadas a segurança e Autoaprovação. Por exemplo, se forem permitidas atualizações não relacionadas à segurança, elas serão instaladas quando o tempo de autoaprovação chegar.
2. Toda notificação de atualização em `updateinfo.xml` inclui vários atributos que denotam as propriedades dos pacotes na notificação, tal como descrito na tabela a seguir.

   Atributos de notificação de atualização

Atributo	Descrição
type	Corresponde ao valor do atributo de chave de classificação no tipo de dados PatchFilter da linha de base de patch. Denota o tipo de pacote incluído na notificação de atualização. Você pode exibir a lista de valores compatíveis usando o comando da AWS CLI, describe-patch-properties, ou a operação DescribePatchProperties da API. Você também pode visualizar a lista na área Approval rules (Regras de aprovação) da página Create patch baseline (Criar lista de referência do patch) ou da página Edit patch baseline (Editar lista de referência do patch) no console do Systems Manager.

Atributo	Descrição
severidade	Corresponde ao valor do atributo de chave de gravidade no tipo de dados PatchFilter da linha de base de patch. Denota a gravidade dos pacotes incluídos na notificação de atualização. Normalmente, só é aplicável a notificações de atualização de segurança. Você pode exibir a lista de valores compatíveis usando o comando da AWS CLI, describe-patch-properties, ou a operação DescribePatchProperties da API. Você também pode visualizar a lista na área Approval rules (Regras de aprovação) da página Create patch baseline (Criar lista de referência do patch) ou da página Edit patch baseline (Editar lista de referência do patch) no console do Systems Manager.
update_id	Denota o ID do Advisory, como ALAS-2017-867. O ID do Advisory pode ser usado no atributo ApprovedPatches ou RejectedPatches na linha de base de patch.
references	Contém informações adicionais sobre o notificação de atualização, como um ID do CVE (formato: CVE-2017-1234567). O ID do CVE pode ser usado no atributo ApprovedPatches ou RejectedPatches na linha de base de patch.

Atributo	Descrição
updated	Corresponde a ApproveAfterDays na linha de base de patch. Denota a data de lançamento (data de atualização) dos pacotes incluídos na notificação de atualização. A comparação entre o carimbo de data/hora atual e o valor desse atributo mais `ApproveAfterDays` é usado para determinar se o patch está aprovado para implantação.

Para obter mais informações sobre os formatos aceitos de listas de patches aprovados e rejeitados, consulte Sobre formatos de nomes de pacotes para listas de patches aprovados e rejeitados.

3. O produto do nó gerenciado é determinado pelo SSM Agent. Esse atributo corresponde ao valor do atributo de chave de produto no tipo de dados PatchFilter.
4. Os pacotes são selecionados para a atualização de acordo com as seguintes diretrizes:

Opção de segurança	Seleção de patches
Linhas de base de patch padrão predefinidas fornecidas pela AWS e linhas de base de patch personalizadas em que Include non-security updates não está marcada	Para cada notificação de atualização `updateinfo.xml` , a linha de base de patch é usada como filtro, permitindo que apenas os pacotes qualificados sejam incluídos na atualização. Se vários pacotes forem aplicáveis depois de aplicar a definição da linha de base de patch, será usada a versão mais recente. No Amazon Linux 1 e no Amazon Linux 2, o comando yum equivalente para esse fluxo de trabalho é: `sudo yum update-minimal --sec-severity=Critical,Important --bugfix -y`

Opção de segurança	Seleção de patches
	No Amazon Linux 2022 e no Amazon Linux 2023, o comando dnf equivalente para esse fluxo de trabalho é: `sudo dnf upgrade-minimal --sec-severity=Critical --sec-severity=Important --bugfix -y`

Opção de segurança	Seleção de patches

Linhas de base de patch personalizadas em que a caixa de seleção Incluir atualizações não relacionadas a segurança está marcada com uma lista de GRAVIDADE de `[Critical, Important]` e uma lista de CLASSIFICAÇÃO de `[Security, Bugfix]`	Além de aplicar as atualizações de segurança que foram selecionadas no `updateinfo.xml` , o Patch Manager aplicará as atualizações não relacionadas a segurança que, do contrário, atenderem às regras de filtragem de patches.
	
	No Linux e no Amazon Linux 2, o comando yum equivalente para esse fluxo de trabalho é:
	
	`sudo yum update --security --sec-severity=Critical,Important --bugfix -y`
	
	No Amazon Linux 2022 e no Amazon Linux 2023, o comando dnf equivalente para esse fluxo de trabalho é:
	
	`sudo dnf upgrade --security --sec-severity=Critical --sec-severity=Important --bugfix -y`
	
	Note
	Novos pacotes que substituem pacotes que se tornaram obsoletos com nomes diferentes serão instalados se os seguintes comandos `yum` ou `dnf` forem executados fora do Patch Manager. Porém, eles não são instalados por meio de operações do Patch Manager equivalentes.

Para obter informações sobre valores de status de conformidade de patches, consulte Valores de estados de conformidade de patches.

Como as regras de lista de referência de patches funcionam no CentOS e CentOS Stream

O CentOS e os repositórios padrão do CentOS Stream não incluem um arquivo `updateinfo.xml`. No entanto, os repositórios personalizados criados ou usados por você podem incluir esse arquivo. Neste tópico, referências a `updateinfo.xml` se aplicam somente a esses repositórios personalizados.

No CentOS e CentOS Stream, o processo de seleção de patches é o seguinte:

1. Em um nó gerenciado, a biblioteca do YUM (nas versões do CentOS 6.x e 7.x) ou a biblioteca do DNF (no CentOS 8.x e CentOS Stream) acessa o arquivo `updateinfo.xml`, se ele existir em um repositório personalizado, para cada repositório configurado.

   Se nenhum `updateinfo.xml` for encontrado, o que sempre inclui o repositório padrão, a instalação dos patches depende das configurações de Incluir atualizações não relacionadas a segurança e Autoaprovação. Por exemplo, se forem permitidas atualizações não relacionadas à segurança, elas serão instaladas quando o tempo de autoaprovação chegar.
2. Se `updateinfo.xml` estiver presente, toda notificação de atualização no arquivo incluirá vários atributos que denotam as propriedades dos pacotes na notificação, tal como descrito na tabela a seguir.

   Atributos de notificação de atualização

Atributo	Descrição
type	Corresponde ao valor do atributo de chave de classificação no tipo de dados PatchFilter da linha de base de patch. Denota o tipo de pacote incluído na notificação de atualização. Você pode exibir a lista de valores compatíveis usando o comando da AWS CLI, describe-patch-properties, ou a operação DescribePatchProperties da API. Você também pode visualizar a lista na área Approval rules (Regras de aprovação) da página Create patch baseline (Criar lista de referência do

Atributo	Descrição
	patch) ou da página Edit patch baseline (Editar lista de referência do patch) no console do Systems Manager.
severidade	Corresponde ao valor do atributo de chave de gravidade no tipo de dados PatchFilter da linha de base de patch. Denota a gravidade dos pacotes incluídos na notificação de atualização. Normalmente, só é aplicável a notificações de atualização de segurança. Você pode exibir a lista de valores compatíve is usando o comando da AWS CLI, describe-patch-properties, ou a operação DescribeP atchProperties da API. Você também pode visualizar a lista na área Approval rules (Regras de aprovação) da página Create patch baseline (Criar lista de referência do patch) ou da página Edit patch baseline (Editar lista de referência do patch) no console do Systems Manager.
update_id	Denota o ID do Advisory, como CVE-2019-17055. O ID do Advisory pode ser usado no atributo ApprovedPatches ou RejectedP atches na linha de base de patch.
references	Contém informações adicionais sobre a notificação de atualização, como um ID do CVE (formato: CVE-2019-17055) ou ID do Bugzilla (formato: 1463241). O ID do CVE e o ID do Bugzilla podem ser usados no atributo ApprovedPatches ou RejectedPatches na linha de base de patch.

Atributo	Descrição
updated	Corresponde a ApproveAfterDays na linha de base de patch. Denota a data de lançamento (data de atualização) dos pacotes incluídos na notificação de atualização. A comparação entre o carimbo de data/hora atual e o valor desse atributo mais `ApproveAfterDays` é usado para determinar se o patch está aprovado para implantação.

Para obter mais informações sobre os formatos aceitos de listas de patches aprovados e rejeitados, consulte Sobre formatos de nomes de pacotes para listas de patches aprovados e rejeitados.

3. Em todos os casos, o produto do nó gerenciado é determinado pelo SSM Agent. Esse atributo corresponde ao valor do atributo de chave de produto no tipo de dados PatchFilter.
4. Os pacotes são selecionados para a atualização de acordo com as seguintes diretrizes:

Opção de segurança	Seleção de patches
Linhas de base de patch padrão predefinidas fornecidas pela AWS e linhas de base de patch personalizadas em que Include non-security updates não está marcada	Para cada notificação de atualização em `updateinfo.xml` , se ele existir em um repositório personalizado, a lista de referência de patches será usada como filtro, permitindo que apenas os pacotes qualificados sejam incluídos na atualização. Se vários pacotes forem aplicáveis depois de aplicar a definição da linha de base de patch, será usada a versão mais recente. Para o CentOS 6 e 7, onde `updateinfo.xml` está presente, o comando yum equivalente para esse fluxo de trabalho é:

Opção de segurança	Seleção de patches

	`sudo yum update-minimal --sec-severity=Critical,Important --bugfix -y`
	
	Para o CentOS 8 e CentOS Stream, onde `updateinfo.xml` está presente, o comando yum equivalente para esse fluxo de trabalho é:
	
	`sudo dnf upgrade-minimal --sec-severity=Critical --sec-severity=Important --bugfix -y`

Opção de segurança	Seleção de patches

Linhas de base de patch personalizadas em que a caixa de seleção Incluir atualizações não relacionadas a segurança está marcada com uma lista de GRAVIDADE de `[Critical, Important]` e uma lista de CLASSIFICAÇÃO de `[Security, Bugfix]`	Além de aplicar as atualizações de segurança que foram selecionadas em `updateinfo.xml`, se ele existir em um repositório personalizado, o Patch Manager aplicará as atualizações não relacionadas a segurança que, de outra forma, atenderem às regras de filtragem de patches.
	
	Para o CentOS 6 e 7, onde `updateinfo.xml` está presente, o comando yum equivalente para esse fluxo de trabalho é:
	
	`sudo yum update --sec-severity=Critical,Important --bugfix -y`
	
	Para o CentOS 8 e CentOS Stream, onde `updateinfo.xml` está presente, o comando yum equivalente para esse fluxo de trabalho é:
	
	`sudo dnf upgrade --security --sec-severity=Critical --sec-severity=Important --bugfix -y`
	
	Para repositórios padrão e repositórios personalizados sem `updateinfo.xml`, você deve marcar a caixa de seleção Incluir atualizações não relacionadas à segurança para atualizar pacotes do sistema operacional (SO).
	
	Note
	Novos pacotes que substituem pacotes que se tornaram obsoletos com nomes diferentes serão instalado

Opção de segurança	Seleção de patches
	s se os seguintes comandos `yum` ou `dnf` forem executados fora do Patch Manager. Porém, eles não são instalados por meio de operações do Patch Manager equivalentes.

Para obter informações sobre valores de status de conformidade de patches, consulte Valores de estados de conformidade de patches.

Como as regras de linha de base de patch funcionam no Debian Server e no Raspberry Pi OS

No Debian Server e no Raspberry Pi OS (anteriormente Raspbian), o serviço de linha de base de patch oferece filtragem nos campos Priority (Prioridade) e Section (Seção). Esses campos geralmente estão presentes em todos os pacotes do Debian Server e do Raspberry Pi OS. Para determinar se um patch é selecionado pela lista de referência de patches, o Patch Manager faz o seguinte:

1. Em sistemas Debian Server e Raspberry Pi OS, o equivalente a `sudo apt-get update` é executado para atualizar a lista de pacotes disponíveis. Os repos não são configurados e os dados são extraídas dos repos configurados em uma lista `sources`.
2. Se uma atualização estiver disponível para o `python3-apt` (uma interface de biblioteca Python para `libapt`), ela será atualizada para a versão mais recente. (Este pacote não relacionado à segurança é atualizado mesmo se você não selecionou a opção Include nonsecurity updates (Incluir atualizações não relacionadas à segurança)).

> **Important**
>
> Somente no Debian Server 8: como os sistemas operacionais Debian Server 8.* se referem a um repositório de pacotes obsoleto (`jessie-backports`), o Patch Manager executa as seguintes etapas adicionais para garantir que as operações de patch sejam bem-sucedidas:
>
> a. Em seu nó gerenciado, a referência ao repositório `jessie-backports` é comentada na lista de locais de origem (`/etc/apt/sources.list.d/jessie-backports`). Como resultado, nenhuma tentativa é feita para baixar patches desse local.

b. Uma chave de assinatura da atualização de segurança do Stretch é importada. Essa chave fornece as permissões necessárias para as operações de atualização e instalação nas distribuições do Debian Server 8.*.

c. Uma operação `apt-get` é executada para garantir que a versão mais recente de `python3-apt` seja instalada antes do início do processo de aplicação de patch.

d. Após a conclusão do processo de instalação, a referência ao repositório `jessie-backports` é restaurada e a chave de assinatura é removida das chaves de origem apt. Isso é feito para deixar a configuração do sistema como estava antes da operação de aplicação de patch.

3. Em seguida, as listas GlobalFilters, ApprovalRules, ApprovedPatches e RejectedPatches são aplicadas.

> **Note**
>
> Como não é possível determinar de forma confiável as datas de lançamento dos pacotes de atualização do Debian Server, as opções de aprovação automática não são compatíveis com esse sistema operacional.

As regras de aprovação, no entanto, também dependem da seleção da opção Include nonsecurity updates (Incluir atualizações não relacionadas à segurança) ao criar ou atualizar pela última vez a lista de referência de patches.

Se nenhuma atualização que não seja de segurança for excluída, uma regra implícita será aplicada para selecionar apenas os pacotes com atualizações nos repositórios de segurança. Para cada pacote, a versão candidata do pacote (que geralmente é a versão mais recente) deve fazer parte de um repo de segurança. Neste caso, para o Debian Server, versões candidatas de patch são limitadas a patches incluídos nos seguintes repositórios:

Esses repositórios são nomeados da seguinte forma:

- Debian Server 8: `debian-security jessie`
- Debian Server e Raspberry Pi OS 9: `debian-security stretch`
- Debian Server 10: `debian-security buster`
- Debian Server 11: `debian-security bullseye`
- Debian Server 12: `debian-security bookworm`

Se atualizações não relacionadas à segurança forem incluídas, os patches de outros repositórios também serão considerados.

Para obter mais informações sobre os formatos aceitos de listas de patches aprovados e rejeitados, consulte Sobre formatos de nomes de pacotes para listas de patches aprovados e rejeitados.

Para visualizar o conteúdo dos campos Priority e Section , execute o comando `aptitude` a seguir:

> **Note**
>
> Pode ser necessário primeiro instalar o Aptitude nos sistemas Debian Server.

```
aptitude search -F '%p %P %s %t %V#' '~U'
```

Em resposta a esse comando, todos os pacotes atualizáveis são relatados no seguinte formato:

```
name, priority, section, archive, candidate version
```

Para obter informações sobre valores de status de conformidade de patches, consulte Valores de estados de conformidade de patches.

Como as regras de linha de base de patch funcionam no macOS

No macOS, o processo de seleção de patches é como o seguinte:

1. Em um nó gerenciado, o Patch Manager acessa o conteúdo analisado do `InstallHistory.plist` e identifica nomes e versões de pacotes.

   Para obter detalhes sobre o processo de análise, consulte a seção macOSem Como os patches são instalados.
2. O produto do nó gerenciado é determinado pelo SSM Agent. Esse atributo corresponde ao valor do atributo de chave de produto no tipo de dados PatchFilter.
3. Os pacotes são selecionados para a atualização de acordo com as seguintes diretrizes:

Opção de segurança	Seleção de patches
Linhas de base de patch padrão predefinidas fornecidas pela AWS e linhas de base de patch personalizadas em que Include non-security updates não está marcada	Para cada atualização de pacote disponível, a lista de referência de patches é usada como filtro, permitindo que somente os pacotes qualificados sejam incluídos na atualização. Se vários pacotes forem aplicáveis depois de aplicar a definição da linha de base de patch, será usada a versão mais recente.
Linhas de base de patch personalizadas em que a caixa de seleção Include non-security updates está marcada	Além de aplicar as atualizações de segurança que foram selecionadas em `InstallHistory.plist`, o Patch Manager aplicará as atualizações não relacionadas a segurança que de outra forma atenderem às regras de filtragem de patches.

Para obter informações sobre valores de status de conformidade de patches, consulte Valores de estados de conformidade de patches.

Como as regras de linha de base de patch funcionam no Oracle Linux

No Oracle Linux, o processo de seleção de patches é como o seguinte:

1. Em um nó gerenciado, a biblioteca do YUM acessa o arquivo `updateinfo.xml` para cada repositório configurado.

   > **Note**
   >
   > O arquivo `updateinfo.xml` talvez não esteja disponível se o repo não for gerenciado pela Oracle. Se nenhum `updateinfo.xml` for encontrado, a instalação dos patches depende das configurações de Incluir atualizações não relacionadas a segurança e Autoaprovação. Por exemplo, se forem permitidas atualizações não relacionadas à segurança, elas serão instaladas quando o tempo de autoaprovação chegar.

2. Toda notificação de atualização em `updateinfo.xml` inclui vários atributos que denotam as propriedades dos pacotes na notificação, tal como descrito na tabela a seguir.

Atributos de notificação de atualização

Atributo	Descrição
type	Corresponde ao valor do atributo de chave de classificação no tipo de dados PatchFilter da linha de base de patch. Denota o tipo de pacote incluído na notificação de atualização. Você pode exibir a lista de valores compatíveis usando o comando da AWS CLI, describe-patch-properties, ou a operação DescribePatchProperties da API. Você também pode visualizar a lista na área Approval rules (Regras de aprovação) da página Create patch baseline (Criar lista de referência do patch) ou da página Edit patch baseline (Editar lista de referência do patch) no console do Systems Manager.
severidade	Corresponde ao valor do atributo de chave de gravidade no tipo de dados PatchFilter da linha de base de patch. Denota a gravidade dos pacotes incluídos na notificação de atualização. Normalmente, só é aplicável a notificações de atualização de segurança. Você pode exibir a lista de valores compatíveis usando o comando da AWS CLI, describe-patch-properties, ou a operação DescribePatchProperties da API. Você também pode visualizar a lista na área Approval rules (Regras de aprovação) da página Create patch baseline (Criar lista de referência do patch) ou da página Edit patch baseline (Editar lista de referência do patch) no console do Systems Manager.

Atributo	Descrição
update_id	Denota o ID do Advisory, como CVE-2019-17055. O ID do Advisory pode ser usado no atributo ApprovedPatches ou RejectedPatches na linha de base de patch.
references	Contém informações adicionais sobre a notificação de atualização, como um ID do CVE (formato: CVE-2019-17055) ou ID do Bugzilla (formato: 1463241). O ID do CVE e o ID do Bugzilla podem ser usados no atributo ApprovedPatches ou RejectedPatches na linha de base de patch.
updated	Corresponde a ApproveAfterDays na linha de base de patch. Denota a data de lançamento (data de atualização) dos pacotes incluídos na notificação de atualização. A comparação entre o carimbo de data/hora atual e o valor desse atributo mais `ApproveAfterDays` é usado para determinar se o patch está aprovado para implantação.

Para obter mais informações sobre os formatos aceitos de listas de patches aprovados e rejeitados, consulte Sobre formatos de nomes de pacotes para listas de patches aprovados e rejeitados.

3. O produto do nó gerenciado é determinado pelo SSM Agent. Esse atributo corresponde ao valor do atributo de chave de produto no tipo de dados PatchFilter.
4. Os pacotes são selecionados para a atualização de acordo com as seguintes diretrizes:

Opção de segurança	Seleção de patches
Linhas de base de patch padrão predefinidas fornecidas pela AWS e linhas de base de	Para cada notificação de atualização `updateinfo.xml` , a linha de base de patch é usada como filtro, permitindo que apenas

Opção de segurança	Seleção de patches

patch personalizadas em que Include non-security updates não está marcada	os pacotes qualificados sejam incluídos na atualização. Se vários pacotes forem aplicáveis depois de aplicar a definição da linha de base de patch, será usada a versão mais recente.
	
	Para nós gerenciados da versão 7, o comando yum equivalente para esse fluxo de trabalho é:
	
	`sudo yum update-minimal --sec-severity=Important,Moderate --bugfix -y`
	
	Para nós gerenciados da versão 8 e 9, o comando dnf equivalente para esse fluxo de trabalho é:
	
	`sudo dnf upgrade-minimal --security --sec-severity=Moderate --sec-severity=Important`

Opção de segurança	Seleção de patches

Linhas de base de patch personalizadas em que a caixa de seleção Incluir atualizações não relacionadas a segurança está marcada com uma lista de GRAVIDADE de `[Critical, Important]` e uma lista de CLASSIFICAÇÃO de `[Security, Bugfix]`	Além de aplicar as atualizações de segurança que foram selecionadas no `updateinfo.xml` , o Patch Manager aplicará as atualizações não relacionadas a segurança que, do contrário, atenderem às regras de filtragem de patches.
	
	Para nós gerenciados da versão 7, o comando yum equivalente para esse fluxo de trabalho é:
	
	`sudo yum update --security --sec-severity=Critical,Important --bugfix -y`
	
	Para nós gerenciados da versão 8 e 9, o comando dnf equivalente para esse fluxo de trabalho é:
	
	`sudo dnf upgrade --security --sec-severity=Critical, --sec-severity=Important --bugfix y`
	
	Note
	Novos pacotes que substituem pacotes que se tornaram obsoletos com nomes diferentes serão instalados se os seguintes comandos `yum` ou `dnf` forem executados fora do Patch Manager. Porém, eles não são instalados por meio de operações do Patch Manager equivalentes.

Para obter informações sobre valores de status de conformidade de patches, consulte Valores de estados de conformidade de patches.

Como as regras de lista de referência de patches funcionam no AlmaLinux, no RHEL e no Rocky Linux

No AlmaLinux, no Red Hat Enterprise Linux (RHEL) e no Rocky Linux, o processo de seleção de patches é o seguinte:

1. Em um nó gerenciado, a biblioteca do YUM (RHEL 7) ou a biblioteca do DNF (AlmaLinux 8 e 9, RHEL 8, 9 e 10 e Rocky Linux 8 e 9) acessa o arquivo `updateinfo.xml` para cada repositório configurado.

   > **Note**
   >
   > O arquivo `updateinfo.xml` talvez não esteja disponível se o repo não for gerenciado pela Red Hat. Se não for encontrado nenhum `updateinfo.xml`, nenhum patch será aplicado.

2. Toda notificação de atualização em `updateinfo.xml` inclui vários atributos que denotam as propriedades dos pacotes na notificação, tal como descrito na tabela a seguir.

   Atributos de notificação de atualização

Atributo	Descrição
type	Corresponde ao valor do atributo de chave de classificação no tipo de dados PatchFilter da linha de base de patch. Denota o tipo de pacote incluído na notificação de atualização. Você pode exibir a lista de valores compatíveis usando o comando da AWS CLI, describe-patch-properties, ou a operação DescribePatchProperties da API. Você também pode visualizar a lista na área Approval rules (Regras de aprovação) da página Create patch baseline (Criar lista de referência do patch) ou da página Edit patch baseline

Atributo	Descrição
	(Editar lista de referência do patch) no console do Systems Manager.
severidade	Corresponde ao valor do atributo de chave de gravidade no tipo de dados PatchFilter da linha de base de patch. Denota a gravidade dos pacotes incluídos na notificação de atualização. Normalmente, só é aplicável a notificações de atualização de segurança. Você pode exibir a lista de valores compatíveis usando o comando da AWS CLI, describe-patch-properties, ou a operação DescribePatchProperties da API. Você também pode visualizar a lista na área Approval rules (Regras de aprovação) da página Create patch baseline (Criar lista de referência do patch) ou da página Edit patch baseline (Editar lista de referência do patch) no console do Systems Manager.
update_id	Denota o ID do Advisory, como RHSA-2017:0864. O ID do Advisory pode ser usado no atributo ApprovedPatches ou RejectedPatches na linha de base de patch.
references	Contém informações adicionais sobre notificação de atualização, como um ID do CVE (formato: CVE-2017-1000371) ou ID do Bugzilla (formato: 1463241). O ID do CVE e o ID do Bugzilla podem ser usados no atributo ApprovedPatches ou RejectedPatches na linha de base de patch.

Atributo	Descrição
updated	Corresponde a ApproveAfterDays na linha de base de patch. Denota a data de lançamento (data de atualização) dos pacotes incluídos na notificação de atualização. A comparação entre o carimbo de data/hora atual e o valor desse atributo mais `ApproveAfterDays` é usado para determinar se o patch está aprovado para implantação.

Para obter mais informações sobre os formatos aceitos de listas de patches aprovados e rejeitados, consulte Sobre formatos de nomes de pacotes para listas de patches aprovados e rejeitados.

3. O produto do nó gerenciado é determinado pelo SSM Agent. Esse atributo corresponde ao valor do atributo de chave de produto no tipo de dados PatchFilter.
4. Os pacotes são selecionados para a atualização de acordo com as seguintes diretrizes:

Opção de segurança	Seleção de patches
Linhas de base de patch padrão predefinidas fornecidas pela AWS e linhas de base de patch personalizadas em que Incluir atualizações não relacionadas a segurança não está marcada em nenhuma regra	Para cada notificação de atualização `updateinfo.xml` , a linha de base de patch é usada como filtro, permitindo que apenas os pacotes qualificados sejam incluídos na atualização. Se vários pacotes forem aplicáveis depois de aplicar a definição da linha de base de patch, será usada a versão mais recente. No RHEL 7, o comando yum equivalente para esse fluxo de trabalho é: `sudo yum update-minimal --sec-severity=Critical,Important --bugfix -y`

Opção de segurança	Seleção de patches
	No AlmaLinux 8 e 9, no RHEL 8, 9 e 10 e no Rocky Linux 8 e 9, o comando dnf equivalente para esse fluxo de trabalho é: `sudo dnf upgrade-minimal --sec-severity=Critical --sec-severity=Important --bugfix -y`

Opção de segurança	Seleção de patches

Linhas de base de patch personalizadas em que a caixa de seleção Incluir atualizações não relacionadas a segurança está marcada com uma lista de GRAVIDADE de `[Critical, Important]` e uma lista de CLASSIFICAÇÃO de `[Security, Bugfix]`	Além de aplicar as atualizações de segurança que foram selecionadas no `updateinfo.xml` , o Patch Manager aplicará as atualizações não relacionadas a segurança que, do contrário, atenderem às regras de filtragem de patches.
	
	No RHEL 7, o comando yum equivalente para esse fluxo de trabalho é:
	
	`sudo yum update --security --sec-severity=Critical,Important --bugfix -y`
	
	No AlmaLinux 8 e 9, no RHEL 8, 9 e 10 e no Rocky Linux 8 e 9, o comando dnf equivalente para esse fluxo de trabalho é:
	
	`sudo dnf upgrade --sec-severity=Critical --sec-severity=Important --bugfix -y`
	
	Note
	Novos pacotes que substituem pacotes que se tornaram obsoletos com nomes diferentes serão instalados se os seguintes comandos `yum` ou `dnf` forem executados fora do Patch Manager. Porém, eles não são instalados por meio de operações do Patch Manager equivalentes.

Para obter informações sobre valores de status de conformidade de patches, consulte Valores de estados de conformidade de patches.

Como as regras de linha de base de patch funcionam no SUSE Linux Enterprise Server

No SLES, cada patch inclui os atributos a seguir, que identificam as propriedades dos pacotes no patch:

- Categoria: corresponde ao valor do atributo de chave de classificação no tipo de dados PatchFilter da linha de base de patch. Denota o tipo de patch incluso na notificação de atualização.

  Você pode exibir a lista de valores compatíveis usando o comando da AWS CLI, describe-patch-properties, ou a operação DescribePatchProperties da API. Você também pode visualizar a lista na área Approval rules (Regras de aprovação) da página Create patch baseline (Criar lista de referência do patch) ou da página Edit patch baseline (Editar lista de referência do patch) no console do Systems Manager.
- Gravidade: corresponde ao valor do atributo de chave de gravidade no tipo de dados PatchFilter da lista de referência de patches. Indica o nível de severidade dos patches.

  Você pode exibir a lista de valores compatíveis usando o comando da AWS CLI, describe-patch-properties, ou a operação DescribePatchProperties da API. Você também pode visualizar a lista na área Approval rules (Regras de aprovação) da página Create patch baseline (Criar lista de referência do patch) ou da página Edit patch baseline (Editar lista de referência do patch) no console do Systems Manager.

O produto do nó gerenciado é determinado pelo SSM Agent. Esse atributo corresponde ao valor do atributo de chave do Produto no tipo de dados PatchFilter da linha de base de patch.

Para cada patch, a linha de base de patch é usada como filtro, permitindo que somente os pacotes qualificados sejam incluídos na atualização. Se vários pacotes forem aplicáveis depois de aplicar a definição da linha de base de patch, será usada a versão mais recente.

Para obter mais informações sobre os formatos aceitos de listas de patches aprovados e rejeitados, consulte Sobre formatos de nomes de pacotes para listas de patches aprovados e rejeitados.

Como as regras de linha de base de patch funcionam no Ubuntu Server

No Ubuntu Server, o serviço de linha de base de patch oferece filtragem nos campos Priority e Section . Esses campos geralmente estão presentes em todos os pacotes do Ubuntu Server. Para

determinar se um patch é selecionado pela lista de referência de patches, o Patch Manager faz o seguinte:

1. Em sistemas Ubuntu Server, o equivalente a `sudo apt-get update` é executado para atualizar a lista de pacotes disponíveis. Os repos não são configurados e os dados são extraídas dos repos configurados em uma lista `sources`.
2. Se uma atualização estiver disponível para o `python3-apt` (uma interface de biblioteca Python para `libapt`), ela será atualizada para a versão mais recente. (Este pacote não relacionado à segurança é atualizado mesmo se você não selecionou a opção Include nonsecurity updates (Incluir atualizações não relacionadas à segurança)).
3. Em seguida, as listas GlobalFilters, ApprovalRules, ApprovedPatches e RejectedPatches são aplicadas.

> **Note**
>
> Como não é possível determinar de forma confiável as datas de lançamento dos pacotes de atualização do Ubuntu Server, as opções de aprovação automática não são compatíveis com esse sistema operacional.

As regras de aprovação, no entanto, também dependem da seleção da opção Include nonsecurity updates (Incluir atualizações não relacionadas à segurança) ao criar ou atualizar pela última vez a lista de referência de patches.

Se nenhuma atualização que não seja de segurança for excluída, uma regra implícita será aplicada para selecionar apenas os pacotes com atualizações nos repositórios de segurança. Para cada pacote, a versão candidata do pacote (que geralmente é a versão mais recente) deve fazer parte de um repo de segurança. Neste caso, para o Ubuntu Server, versões candidatas de patch são limitadas a patches incluídos nos seguintes repositórios:

- Ubuntu Server 14.04 LTS: `trusty-security`
- Ubuntu Server 16.04 LTS: `xenial-security`
- Ubuntu Server 18.04 LTS: `bionic-security`
- Ubuntu Server 20.04 LTS: `focal-security`
- Ubuntu Server 20.10 STR: `groovy-security`
- Ubuntu Server 22.04 LTS (`jammy-security`)
- Ubuntu Server 23.04 (`lunar-security`)

- Ubuntu Server 23.10 (`mantic-security`)
- Ubuntu Server 24.04 LTS (`noble-security`)
- Ubuntu Server 24.10 (`oracular-security`)
- Ubuntu Server 25.04 (`plucky-security`)

Se atualizações não relacionadas à segurança forem incluídas, os patches de outros repositórios também serão considerados.

Para obter mais informações sobre os formatos aceitos de listas de patches aprovados e rejeitados, consulte Sobre formatos de nomes de pacotes para listas de patches aprovados e rejeitados.

Para visualizar o conteúdo dos campos Priority e Section , execute o comando `aptitude` a seguir:

> **Note**
>
> Pode ser necessário primeiro instalar o Aptitude em sistemas Ubuntu Server 16.

```
aptitude search -F '%p %P %s %t %V#' '~U'
```

Em resposta a esse comando, todos os pacotes atualizáveis são relatados no seguinte formato:

```
name, priority, section, archive, candidate version
```

Para obter informações sobre valores de status de conformidade de patches, consulte Valores de estados de conformidade de patches.

Principais diferenças entre a aplicação de patches em nós do Linux e em nós do Windows Server

Este tópico descreve diferenças importantes entre a aplicação de patches do Windows Server e do Linux no Patch Manager, uma ferramenta do AWS Systems Manager.

> **Note**
>
> Para aplicar patches a nós gerenciados do Linux, os nós devem estar em execução no SSM Agent versão 2.0.834.0 ou posterior.

Uma versão atualizada do SSM Agent é lançada sempre que novas ferramentas são adicionadas ao Systems Manager ou sempre que atualizações são feitas nas ferramentas existentes. Deixar de usar a versão mais recente do agente pode impedir que seu nó gerenciado use várias ferramentas do Systems Manager. Por isso, recomendamos automatizar o processo de manter o SSM Agent atualizado em suas máquinas. Para mais informações, consulte Automatizar atualizações do SSM Agent. Inscreva-se na página Notas de versão do SSM Agent no GitHub para receber notificações sobre atualizações do SSM Agent.

Diferença 1: avaliação de patches

O Patch Manager usa processos diferentes em nós gerenciados do Windows e do Linux para avaliar quais patches devem estar presentes.

Linux

Para patches do Linux, o Systems Manager avalia as regras da lista de referência de patches e a lista de patches aprovados e rejeitados em cada nó gerenciado. O Systems Manager deve avaliar o patch em cada nó porque o serviço recupera a lista de patches conhecidos e atualizações dos repositórios configurados em seu nó gerenciado.

Windows

Para aplicação de patch do Windows, o Systems Manager avalia as regras da lista de referência do patch e a lista de patches aprovados e rejeitados diretamente no serviço. Isso pode ser feito porque os patches do Windows são extraídos de um único repositório (Windows Update).

Diferença 2: patches **`Not Applicable`**

Devido à grande quantidade de pacotes disponíveis para sistemas operacionais Linux, o Systems Manager não relata detalhes sobre patches no estado Not Applicable (Não aplicável). Um patch `Not Applicable` é, por exemplo, um patch para o software Apache quando a instância não tiver o Apache instalado. O Systems Manager relata o número de patches `Not Applicable` no resumo, mas, se você chamar a API DescribeInstancePatches para um nó gerenciado, os dados retornados não incluirão patches com um estado `Not Applicable`. Esse comportamento é diferente do Windows.

Diferença 3: suporte a documentos do SSM

O documento do Systems Manager (documento SSM) `AWS-ApplyPatchBaseline` não oferece suporte a nós gerenciados do Linux. Para aplicar listas de referência de patches a nós gerenciados do Linux, do macOS e do Windows Server, o documento SSM recomendado é o `AWS-RunPatchBaseline`. Para obter mais informações, consulte Documentos do comando do SSM para aplicação de patches em nós gerenciados e Documento de comando do SSM para a aplicação de patches: `AWS-RunPatchBaseline`.

Diferença 4: patches de aplicativos

O foco principal do Patch Manager é aplicar patches a sistemas operacionais. No entanto, você também pode usar o Patch Manager para aplicar patches em algumas aplicações dos nós gerenciados.

Linux

Em sistemas operacionais Linux, o Patch Manager usa os repositórios configurados para atualizações e não diferencia entre patches de sistemas operacionais e de aplicações. Você pode usar o Patch Manager para definir de quais repositórios deseja buscar atualizações. Para obter mais informações, consulte Como especificar um repositório de origem de patches alternativo (Linux).

Windows

Em nós gerenciados do Windows Server, você pode aplicar regras de aprovação, bem como exceções de patches Aprovados e Rejeitados, para aplicações lançadas pela Microsoft, como o Microsoft Word 2016 e o Microsoft Exchange Server 2016. Para obter mais informações, consulte Trabalhando com linhas de base de patch personalizadas.

Diferença 5: opções de lista de patches rejeitados em listas de referência de patches personalizados

Ao criar uma lista de referência de patches personalizados, é possível especificar um ou mais patches para uma lista de Patches rejeitados. Para nós gerenciados do Linux, você também pode optar por permitir que eles sejam instalados se forem dependências de outro patch permitido pela lista de referência.

O Windows Server, no entanto, não é compatível com o conceito de dependências de patches. Você pode adicionar um patch à lista de Patches rejeitados em uma lista de referência personalizada para o Windows Server, mas o resultado depende de (1) se o patch rejeitado já está ou não instalado em um nó gerenciado e (2) de qual opção você escolhe para a ação de Patches rejeitados.

Consulte a tabela a seguir para obter detalhes sobre as opções de patch rejeitadas no Windows Server.

Status da instalação	Opção: "Permitir como dependência"	Opção: "Bloquear"
O patch já está instalado	Status relatado: `INSTALLED_OTHER`	Status relatado: `INSTALLED_REJECTED`
O patch ainda não está instalado	Patch ignorado	Patch ignorado

Cada patch para Windows Server que a Microsoft lança normalmente contém todas as informações necessárias para que a instalação seja bem-sucedida. Ocasionalmente, no entanto, pode ser necessário um pacote de pré-requisitos, que você deve instalar manualmente. O Patch Manager não relata informações sobre esses pré-requisitos. Para obter informações relacionadas, consulte a solução de problemas do Windows Update no site da Microsoft.

Documentos do comando do SSM para aplicação de patches em nós gerenciados

Este tópico descreve os sete documentos do Systems Manager (documentos SSM) atualmente disponíveis para ajudar você a manter seus nós gerenciados protegidos com as mais recentes atualizações relacionadas à segurança.

Recomendamos usar apenas cinco desses documentos em suas operações de aplicação de patches. Juntos, esses cinco documentos do SSM fornecem uma variedade completa de opções de aplicação de patches usando o AWS Systems Manager. Quatro desses documentos foram lançados após os quatro documentos do SSM legados para substituí-los e representam expansões ou consolidações de funcionalidade.

Documentos do SSM recomendados para aplicação de patches

Recomendamos usar os cinco documentos do SSM a seguir em suas operações de aplicação de patches.

- `AWS-ConfigureWindowsUpdate`
- `AWS-InstallWindowsUpdates`
- `AWS-RunPatchBaseline`

- `AWS-RunPatchBaselineAssociation`
- `AWS-RunPatchBaselineWithHooks`

Documentos do SSM legados para aplicação de patches

Os quatro documentos do SSM legados a seguir ainda estão disponíveis para uso em algumas Regiões da AWS, mas não estão mais atualizados, não é garantido que funcionem em todos os cenários e talvez não tenham mais suporte no futuro. Recomendamos não utilizá-los em suas operações de patch.

- `AWS-ApplyPatchBaseline`
- `AWS-FindWindowsUpdates`
- `AWS-InstallMissingWindowsUpdates`
- `AWS-InstallSpecificWindowsUpdates`

Consulte as seções a seguir para obter mais informações sobre como usar esses documentos do SSM em suas operações de aplicação de patches.

Tópicos

- Documentos do SSM recomendados para aplicação de patches em nós gerenciados
- Documentos do SSM para aplicação de patches em nós gerenciados
- Documento de comando do SSM para a aplicação de patches: AWS-RunPatchBaseline
- Documento de comando do SSM para a aplicação de patches: AWS-RunPatchBaselineAssociation
- Documento de comando do SSM para a aplicação de patches: AWS-RunPatchBaselineWithHooks
- Cenário de exemplo do uso do parâmetro InstallOverrideList em AWS-RunPatchBaseline ou AWS-RunPatchBaselineAssociation
- Usar o parâmetro BaselineOverride

Documentos do SSM recomendados para aplicação de patches em nós gerenciados

Os cinco documentos do SSM a seguir são recomendados para uso nas operações de aplicação de patches em nós gerenciados.

Documentos do SSM recomendados

- AWS-ConfigureWindowsUpdate
- AWS-InstallWindowsUpdates
- AWS-RunPatchBaseline
- AWS-RunPatchBaselineAssociation
- AWS-RunPatchBaselineWithHooks

`AWS-ConfigureWindowsUpdate`

Oferece suporte à configuração e uso de funções básicas do Windows Update para instalar atualizações automaticamente (ou desativar atualizações automáticas). Disponível em todas as Regiões da AWS

Esse documento do SSM configura o Windows Update para baixar e instalar as atualizações especificadas e reiniciar os nós gerenciados conforme necessário. Use esse documento com o State Manager, uma ferramenta do AWS Systems Manager, para garantir que o Windows Update mantenha sua configuração. Você também pode executá-lo manualmente usando o Run Command, uma ferramenta do AWS Systems Manager, para alterar a configuração do Windows Update.

Os parâmetros disponíveis neste documento oferecem suporte à especificação de uma categoria de atualizações a serem instaladas (ou se as atualizações automáticas devem ser desativadas), bem como especificar o dia da semana e a hora do dia para executar operações de aplicação de patches. Esse documento do SSM é mais útil se você não precisa de controles rigorosos sobre as atualizações do Windows e não precisa coletar informações de conformidade.

Substitui estes documentos do SSM legados:

- Nenhum

`AWS-InstallWindowsUpdates`

Instala atualizações em um nó gerenciado do Windows Server. Disponível em todas as Regiões da AWS

Esse documento do SSM fornece a funcionalidade básica de aplicação de patches nos casos em que você deseja instalar uma atualização específica (usando o parâmetro `Include Kbs`) ou deseja instalar patches com classificações ou categorias específicas, mas não precisa de informações de conformidade do patch.

Substitui estes documentos do SSM legados:

- `AWS-FindWindowsUpdates`
- `AWS-InstallMissingWindowsUpdates`
- `AWS-InstallSpecificWindowsUpdates`

Os três documentos legados executam funções diferentes, mas você pode alcançar os mesmos resultados usando diferentes configurações de parâmetro com o documento do SSM mais recente `AWS-InstallWindowsUpdates`. Essas configurações de parâmetro são descritas em Documentos do SSM para aplicação de patches em nós gerenciados.

`AWS-RunPatchBaseline`

Instala patches em nós gerenciados ou verifica os nós para determinar se qualquer patch qualificado está ausente. Disponível em todas as Regiões da AWS

O `AWS-RunPatchBaseline` permite controlar aprovações de patches usando a linha de base do patch especificada como “padrão” para um tipo de sistema operacional. Ele relata informações de conformidade de patch que você pode exibir usando as ferramentas de conformidade do Systems Manager. Essas ferramentas fornecem informações sobre o estado de conformidade de patch dos nós gerenciados, como quais nós têm patches ausentes e quais são esses patches. Quando você usa`AWS-RunPatchBaseline`, as informações de conformidade de patch são registradas usando o`PutInventory`Comando da API. Para sistemas operacionais Linux, as informações de conformidade são fornecidas para patches do repositório de origem padrão configurado em um nó gerenciado e de qualquer repositório de origem alternativo especificado em uma lista personalizada de referência de patches. Para obter mais informações sobre repositórios de origem alternativos, consulte Como especificar um repositório de origem de patches alternativo (Linux). Para obter mais informações sobre as ferramentas de conformidade do Systems Manager, consulte AWS Systems Manager Compliance.

Substitui estes documentos legados:

- `AWS-ApplyPatchBaseline`

O documento legado `AWS-ApplyPatchBaseline` se aplica apenas a nós gerenciados do Windows Server e não é compatível com a aplicação de patches em aplicações. O `AWS-RunPatchBaseline` mais recente fornece o mesmo suporte para sistemas Windows e Linux. A versão 2.0.834.0 ou posterior do SSM Agent é necessária para usar o documento `AWS-RunPatchBaseline`.

Para obter mais informações sobre o documento do SSM, `AWS-RunPatchBaseline`, consulte Documento de comando do SSM para a aplicação de patches: `AWS-RunPatchBaseline`.

`AWS-RunPatchBaselineAssociation`

Instala patches nas suas instâncias ou verifica as instâncias para determinar se qualquer patch qualificado está ausente. Disponível em todas as Regiões da AWS comerciais.

O `AWS-RunPatchBaselineAssociation` difere do `AWS-RunPatchBaseline` de algumas maneiras importantes:

- O `AWS-RunPatchBaselineAssociation` deve ser usado principalmente com associações do State Manager criadas usando o Quick Setup, uma ferramenta do AWS Systems Manager. Especificamente, quando você usar o tipo de configuração do Host Management do Quick Setup, se você escolher a opção Scan instances for missing patches daily (Verificar patches ausentes nas instâncias diariamente), o sistema usará `AWS-RunPatchBaselineAssociation` para a operação.

  Na maioria dos casos, no entanto, ao configurar suas próprias operações de patch, você deve escolher `AWS-RunPatchBaseline` ou `AWS-RunPatchBaselineWithHooks`, ao invés do `AWS-RunPatchBaselineAssociation`.

  Para obter mais informações, consulte os tópicos a seguir.

  - AWS Systems Manager Quick Setup
  - Documento de comando do SSM para a aplicação de patches: `AWS-RunPatchBaselineAssociation`
- O `AWS-RunPatchBaselineAssociation` suporta o uso de tags para identificar qual linha de base de patch usar com um conjunto de destinos, quando ele for executado.
- Para operações de patch que usam o `AWS-RunPatchBaselineAssociation`, os dados de conformidade de patches são compilados em termos de uma associação do State Manager. Os dados de conformidade de patches coletados quando `AWS-RunPatchBaselineAssociation` é executado são gravados usando a API `PutComplianceItems` em vez do comando `PutInventory`. Isso impede que os dados de conformidade que não estão associados a essa associação específica sejam substituídos.

  Para sistemas operacionais Linux, as informações de conformidade são fornecidas para patches do repositório de origem padrão configurado em uma instância e de qualquer repositório de origem alternativo especificado em uma linha de base de patch personalizada. Para obter mais

informações sobre repositórios de origem alternativos, consulte Como especificar um repositório de origem de patches alternativo (Linux). Para obter mais informações sobre as ferramentas de conformidade do Systems Manager, consulte AWS Systems Manager Compliance.

Substitui estes documentos legados:

- Nenhum

Para obter mais informações sobre o documento do SSM, `AWS-RunPatchBaselineAssociation`, consulte Documento de comando do SSM para a aplicação de patches: `AWS-RunPatchBaselineAssociation`.

`AWS-RunPatchBaselineWithHooks`

Instala patches em seus nós gerenciados ou verifica os nós para determinar se qualquer patch qualificado está ausente, com hooks opcionais que você pode usar para executar documentos do SSM em três pontos durante o ciclo de aplicação de patches. Disponível em todas as Regiões da AWS comerciais. Não compatível com o macOS.

O `AWS-RunPatchBaselineWithHooks` difere do `AWS-RunPatchBaseline` na operação `Install`.

O `AWS-RunPatchBaselineWithHooks` oferece suporte a hooks de ciclo de vida executados em pontos designados durante a aplicação de patches em nós gerenciados. Como as instalações de patches às vezes exigem nós gerenciados para reinicializar, a operação de aplicação patches é dividida em dois eventos, para um total de três ganchos que suportam funcionalidade personalizada. O primeiro hook é antes da operação `Install with NoReboot`. O segundo hook é depois da operação `Install with NoReboot`. O terceiro hook está disponível após a reinicialização do nó.

Substitui estes documentos legados:

- Nenhum

Para obter mais informações sobre o documento do SSM, `AWS-RunPatchBaselineWithHooks`, consulte Documento de comando do SSM para a aplicação de patches: `AWS-RunPatchBaselineWithHooks`.

Documentos do SSM para aplicação de patches em nós gerenciados

Os quatro documentos do SSM a seguir ainda estão disponíveis em algumas Regiões da AWS. No entanto, eles não estão sendo mais atualizados e podem não ter mais suporte no futuro. Por isso, não recomendamos seu uso. Em vez disso, use os documentos descritos em Documentos do SSM recomendados para aplicação de patches em nós gerenciados.

Documentos do SSM legados

- AWS-ApplyPatchBaseline
- AWS-FindWindowsUpdates
- AWS-InstallMissingWindowsUpdates
- AWS-InstallSpecificWindowsUpdates

`AWS-ApplyPatchBaseline`

Oferece suporte apenas aos nós gerenciados do Windows Server, mas não inclui suporte para a aplicação de patches em aplicações, como ocorre em seu substituto, o `AWS-RunPatchBaseline`. Não disponível em Regiões da AWS lançadas após agosto de 2017.

> **Note**
>
> O substituto deste documento do SSM, `AWS-RunPatchBaseline`, requer a versão 2.0.834.0 ou posterior do SSM Agent. Você pode usar o documento `AWS-UpdateSSMAgent` para atualizar os nós gerenciados para a versão mais recente do agente.

`AWS-FindWindowsUpdates`

Substituído por `AWS-InstallWindowsUpdates`, que pode realizar as mesmas ações. Não disponível em Regiões da AWS lançadas após abril de 2017.

Para obter o mesmo resultado que teria com esse documento do SSM legado, use a seguinte configuração de parâmetro com o documento substituto recomendado, `AWS-InstallWindowsUpdates`:

- `Action` = `Scan`
- `Allow Reboot` = `False`

AWS-InstallMissingWindowsUpdates

Substituído por `AWS-InstallWindowsUpdates`, que pode realizar as mesmas ações. Não disponível em nenhuma Regiões da AWS lançada após abril de 2017.

Para obter o mesmo resultado que teria com esse documento do SSM legado, use a seguinte configuração de parâmetro com o documento substituto recomendado, `AWS-InstallWindowsUpdates`:

- `Action` = `Install`
- `Allow Reboot` = `True`

AWS-InstallSpecificWindowsUpdates

Substituído por `AWS-InstallWindowsUpdates`, que pode realizar as mesmas ações. Não disponível em nenhuma Regiões da AWS lançada após abril de 2017.

Para obter o mesmo resultado que teria com esse documento do SSM legado, use a seguinte configuração de parâmetro com o documento substituto recomendado, `AWS-InstallWindowsUpdates`:

- `Action` = `Install`
- `Allow Reboot` = `True`
- `Include Kbs` = *`lista de artigos da base de dados de conhecimento separados por vírgula`*

Documento de comando do SSM para a aplicação de patches: **AWS-RunPatchBaseline**

O AWS Systems Manager é compatível com o `AWS-RunPatchBaseline`, um documento do Systems Manager (documento SSM) para o Patch Manager, uma ferramenta do AWS Systems Manager. Este documento do SSM executa operações de aplicação de patches em nós gerenciados para atualizações de segurança e outros tipos de atualizações. Quando o documento é executado, ele usa a linha de base do patch especificada como “padrão” para um tipo de sistema operacional se nenhum grupo de patches for especificado. Caso contrário, ele usa a linha de base do patch associada ao grupo de patches. Para obter mais informações sobre grupos de patches, consulte Grupos de patches.

Você pode usar o documento `AWS-RunPatchBaseline` para aplicar patches aos sistemas operacionais e aplicações. (No Windows Server, o suporte a aplicações é limitado a atualizações de aplicações da Microsoft.)

Este documento oferece suporte a nós gerenciados do Linux, macOS e Windows Server. O documento executará as ações adequadas a cada plataforma.

> **Note**
>
> O Patch Manager também oferece suporte ao documento SSM legado, `AWS-ApplyPatchBaseline`. No entanto, esse documento comporta a aplicação de patches somente em nós gerenciados do Windows. Em vez disso, é recomendável usar o `AWS-RunPatchBaseline` por ser compatível com a aplicação de patches em nós gerenciados do Linux, do macOS e do Windows Server. A versão 2.0.834.0 ou posterior do SSM Agent é necessária para usar o documento `AWS-RunPatchBaseline`.

Windows Server

Em nós gerenciados do Windows Server, o documento `AWS-RunPatchBaseline` baixa e invoca um módulo do PowerShell, que, por sua vez, baixa um snapshot da lista de referência de patches que se aplica ao nó gerenciado. Esse snapshot da lista de referência de patches contém uma lista de patches aprovados que é compilada consultando a lista de referência de patches em um servidor Windows Server Update Service (WSUS). Esse snapshot da lista de referência de patches é passado para a API do Windows Update, que controla o download e a instalação de patches aprovados, conforme apropriado.

Linux

Em nós gerenciados do Linux, o documento `AWS-RunPatchBaseline` invoca um módulo do Python, que, por sua vez, baixa um snapshot da lista de referência de patches que se aplica ao nó gerenciado. Esse snapshot da lista de referência de patches usa regras e listas de patches aprovados e bloqueados para conduzir o gerenciador de pacotes apropriado para cada tipo de nó:

- Nós gerenciados do Amazon Linux 1, Amazon Linux 2, CentOS, Oracle Linux e RHEL 7 usam YUM. Para operações YUM, o Patch Manager requer o `Python 2.6` ou uma versão posterior compatível (2.6-3.10).

- Os nós gerenciados do RHEL 8 usam o DNF. Para operações DNF, o Patch Manager requer uma versão compatível do `Python 2` ou `Python 3` (2.6-3.10). (Nenhuma versão é instalada por padrão no RHEL 8. É necessário instalar um deles manualmente.)
- As instâncias do Debian Server, do Raspberry Pi OS e do Ubuntu Server usam o APT. Para operações APT, o Patch Manager requer uma versão compatível do `Python 3` (3.0-3.10).
- Os nós gerenciados do SUSE Linux Enterprise Server usam o Zypper. Para operações Zypper, o Patch Manager requer o `Python 2.6` ou uma versão posterior compatível (2.6-3.10).

macOS

Em nós gerenciados do macOS, o documento `AWS-RunPatchBaseline` invoca um módulo do Python, que, por sua vez, baixa um snapshot da lista de referência de patches que se aplica ao nó gerenciado. Em seguida, um subprocesso do Python invoca o AWS Command Line Interface (AWS CLI) em seu nó para recuperar as informações de instalação e atualização para os gerenciadores de pacotes especificados e para direcionar o gerenciador de pacotes apropriado para cada pacote de atualização.

Cada snapshot é específico para uma Conta da AWS, um grupo de patches, um sistema operacional e um ID de snapshot. O snapshot é entregue por meio de um URL pré-assinado do Amazon Simple Storage Service (Amazon S3), que expira 24 horas após a criação do snapshot. No entanto, se você quiser aplicar o mesmo conteúdo de snapshot a outros nós gerenciados depois que o URL expirar, você poderá gerar um novo URL pré-assinado do Amazon S3 até três dias após a criação do snapshot. Para fazer isso, use o comando get-deployable-patch-snapshot-for-instance.

Depois que todas as atualizações aprovadas e aplicáveis forem instaladas, e as reinicializações realizadas conforme a necessidade, as informações de conformidade do patch serão geradas em um nó gerenciado e relatadas ao Patch Manager.

> **Note**
>
> Se o parâmetro `RebootOption` estiver definido como `NoReboot` no documento `AWS-RunPatchBaseline`, o nó gerenciado não será reinicializado após a execução do Patch Manager. Para obter mais informações, consulte Nome do parâmetro: `RebootOption`.

Para obter informações sobre como visualizar dados de conformidade de patches, consulte Sobre a conformidade de patches.

Parâmetros do **AWS-RunPatchBaseline**

O AWS-RunPatchBaseline oferece suporte a seis parâmetros. O parâmetro Operation é obrigatório. Os parâmetros InstallOverrideList, BaselineOverride e RebootOption são opcionais. O Snapshot-ID é tecnicamente opcional, mas recomendamos que você forneça um valor personalizado para ele quando executar o AWS-RunPatchBaseline fora de uma janela de manutenção. O Patch Manager pode fornecer o valor personalizado automaticamente quando o documento for executado como parte de uma operação da janela de manutenção.

Parâmetros

- Nome do parâmetro: Operation
- Nome do parâmetro: AssociationId
- Nome do parâmetro: Snapshot ID
- Nome do parâmetro: InstallOverrideList
- Nome do parâmetro: RebootOption
- Nome do parâmetro: BaselineOverride
- Nome do parâmetro: StepTimeoutSeconds

Nome do parâmetro: **Operation**

Uso: obrigatório.

Opções: Scan | Install.

Verificar

Quando você escolhe a opção Scan, o AWS-RunPatchBaseline determina o estado de conformidade dos patches do nó gerenciado e retorna essas informações para o Patch Manager. A opção Scan não solicita que atualizações sejam instaladas ou que nós gerenciados sejam reinicializados. Em vez disso, a operação identifica onde existem atualizações ausentes que estão aprovadas e são aplicáveis ao nó.

Instalar

Quando você escolhe a opção Install, o AWS-RunPatchBaseline tenta instalar as atualizações aprovadas e aplicáveis que estiverem ausentes em seu nó gerenciado. As informações de conformidade dos patches geradas como parte de uma operação Install não

listam atualizações ausentes, mas poderão indicar atualizações em estado malsucedido se, por qualquer motivo, a instalação da atualização não tiver tido êxito. Sempre que uma atualização é instalada em um nó gerenciado, o nó é reinicializado para garantir que a atualização esteja instalada e ativa. (Exceção: Se o parâmetro `RebootOption` estiver definido como `NoReboot` no documento `AWS-RunPatchBaseline`, o nó gerenciado não será reinicializado depois que o Patch Manager for executado. Para obter mais informações, consulte Nome do parâmetro: `RebootOption`).

> **Note**
>
> Se um patch especificado pelas regras da lista de referência for instalado antes que o Patch Manager atualize o nó gerenciado, o sistema poderá não ser reinicializado conforme esperado. Isso pode acontecer quando um patch é instalado manualmente por um usuário ou instalado automaticamente por outro programa, como o pacote `unattended-upgrades` no Ubuntu Server.

Nome do parâmetro: **`AssociationId`**

Uso: opcional.

O `AssociationId` é o ID de uma associação existente no State Manager, uma ferramenta do AWS Systems Manager. Ela é usada pelo Patch Manager para adicionar dados de conformidade à associação especificada. Essa associação está relacionada a uma operação de aplicação de patches configurada em uma política de patch no Quick Setup.

> **Note**
>
> Com o `AWS-RunPatchBaseline`, se um valor de `AssociationId` for fornecido junto com uma substituição da lista de referência da política de patch, a aplicação de patches será feita como uma operação `PatchPolicy` e o valor `ExecutionType` informado em `AWS:ComplianceItem` também será `PatchPolicy`. Se nenhum valor de `AssociationId` for fornecido, a aplicação de patches será feita como uma operação `Command` e a informação do valor `ExecutionType` no `AWS:ComplianceItem` enviado também será `Command`.

Se ainda não tiver uma associação que você deseja usar, crie uma associação executando o comando create-association.

Nome do parâmetro: **`Snapshot ID`**

Uso: opcional.

O `Snapshot ID` é um ID exclusivo (GUID) usado pelo Patch Manager para garantir que um conjunto de nós gerenciados, corrigidos em uma única operação, tenham o mesmo conjunto de patches aprovados. Embora o parâmetro seja definido como opcional, nossa recomendação baseada em práticas recomendadas depende de estar executando ou não o `AWS-RunPatchBaseline` em uma janela de manutenção, conforme descrito na tabela a seguir.

Práticas recomendadas do **`AWS-RunPatchBaseline`**

Modo	Prática recomendada	Detalhes
Running`AWS-RunPatchBaseline` Dentro de uma janela de manutenção	Não forneça um ID de snapshot. O Patch Manager fornecerá para você.	Se você usar uma janela de manutenção para executar o `AWS-RunPatchBaseline` , não forneça seu próprio ID de snapshot gerado. Nesse cenário, o Systems Manager fornece um valor GUID com base no ID de execução da janela de manutenção. Isso garante que seja usado um ID correto para todas as invocações do `AWS-RunPatchBaseline` nessa janela de manutenção. Se você especificar um valor nesse cenário, observe que talvez o snapshot da lista de referência de patches não permaneça vigente por mais de três dias. Depois disso, um novo snapshot será gerado, mesmo que você especific

Modo	Prática recomendada	Detalhes
		ar o mesmo ID depois que o snapshot expirar.

Modo	Prática recomendada	Detalhes
Running`AWS-RunPatchBaseline` Fora de uma janela de manutenção	Gere e especifique um valor de GUID personalizado para o ID do snapshot..[1]	Quando você não estiver usando uma janela de manutenção para executar o `AWS-RunPatchBaseline`, recomendamos gerar e especificar um ID de snapshot exclusivo para cada lista de referência de patches, principalmente se estiver executando o documento `AWS-RunPatchBaseline` em vários nós gerenciados na mesma operação. Se você não especificar um ID nesse cenário, o Systems Manager gerará um ID de snapshot diferente para cada nó gerenciado para o qual o comando for enviado. Em consequência disso, vários conjuntos de patches podem ser especificados entre os nós gerenciados. Por exemplo, digamos que você esteja executando o documento `AWS-RunPatchBaseline` diretamente do Run Command, uma ferramenta do AWS Systems Manager, e visando um grupo de 50 nós gerenciados. Ao especificar os resultados de um ID de snapshot personalizado, na geração de um

Modo	Prática recomendada	Detalhes
		único snapshot da lista de referência, que é usado para avaliar e corrigir todos os nós, garantindo que eles adquiram um estado consistente.

¹ Você pode usar qualquer ferramenta capaz de gerar um GUID para gerar um valor para o parâmetro ID do snapshot. Por exemplo, no PowerShell, você pode usar o cmdlet `New-Guid` para gerar um GUID no formato `12345699-9405-4f69-bc5e-9315aEXAMPLE` .

Nome do parâmetro: **`InstallOverrideList`**

Uso: opcional.

Usando o `InstallOverrideList`, especifique um URL de estilo caminho do Amazon S3 para uma lista de patches a serem instalados. Esta lista de instalação de patches mantida no formato YAML substitui os patches especificados pela linha de base de patch padrão atual. Isso fornece a você mais controle granular sobre quais patches estão instalados em seus nós gerenciados.

> **Important**
>
> O nome do arquivo `InstallOverrideList` não pode conter os seguintes caracteres: crase (`), aspas simples ('), aspas duplas (") e cifrão ($).

O comportamento da operação de aplicação de patches ao usar o parâmetro `InstallOverrideList` difere entre os nós gerenciados pelo Linux e pelo macOS e os nós gerenciados pelo Windows Server. No Linux e no macOS, o Patch Manager tenta aplicar os patches incluídos na lista de patches `InstallOverrideList` que estão presentes em qualquer repositório habilitado no nó, independentemente de os patches corresponderem ou não às regras da lista de referência de patches. Em nós Windows Server, no entanto, os patches na lista de patches `InstallOverrideList` são aplicados somente se eles também correspondem às regras da lista de referência de patches.

Lembre-se de que os relatórios de conformidade de patch refletem estados de acordo com o que está especificado na linha de base de patch, e não o que você especifica em uma lista de

patches `InstallOverrideList`. Em outras palavras, operações de verificação ignoram o parâmetro `InstallOverrideList`. Isso é para garantir que os relatórios de conformidade reflitam consistentemente os estados de patch de acordo com a política, em vez de algo aprovado para uma determinada operação de patch.

Para obter uma descrição de como você pode usar o parâmetro `InstallOverrideList` para aplicar diferentes tipos de patches a um grupo de destino, em diferentes programações de janelas de manutenção, enquanto ainda usa uma única lista de referência de patches, consulte Cenário de exemplo do uso do parâmetro InstallOverrideList em `AWS-RunPatchBaseline` ou `AWS-RunPatchBaselineAssociation`.

Formatos URL válidos

> **Note**
>
> Se o arquivo estiver armazenado em um bucket disponível publicamente, você poderá especificar um formato de URL https ou um URL de estilo de caminho do Amazon S3. Se o arquivo estiver armazenado em um bucket privado, você deverá especificar um URL do estilo de caminho do Amazon S3.

- Formato URL em https:

```
https://s3.aws-api-domain/amzn-s3-demo-bucket/my-windows-override-list.yaml
```

- URL estilo de caminho do Amazon S3:

```
s3://amzn-s3-demo-bucket/my-windows-override-list.yaml
```

Formatos de conteúdo YAML válidos

Os formatos que você usa para especificar patches em sua lista depende do sistema operacional do nó gerenciado. O formato geral, no entanto, é o seguinte:

```
patches:
    -
        id: '{patch-d}'
        title: '{patch-title}'
```

```
        {additional-fields}:{values}
```

Embora você possa fornecer campos adicionais em seu arquivo YAML, eles serão ignorados durante operações de patch.

Além disso, é recomendável verificar se o formato do arquivo YAML é válido antes de adicionar ou atualizar a lista no seu bucket do S3. Para obter mais informações sobre o formato YAML, consulte yaml.org. Para as opções de ferramenta de validação, pesquise na Internet por "validadores de formato yaml".

Linux

id

O campo id é obrigatório. Use-o para especificar patches usando o nome do pacote e a arquitetura. Por exemplo: `'dhclient.x86_64'`. Você pode usar caracteres curinga no id para indicar vários pacotes. Por exemplo: `'dhcp*'` e `'dhcp*1.*'`.

Cargo

O campo título é opcional, mas em sistemas Linux ele fornece recursos de filtragem adicionais. Se você usar o título, ele deve conter informações sobre a versão do pacote em um dos seguintes formatos:

YUM/SUSE Linux Enterprise Server (SLES):

```
{name}.{architecture}:{epoch}:{version}-{release}
```

APT

```
{name}.{architecture}:{version}
```

Para títulos de patches do Linux, você pode usar um ou mais caracteres curinga em qualquer posição para expandir o número de correspondências de pacotes. Por exemplo: `'*32:9.8.2-0.*.rc1.57.amzn1'`.

Por exemplo:

- A versão do pacote apt 1.2.25 está atualmente instalada em seu nó gerenciado, mas a versão 1.2.27 agora está disponível.

- Você adiciona a versão apt.amd64 1.2.27 à lista de patches. Depende da versão do apt utils.amd64 1.2.27, mas a versão apt utils.amd64 1.2.25 é especificada na lista.

Neste caso, a versão apt 1.2.27 não poderá ser instalada e será relatada como "Failed-NonCompliant."

Windows Server

id

O campo id é obrigatório. Use-o para especificar os patches usando IDs da Base de Dados de Conhecimento Microsoft (por exemplo, KB2736693 e IDs do boletim de segurança da Microsoft (por exemplo, MS17-023).

Todos os outros campos que você deseja fornecer em uma lista de patches para Windows são opcionais e são apenas para seu próprio uso informativo. Você pode usar campos adicionais, como título, classificação, severidade, ou qualquer outra coisa para fornecer informações mais detalhadas sobre os patches especificados.

macOS

id

O campo id é obrigatório. O valor doidpode ser fornecido usando um campo`{package-name}.{package-version}`ou um formato {package_name}.

Exemplo de listas de patch

- Amazon Linux

```
patches:
    -
        id: 'kernel.x86_64'
    -
        id: 'bind*.x86_64'
        title: '32:9.8.2-0.62.rc1.57.amzn1'
    -
        id: 'glibc*'
    -
        id: 'dhclient*'
        title: '*12:4.1.1-53.P1.28.amzn1'
    -
```

```
        id: 'dhcp*'
        title: '*10:3.1.1-50.P1.26.amzn1'
```

- CentOS

```
patches:
    -
        id: 'kernel.x86_64'
    -
        id: 'bind*.x86_64'
        title: '32:9.8.2-0.62.rc1.57.amzn1'
    -
        id: 'glibc*'
    -
        id: 'dhclient*'
        title: '*12:4.1.1-53.P1.28.amzn1'
    -
        id: 'dhcp*'
        title: '*10:3.1.1-50.P1.26.amzn1'
```

- Debian Server

```
patches:
    -
        id: 'apparmor.amd64'
        title: '2.10.95-0ubuntu2.9'
    -
        id: 'cryptsetup.amd64'
        title: '*2:1.6.6-5ubuntu2.1'
    -
        id: 'cryptsetup-bin.*'
        title: '*2:1.6.6-5ubuntu2.1'
    -
        id: 'apt.amd64'
        title: '*1.2.27'
    -
        id: 'apt-utils.amd64'
        title: '*1.2.25'
```

- macOS

```
patches:
    -
```

```
        id: 'XProtectPlistConfigData'
    -
        id: 'MRTConfigData.1.61'
    -
        id: 'Command Line Tools for Xcode.11.5'
    -
        id: 'Gatekeeper Configuration Data'
```

- Oracle Linux

```
patches:
    -
        id: 'audit-libs.x86_64'
        title: '*2.8.5-4.el7'
    -
        id: 'curl.x86_64'
        title: '*.el7'
    -
        id: 'grub2.x86_64'
        title: 'grub2.x86_64:1:2.02-0.81.0.1.el7'
    -
        id: 'grub2.x86_64'
        title: 'grub2.x86_64:1:*-0.81.0.1.el7'
```

- Red Hat Enterprise Linux (RHEL)

```
patches:
    -
        id: 'NetworkManager.x86_64'
        title: '*1:1.10.2-14.el7_5'
    -
        id: 'NetworkManager-*.x86_64'
        title: '*1:1.10.2-14.el7_5'
    -
        id: 'audit.x86_64'
        title: '*0:2.8.1-3.el7'
    -
        id: 'dhclient.x86_64'
        title: '*.el7_5.1'
    -
        id: 'dhcp*.x86_64'
        title: '*12:5.2.5-68.el7'
```

- SUSE Linux Enterprise Server (SLES)

```
patches:
    -
        id: 'amazon-ssm-agent.x86_64'
    -
        id: 'binutils'
        title: '*0:2.26.1-9.12.1'
    -
        id: 'glibc*.x86_64'
        title: '*2.19*'
    -
        id: 'dhcp*'
        title: '0:4.3.3-9.1'
    -
        id: 'lib*'
```

- Ubuntu Server

```
patches:
    -
        id: 'apparmor.amd64'
        title: '2.10.95-0ubuntu2.9'
    -
        id: 'cryptsetup.amd64'
        title: '*2:1.6.6-5ubuntu2.1'
    -
        id: 'cryptsetup-bin.*'
        title: '*2:1.6.6-5ubuntu2.1'
    -
        id: 'apt.amd64'
        title: '*1.2.27'
    -
        id: 'apt-utils.amd64'
        title: '*1.2.25'
```

- Windows

```
patches:
    -
        id: 'KB4284819'
```

```
        title: '2018-06 Cumulative Update for Windows Server 2016 (1709) for x64-
based Systems (KB4284819)'
    -
        id: 'KB4284833'
    -
        id: 'KB4284835'
        title: '2018-06 Cumulative Update for Windows Server 2016 (1803) for x64-
based Systems (KB4284835)'
    -
        id: 'KB4284880'
    -
        id: 'KB4338814'
```

Nome do parâmetro: **`RebootOption`**

Uso: opcional.

Opções: `RebootIfNeeded` | `NoReboot`

Padrão: `RebootIfNeeded`

> **Warning**
>
> A opção padrão é `RebootIfNeeded`. Selecione a opção correta para o seu caso de uso. Por exemplo, se os nós gerenciados precisarem reinicializar imediatamente para concluir um processo de configuração, escolha `RebootIfNeeded`. Ou, se você precisar manter a disponibilidade do nó gerenciado até um horário de reinicialização agendado, escolha `NoReboot`.

> **Important**
>
> Não é recomendável usar o Patch Manager para aplicar patches em instâncias de cluster no Amazon EMR (antes chamado de Amazon Elastic MapReduce). Em particular, não selecione a opção `RebootIfNeeded` para o parâmetro `RebootOption`. (Essa opção está disponível nos documentos do SSM Command para aplicação de patches em `AWS-RunPatchBaseline`, `AWS-RunPatchBaselineAssociation` e `AWS-RunPatchBaselineWithHooks`.)
>
> Os comandos subjacentes para aplicação de patches usando o Patch Manager utilizam os comandos `yum` e `dnf`. Portanto, as operações resultam em incompatibilidades por

causa da forma como os pacotes são instalados. Para obter informações sobre os métodos preferenciais para atualizar o software nos clusters do Amazon EMR, consulte Using the default AMI for Amazon EMR no Guia de gerenciamento do Amazon EMR.

RebootIfNeeded

Quando você escolhe a opção `RebootIfNeeded`, o nó gerenciado é reinicializado em um dos seguintes casos:

- O Patch Manager instalou um ou mais patches.

  O Patch Manager não avalia se uma reinicialização é exigida pelo patch. O sistema é reinicializado mesmo que o patch não exija uma reinicialização.

- O Patch Manager detecta um ou mais patches com um status de `INSTALLED_PENDING_REBOOT` durante a operação `Install`.

  O status `INSTALLED_PENDING_REBOOT` pode significar que a opção `NoReboot` foi selecionada na última vez em que a operação `Install` foi executada ou que um patch foi instalado fora do Patch Manager na última vez que o nó gerenciado foi reinicializado.

A reinicialização de nós gerenciados nesses dois casos garante que os pacotes atualizados sejam liberados da memória e mantenham o comportamento da aplicação de patches e da reinicialização consistente em todos os sistemas operacionais.

NoReboot

Quando você escolhe a opção `NoReboot`, o Patch Manager não reinicializa um nó gerenciado, mesmo que tenha instalado patches durante a operação `Install`. Essa opção é útil se você souber que os nós gerenciados não exigem reinicialização após a aplicação de patches, ou se houver aplicações ou processos em execução em um nó que não devam ser interrompidos por uma reinicialização da operação de aplicação de patch. Também é útil quando você deseja mais controle sobre o tempo de reinicializações de nós gerenciados, como, por exemplo, usando uma janela de manutenção.

Se você escolher a opção `NoReboot` e um patch for instalado, o patch receberá um status de `InstalledPendingReboot`. O nó gerenciado em si, no entanto, é marcado como `Non-Compliant`. Depois que ocorre uma reinicialização e uma operação `Scan` é executada, o status do nó gerenciado é atualizado para `Compliant`.

> **Important**
>
> A opção `NoReboot` só impede reinicializações no nível do sistema operacional. Reinicializações no nível de serviço ainda podem ocorrer como parte do processo de aplicação de patches. Por exemplo, quando o Docker é atualizado, serviços dependentes, como o Amazon Elastic Container Service, podem ser reiniciados automaticamente mesmo com `NoReboot` habilitado. Se você tiver serviços críticos que não devem ser interrompidos, considere medidas adicionais, como remover temporariamente as instâncias do serviço ou programar a aplicação de patches durante as janelas de manutenção.

Arquivo de rastreamento de instalação de patches: para rastrear a instalação de patches, sobretudo os patches que foram instalados desde a última reinicialização do sistema, o Systems Manager mantém um arquivo em seu nó gerenciado.

> **Important**
>
> Não exclua nem modifique o arquivo de monitoramento. Se esse arquivo for excluído ou corrompido, o relatório de conformidade de patch do nó gerenciado será impreciso. Se isso acontecer, reinicialize o nó e execute uma operação de verificação de patch para restaurar o arquivo.

Esse arquivo de rastreamento é armazenado nos seguintes locais em seus nós gerenciados:

- Sistemas operacionais Linux:
  - `/var/log/amazon/ssm/patch-configuration/patch-states-configuration.json`
  - `/var/log/amazon/ssm/patch-configuration/patch-inventory-from-last-operation.json`
- Sistema operacional Windows Server:
  - `C:\ProgramData\Amazon\PatchBaselineOperations\State\PatchStatesConfiguration.json`
  - `C:\ProgramData\Amazon\PatchBaselineOperations\State\PatchInventoryFromLastOperation.json`

Nome do parâmetro: **`BaselineOverride`**

Uso: opcional.

Você pode definir preferências de patches no runtime usando o parâmetro `BaselineOverride`. Essa substituição de linha de base é mantida como um objeto JSON em um bucket do S3. Ele garante que as operações de patch usem as linhas de base fornecidas que correspondem ao sistema operacional host em vez de aplicar as regras da linha de base do patch padrão

> **Important**
>
> O nome do arquivo `BaselineOverride` não pode conter os seguintes caracteres: crase (`), aspas simples ('), aspas duplas (") e cifrão ($).

Para obter mais informações sobre como usar o parâmetro `BaselineOverride`, consulte Usar o parâmetro BaselineOverride.

Nome do parâmetro: **`StepTimeoutSeconds`**

Uso: opcional.

O tempo em segundos, entre 1 e 36.000 segundos (10 horas), para que um comando seja concluído antes de que seja considerado que ele falhou.

Documento de comando do SSM para a aplicação de patches: **`AWS-RunPatchBaselineAssociation`**

Como o`AWS-RunPatchBaseline`Documento,`AWS-RunPatchBaselineAssociation`O executa operações de aplicação de patches em instâncias para atualizações relacionadas à segurança e outros tipos de atualizações. Você pode usar o documento `AWS-RunPatchBaselineAssociation` para aplicar patches aos sistemas operacionais e aplicações.. (No Windows Server, o suporte a aplicações é limitado a atualizações de aplicações da Microsoft.)

Este documento oferece suporte a instâncias do Amazon Elastic Compute Cloud (Amazon EC2) para Linux, macOS, eWindows Server. Não há suporte para nós que não sejam do EC2 em um ambiente híbrido e multinuvem. O documento executará as ações adequadas a cada plataforma, invocando um módulo Python no Linux e no macOS e um módulo do PowerShell em instâncias do Windows.

O `AWS-RunPatchBaselineAssociation`, porém, difere do `AWS-RunPatchBaseline` das seguintes maneiras:

- O `AWS-RunPatchBaselineAssociation` deve ser usado principalmente com associações do State Manager criadas usando o Quick Setup, uma ferramenta do AWS Systems Manager. Especificamente, quando você usar o tipo de configuração do Host Management do Quick Setup, se você escolher a opção Scan instances for missing patches daily (Verificar patches ausentes nas instâncias diariamente), o sistema usará `AWS-RunPatchBaselineAssociation` para a operação.

  Na maioria dos casos, no entanto, ao configurar suas próprias operações de patch, você deve escolher `AWS-RunPatchBaseline` ou `AWS-RunPatchBaselineWithHooks`, ao invés do `AWS-RunPatchBaselineAssociation`.

- Quando você usa o`AWS-RunPatchBaselineAssociation`, você pode especificar um key pair de tag no`BaselineTags`campo de parâmetro. Se uma lista de referência de patches personalizada na Conta da AWS compartilha essas tags, o Patch Manager, uma ferramenta do AWS Systems Manager, usa essa lista marcada quando ela é executada nas instâncias de destino em vez da lista de referência de patches "padrão" especificada no momento para o tipo de sistema operacional.

  > **Note**
  >
  > Se você optar por usar `AWS-RunPatchBaselineAssociation` em operações de patch diferentes daquelas configuradas usando o Quick Setup e quiser usar o parâmetro `BaselineTags`, forneça algumas permissões adicionais para o perfil de instâncias do Amazon Elastic Compute Cloud (Amazon EC2). Para obter mais informações, consulte Nome do parâmetro: `BaselineTags`.

  Ambos os formatos a seguir são válidos para o parâmetro `BaselineTags`:

  ```
  Key=tag-key,Values=tag-value
  ```

  ```
  Key=tag-key,Values=tag-value1,tag-value2,tag-value3
  ```

  > **Important**
  >
  > As chaves e os valores das chaves não podem conter os seguintes caracteres: crase (`), aspas simples ('), aspas duplas (") e cifrão ($).

- Quando `AWS-RunPatchBaselineAssociation` é executado, os dados de conformidade de patches coletados são registrados usando o comando da API `PutComplianceItems` em vez do comando `PutInventory`, que é usado pelo `AWS-RunPatchBaseline`. Essa diferença significa que as informações de conformidade de patch que são armazenadas e relatadas por uma Associação. Os dados de conformidade de patch gerados fora dessa associação não são substituídos.
- As informações de conformidade de patch relatadas após a execução de `AWS-RunPatchBaselineAssociation` indica se uma instância está ou não em conformidade. Ele não inclui detalhes em nível de patch, como demonstrado pela saída do comando da AWS Command Line Interface (AWS CLI) a seguir. Os filtros de comando em `Association` como o tipo de conformidade:

```
aws ssm list-compliance-items \
    --resource-ids "i-02573cafcfEXAMPLE" \
    --resource-types "ManagedInstance" \
    --filters "Key=ComplianceType,Values=Association,Type=EQUAL" \
    --region us-east-2
```

O sistema retorna informações como estas.

```
{
    "ComplianceItems": [
        {
            "Status": "NON_COMPLIANT",
            "Severity": "UNSPECIFIED",
            "Title": "MyPatchAssociation",
            "ResourceType": "ManagedInstance",
            "ResourceId": "i-02573cafcfEXAMPLE",
            "ComplianceType": "Association",
            "Details": {
                "DocumentName": "AWS-RunPatchBaselineAssociation",
                "PatchBaselineId": "pb-0c10e65780EXAMPLE",
                "DocumentVersion": "1"
            },
            "ExecutionSummary": {
                "ExecutionTime": 1590698771.0
            },
            "Id": "3e5d5694-cd07-40f0-bbea-040e6EXAMPLE"
        }
    ]
```

```
}
```

Se um valor de par de chaves de etiqueta tiver sido especificado como um parâmetro para o documento `AWS-RunPatchBaselineAssociation`, o Patch Manager pesquisa uma lista de referência de patches personalizada que corresponda ao tipo de sistema operacional e que tenha sido marcada com esse mesmo par de chaves de marca. Essa pesquisa não se limita à linha de base do patch padrão especificada atual ou à linha de base atribuída a um grupo de patches. Se nenhuma lista de referência for encontrada com as etiquetas especificadas, o Patch Manager procura por um grupo de patches, se um tiver sido especificado no comando que executa `AWS-RunPatchBaselineAssociation`. Se não houver correspondência com nenhum grupo de patches, o Patch Manager retorna à lista de referência de patches padrão atual da conta do sistema operacional.

Se mais de uma lista de referência de patches for encontrada com as tags especificadas no documento `AWS-RunPatchBaselineAssociation`, o Patch Manager retorna uma mensagem de erro indicando que apenas uma lista de referência de patches pode ser marcada com esse par de chave-valor para que a operação prossiga.

> **Note**
>
> Em instâncias Linux, o gerenciador de pacotes apropriado para cada tipo de instância é usado para instalar pacotes:
>
> - Instâncias do Amazon Linux 1, Amazon Linux 2, CentOS, Oracle Linux e RHEL usam YUM. Para operações YUM, o Patch Manager requer o `Python 2.6` ou uma versão posterior compatível (2.6-3.10).
> - As instâncias do Debian Server, do Raspberry Pi OS e do Ubuntu Server usam o APT. Para operações APT, o Patch Manager requer uma versão compatível do `Python 3` (3.0-3.10).
> - As instâncias do SUSE Linux Enterprise Server usam o Zypper. Para operações Zypper, o Patch Manager requer o `Python 2.6` ou uma versão posterior compatível (2.6-3.10).

Depois que todas as atualizações aprovadas e aplicáveis forem instaladas, e as reinicializações realizadas conforme a necessidade, as informações de conformidade do patch serão geradas em uma instância e relatadas ao Patch Compliance Service.

> **Note**
>
> Se o parâmetro `RebootOption` estiver definido como `NoReboot` no documento `AWS-RunPatchBaselineAssociation`, a instância não será reinicializada após a execução do Patch Manager. Para obter mais informações, consulte Nome do parâmetro: `RebootOption`.

Para obter informações sobre como visualizar dados de conformidade de patches, consulte Sobre a conformidade de patches.

Parâmetros do **`AWS-RunPatchBaselineAssociation`**

O `AWS-RunPatchBaselineAssociation` oferece suporte a cinco parâmetros. Os parâmetros `Operation` e `AssociationId` são obrigatórios. Os parâmetros `InstallOverrideList`, `RebootOption` e `BaselineTags` são opcionais.

Parâmetros

- Nome do parâmetro: Operation
- Nome do parâmetro: BaselineTags
- Nome do parâmetro: AssociationId
- Nome do parâmetro: InstallOverrideList
- Nome do parâmetro: RebootOption

Nome do parâmetro: **`Operation`**

Uso: obrigatório.

Opções: `Scan` | `Install`.

Verificar

Quando você escolhe a opção `Scan`, o `AWS-RunPatchBaselineAssociation` determina o estado de conformidade dos patches da instância e retorna essas informações para o Patch Manager. A opção `Scan` não solicita que atualizações sejam instaladas ou que instâncias sejam reinicializadas. Em vez disso, a operação identifica onde existem atualizações ausentes que estão aprovadas e são aplicáveis à instância.

Instalar

Quando você escolhe a opção `Install`, o `AWS-RunPatchBaselineAssociation` tenta instalar as atualizações aprovadas e aplicáveis que estiverem ausentes na instância. As informações de conformidade dos patches geradas como parte de uma operação `Install` não listam atualizações ausentes, mas poderão indicar atualizações em estado malsucedido se, por qualquer motivo, a instalação da atualização não tiver tido êxito. Sempre que uma atualização é instalada em uma instância, a instância é reinicializada para garantir que a atualização está instalada e ativa. (Exceção: Se o parâmetro `RebootOption` estiver definido como `NoReboot` no `AWS-RunPatchBaselineAssociation`, a instância não será reinicializada depis que o Patch Manager for executado. Para obter mais informações, consulte Nome do parâmetro: `RebootOption`).

> **Note**
>
> Se um patch especificado pelas regras da lista de referência for instalado antes que o Patch Manager atualize a instância, o sistema poderá não ser reinicializado conforme esperado. Isso pode acontecer quando um patch é instalado manualmente por um usuário ou instalado automaticamente por outro programa, como o pacote `unattended-upgrades` no Ubuntu Server.

Nome do parâmetro: **`BaselineTags`**

Uso: opcional.

O `BaselineTags` é um par de chaves/valores de tags exclusivo que você escolhe e atribui a uma linha de base de patch personalizada individual. Você pode especificar um ou mais valores para esse parâmetro. Ambos os formatos a seguir são válidos:

`Key=`*`tag-key`*`,Values=`*`tag-value`*

`Key=`*`tag-key`*`,Values=`*`tag-value1`*`,`*`tag-value2`*`,`*`tag-value3`*

> **Important**
>
> As chaves e os valores das chaves não podem conter os seguintes caracteres: crase (`), aspas simples ('), aspas duplas (") e cifrão ($).

O valor de `BaselineTags` é usado pelo Patch Manager para garantir que todas as instâncias de um conjunto que são corrigidas em uma única operação tenham o mesmo conjunto de patches aprovados. Quando a operação de patch é executada, o Patch Manager verifica se uma linha de base de patch para o tipo de sistema operacional está marcada com o mesmo par chave-valor especificado para `BaselineTags`. Se houver uma correspondência, essa lista de referência do patch personalizada será usada. Se não houver uma correspondência, uma linha de base de patch será identificada de acordo com qualquer grupo de patches especificado para a operação de patch. Se não houver nenhum, a linha de base de patch predefinida gerenciada da AWS para esse sistema operacional é usada.

Requisitos de permissão adicionais

Se você usar o `AWS-RunPatchBaselineAssociation` em operações de patch diferentes daquelas configuradas usando o Quick Setup, e quiser usar o parâmetro `BaselineTags`, adicione as permissões a seguir ao perfil para instâncias do Amazon Elastic Compute Cloud (Amazon EC2).

> **Note**
>
> O Quick Setup e `AWS-RunPatchBaselineAssociation` não oferecem suporte a servidores on-premises e máquinas virtuais (VMs).

```
{
    "Effect": "Allow",
    "Action": [
        "ssm:DescribePatchBaselines",
        "tag:GetResources"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "ssm:GetPatchBaseline",
        "ssm:DescribeEffectivePatchesForPatchBaseline"
    ],
    "Resource": "patch-baseline-arn"
}
```

Substitua *patch-baseline-arn* pelo nome do recurso da Amazon (ARN) da linha de referência de patches à qual você deseja fornecer acesso no formato`arn:aws:ssm:us-east-2:123456789012:patchbaseline/pb-0c10e65780EXAMPLE`.

Nome do parâmetro: **`AssociationId`**

Uso: obrigatório.

O `AssociationId` é o ID de uma associação existente no State Manager, uma ferramenta do AWS Systems Manager. Isso é usado pelo Patch Manager para adicionar dados de conformidade a uma associação especificada. Essa associação está relacionada a uma operação de `Scan` de patch habilitada em uma configuração do Host Management criada no Quick Setup. Ao enviar resultados da aplicação de patch como dados de conformidade de associação em vez de dados de conformidade do inventário, as informações de conformidade do inventário existentes para as instâncias não são substituídas após uma operação de aplicação de patch, nem em outras IDs de associação. Se ainda não tiver uma associação que você deseja usar, crie uma associação executando o comando create-association. Por exemplo:

Linux & macOS

```
aws ssm create-association \
    --name "AWS-RunPatchBaselineAssociation" \
    --association-name "MyPatchHostConfigAssociation" \
    --targets
 "Key=instanceids,Values=[i-02573cafcfEXAMPLE,i-07782c72faEXAMPLE,i-07782c72faEXAMPLE]"
 \
    --parameters "Operation=Scan" \
    --schedule-expression "cron(0 */30 * * * ? *)" \
    --sync-compliance "MANUAL" \
    --region us-east-2
```

Windows Server

```
aws ssm create-association ^
    --name "AWS-RunPatchBaselineAssociation" ^
    --association-name "MyPatchHostConfigAssociation" ^
    --targets
 "Key=instanceids,Values=[i-02573cafcfEXAMPLE,i-07782c72faEXAMPLE,i-07782c72faEXAMPLE]"
 ^
    --parameters "Operation=Scan" ^
    --schedule-expression "cron(0 */30 * * * ? *)" ^
```

```
    --sync-compliance "MANUAL" ^
    --region us-east-2
```

Nome do parâmetro: **`InstallOverrideList`**

Uso: opcional.

O uso do`InstallOverrideList`, você especifica um URL https ou um URL de estilo caminho do Amazon Simple Storage Service (Amazon S3) para uma lista de patches a serem instalados. Esta lista de instalação de patches mantida no formato YAML substitui os patches especificados pela linha de base de patch padrão atual. Isso fornece a você mais controle granular sobre quais patches estão instalados em suas instâncias.

> **Important**
>
> O nome do arquivo `InstallOverrideList` não pode conter os seguintes caracteres: crase (`), aspas simples ('), aspas duplas (") e cifrão ($).

O comportamento da operação de aplicação de patches ao usar o parâmetro `InstallOverrideList` difere entre os nós gerenciados pelo Linux e pelo macOS e os nós gerenciados pelo Windows Server. No Linux e no macOS, o Patch Manager tenta aplicar os patches incluídos na lista de patches `InstallOverrideList` que estão presentes em qualquer repositório habilitado no nó, independentemente de os patches corresponderem ou não às regras da lista de referência de patches. Em nós Windows Server, no entanto, os patches na lista de patches `InstallOverrideList` são aplicados somente se eles também correspondem às regras da lista de referência de patches.

Lembre-se de que os relatórios de conformidade de patch refletem estados de acordo com o que está especificado na linha de base de patch, e não o que você especifica em uma lista de patches `InstallOverrideList`. Em outras palavras, operações de verificação ignoram o parâmetro `InstallOverrideList`. Isso é para garantir que os relatórios de conformidade reflitam consistentemente os estados de patch de acordo com a política, em vez de algo aprovado para uma determinada operação de patch.

Formatos URL válidos

> **Note**
>
> Se o arquivo estiver armazenado em um bucket disponível publicamente, você poderá especificar um formato de URL https ou um URL de estilo de caminho do Amazon S3. Se o arquivo estiver armazenado em um bucket privado, você deverá especificar um URL do estilo de caminho do Amazon S3.

- Exemplo de formato de URL https:

```
https://s3.amazonaws.com/amzn-s3-demo-bucket/my-windows-override-list.yaml
```

- Exemplo de URL estilo de caminho do Amazon S3:

```
s3://amzn-s3-demo-bucket/my-windows-override-list.yaml
```

Formatos de conteúdo YAML válidos

Os formatos que você usa para especificar patches em sua lista depende do sistema operacional da instância. O formato geral, no entanto, é o seguinte:

```
patches:
    -
        id: '{patch-d}'
        title: '{patch-title}'
        {additional-fields}:{values}
```

Embora você possa fornecer campos adicionais em seu arquivo YAML, eles serão ignorados durante operações de patch.

Além disso, é recomendável verificar se o formato do arquivo YAML é válido antes de adicionar ou atualizar a lista no seu bucket do S3. Para obter mais informações sobre o formato YAML, consulte yaml.org. Para as opções de ferramenta de validação, pesquise na Internet por "validadores de formato yaml".

- Microsoft Windows

  id

O campo id é obrigatório. Use-o para especificar os patches usando IDs da Base de Dados de Conhecimento Microsoft (por exemplo, KB2736693 e IDs do boletim de segurança da Microsoft (por exemplo, MS17-023).

Todos os outros campos que você deseja fornecer em uma lista de patches para Windows são opcionais e são apenas para seu próprio uso informativo. Você pode usar campos adicionais, como título, classificação, severidade, ou qualquer outra coisa para fornecer informações mais detalhadas sobre os patches especificados.

- Linux

  id

  O campo id é obrigatório. Use-o para especificar patches usando o nome do pacote e a arquitetura. Por exemplo: `'dhclient.x86_64'`. Você pode usar caracteres curinga no id para indicar vários pacotes. Por exemplo: `'dhcp*'` e `'dhcp*1.*'`.

  título

  O campo título é opcional, mas em sistemas Linux ele fornece recursos de filtragem adicionais. Se você usar o título, ele deve conter informações sobre a versão do pacote em um dos seguintes formatos:

  YUM/SUSE Linux Enterprise Server (SLES):

  ```
  {name}.{architecture}:{epoch}:{version}-{release}
  ```

  APT

  ```
  {name}.{architecture}:{version}
  ```

  Para títulos de patches do Linux, você pode usar um ou mais caracteres curinga em qualquer posição para expandir o número de correspondências de pacotes. Por exemplo: `'*32:9.8.2-0.*.rc1.57.amzn1'`.

  Por exemplo:

  - A versão do pacote apt 1.2.25 está atualmente instalada em sua instância, mas a versão 1.2.27 agora está disponível.

- Você adiciona a versão apt.amd64 1.2.27 à lista de patches. Depende da versão do apt utils.amd64 1.2.27, mas a versão apt utils.amd64 1.2.25 é especificada na lista.

Neste caso, a versão apt 1.2.27 não poderá ser instalada e será relatada como "Failed-NonCompliant."

Outros campos

Todos os outros campos que você deseja fornecer em uma lista de patches para Linux são opcionais e são apenas para seu próprio uso informativo. Você pode usar campos adicionais, como classificação, severidade, ou qualquer outra coisa para fornecer informações mais detalhadas sobre os patches especificados.

Exemplo de listas de patch

- Windows

```
patches:
    -
        id: 'KB4284819'
        title: '2018-06 Cumulative Update for Windows Server 2016 (1709) for x64-
based Systems (KB4284819)'
    -
        id: 'KB4284833'
    -
        id: 'KB4284835'
        title: '2018-06 Cumulative Update for Windows Server 2016 (1803) for x64-
based Systems (KB4284835)'
    -
        id: 'KB4284880'
    -
        id: 'KB4338814'
```

- APT

```
patches:
    -
        id: 'apparmor.amd64'
        title: '2.10.95-0ubuntu2.9'
    -
        id: 'cryptsetup.amd64'
        title: '*2:1.6.6-5ubuntu2.1'
```

```
    -
        id: 'cryptsetup-bin.*'
        title: '*2:1.6.6-5ubuntu2.1'
    -
        id: 'apt.amd64'
        title: '*1.2.27'
    -
        id: 'apt-utils.amd64'
        title: '*1.2.25'
```

- Amazon Linux

```
patches:
    -
        id: 'kernel.x86_64'
    -
        id: 'bind*.x86_64'
        title: '32:9.8.2-0.62.rc1.57.amzn1'
    -
        id: 'glibc*'
    -
        id: 'dhclient*'
        title: '*12:4.1.1-53.P1.28.amzn1'
    -
        id: 'dhcp*'
        title: '*10:3.1.1-50.P1.26.amzn1'
```

- Red Hat Enterprise Linux (RHEL)

```
patches:
    -
        id: 'NetworkManager.x86_64'
        title: '*1:1.10.2-14.el7_5'
    -
        id: 'NetworkManager-*.x86_64'
        title: '*1:1.10.2-14.el7_5'
    -
        id: 'audit.x86_64'
        title: '*0:2.8.1-3.el7'
    -
        id: 'dhclient.x86_64'
        title: '*.el7_5.1'
    -
```

```
        id: 'dhcp*.x86_64'
        title: '*12:5.2.5-68.el7'
```

- SUSE Linux Enterprise Server (SLES)

```
patches:
    -
        id: 'amazon-ssm-agent.x86_64'
    -
        id: 'binutils'
        title: '*0:2.26.1-9.12.1'
    -
        id: 'glibc*.x86_64'
        title: '*2.19*'
    -
        id: 'dhcp*'
        title: '0:4.3.3-9.1'
    -
        id: 'lib*'
```

- Ubuntu Server

```
patches:
    -
        id: 'apparmor.amd64'
        title: '2.10.95-0ubuntu2.9'
    -
        id: 'cryptsetup.amd64'
        title: '*2:1.6.6-5ubuntu2.1'
    -
        id: 'cryptsetup-bin.*'
        title: '*2:1.6.6-5ubuntu2.1'
    -
        id: 'apt.amd64'
        title: '*1.2.27'
    -
        id: 'apt-utils.amd64'
        title: '*1.2.25'
```

- Windows

```
patches:
    -
        id: 'KB4284819'
        title: '2018-06 Cumulative Update for Windows Server 2016 (1709) for x64-
based Systems (KB4284819)'
    -
        id: 'KB4284833'
    -
        id: 'KB4284835'
        title: '2018-06 Cumulative Update for Windows Server 2016 (1803) for x64-
based Systems (KB4284835)'
    -
        id: 'KB4284880'
    -
        id: 'KB4338814'
```

Nome do parâmetro: **`RebootOption`**

Uso: opcional.

Opções: `RebootIfNeeded` | `NoReboot`

Padrão: `RebootIfNeeded`

> **Warning**
>
> A opção padrão é `RebootIfNeeded`. Selecione a opção correta para o seu caso de uso. Por exemplo, se as instâncias precisarem reinicializar imediatamente para concluir um processo de configuração, escolha `RebootIfNeeded`. Ou, se você precisar manter a disponibilidade das instâncias até um horário de reinicialização agendado, escolha `NoReboot`.

> **Important**
>
> A opção `NoReboot` só impede reinicializações no nível do sistema operacional. Reinicializações no nível de serviço ainda podem ocorrer como parte do processo de aplicação de patches. Por exemplo, quando o Docker é atualizado, serviços dependentes, como o Amazon Elastic Container Service, podem ser reiniciados automaticamente mesmo

com `NoReboot` habilitado. Se você tiver serviços críticos que não devem ser interrompidos, considere medidas adicionais, como remover temporariamente as instâncias do serviço ou programar a aplicação de patches durante as janelas de manutenção.

Important

Não é recomendável usar o Patch Manager para aplicar patches em instâncias de cluster no Amazon EMR (antes chamado de Amazon Elastic MapReduce). Em particular, não selecione a opção `RebootIfNeeded` para o parâmetro `RebootOption`. (Essa opção está disponível nos documentos do SSM Command para aplicação de patches em `AWS-RunPatchBaseline`, `AWS-RunPatchBaselineAssociation` e `AWS-RunPatchBaselineWithHooks`.)
Os comandos subjacentes para aplicação de patches usando o Patch Manager utilizam os comandos `yum` e `dnf`. Portanto, as operações resultam em incompatibilidades por causa da forma como os pacotes são instalados. Para obter informações sobre os métodos preferenciais para atualizar o software nos clusters do Amazon EMR, consulte Using the default AMI for Amazon EMR no Guia de gerenciamento do Amazon EMR.

RebootIfNeeded

Quando você escolhe a opção `RebootIfNeeded`, a instância é reinicializada em um dos seguintes casos:

- O Patch Manager instalou um ou mais patches.

  O Patch Manager não avalia se uma reinicialização é exigida pelo patch. O sistema é reinicializado mesmo que o patch não exija uma reinicialização.
- O Patch Manager detecta um ou mais patches com um status de `INSTALLED_PENDING_REBOOT` durante a operação `Install`.

  O status `INSTALLED_PENDING_REBOOT` pode significar que a opção `NoReboot` foi selecionada na última vez em que a operação `Install` foi executada ou que um patch foi instalado fora do Patch Manager na última vez que o nó gerenciado foi reinicializado.

A reinicialização de instâncias nesses dois casos garante que os pacotes atualizados sejam liberados da memória e mantém o comportamento de patch e reinicialização consistente em todos os sistemas operacionais.

NoReboot

Quando você escolhe a opção `NoReboot`, o Patch Manager não reinicializa uma instância, mesmo que tenha instalado patches durante a operação `Install`. Essa opção é útil se você souber que as instâncias não exigem reinicialização após a aplicação de patches, ou se houver aplicações ou processos em execução em uma instância que não devam ser interrompidos por uma reinicialização de operação de patch. Também é útil quando você deseja mais controle sobre o tempo de reinicializações de instâncias, como, por exemplo, usando uma janela de manutenção.

Arquivo de rastreamento de instalação de patches: para rastrear a instalação de patches, sobretudo os patches que foram instalados desde a última reinicialização do sistema, o Systems Manager mantém um arquivo na instância gerenciada.

> **Important**
>
> Não exclua nem modifique o arquivo de monitoramento. Se esse arquivo for excluído ou corrompido, o relatório de conformidade de patch da instância será impreciso. Se isso acontecer, reinicialize a instância e execute uma operação de verificação de patch para restaurar o arquivo.

Esse arquivo de rastreamento é armazenado nos seguintes locais em suas instâncias gerenciadas:

- Sistemas operacionais Linux:
  - `/var/log/amazon/ssm/patch-configuration/patch-states-configuration.json`
  - `/var/log/amazon/ssm/patch-configuration/patch-inventory-from-last-operation.json`
- Sistema operacional Windows Server:
  - `C:\ProgramData\Amazon\PatchBaselineOperations\State\PatchStatesConfiguration.json`
  - `C:\ProgramData\Amazon\PatchBaselineOperations\State\PatchInventoryFromLastOperation.json`

Documento de comando do SSM para a aplicação de patches: **AWS-RunPatchBaselineWithHooks**

O AWS Systems Manager é compatível com o `AWS-RunPatchBaselineWithHooks`, um documento do Systems Manager (documento SSM) para o Patch Manager, uma ferramenta do AWS Systems Manager. Este documento do SSM executa operações de aplicação de patches em nós gerenciados para atualizações de segurança e outros tipos de atualizações.

O `AWS-RunPatchBaselineWithHooks` difere do `AWS-RunPatchBaseline` das seguintes maneiras:

- Um documento do wrapper, `AWS-RunPatchBaselineWithHooks`, é um wrapper para `AWS-RunPatchBaseline` e depende do `AWS-RunPatchBaseline` para algumas de suas operações.
- A operação **`Install`**: o `AWS-RunPatchBaselineWithHooks` oferece suporte a hooks de ciclo de vida executados em pontos designados durante a aplicação de patches do nó gerenciado. Como as instalações de patches às vezes exigem nós gerenciados para reinicializar, a operação de aplicação patches é dividida em dois eventos, para um total de três hooks que oferecem suporte à funcionalidade personalizada. O primeiro hook é antes da operação `Install with NoReboot`. O segundo hook é depois da operação `Install with NoReboot`. O terceiro hook está disponível após a reinicialização do nó gerenciado.
- Sem suporte à lista de patches personalizada–`AWS-RunPatchBaselineWithHooks`não é compatível com o`InstallOverrideList`parâmetro .
- Suporte ao SSM Agent: o `AWS-RunPatchBaselineWithHooks` exige que o SSM Agent 3.0.502 ou posterior seja instalado em um nó gerenciado para a aplicação de patches.

Quando o documento é executado, ele usa a linha de base do patch atualmente especificada como “padrão” para um tipo de sistema operacional se nenhum grupo de patches for especificado. Caso contrário, ele usa as linhas de base do patch associadas ao grupo de patches. Para obter mais informações sobre grupos de patches, consulte Grupos de patches.

Você pode usar o documento `AWS-RunPatchBaselineWithHooks` para aplicar patches aos sistemas operacionais e aplicações. (No Windows Server, o suporte a aplicações é limitado a atualizações de aplicações da Microsoft.)

Este documento oferece suporte a nós gerenciados do Linux e Windows Server. O documento executará as ações adequadas a cada plataforma.

> **Note**
>
> `AWS-RunPatchBaselineWithHooks` não é compatível com macOS.

Linux

Em nós gerenciados do Linux, o documento `AWS-RunPatchBaselineWithHooks` invoca um módulo do Python, que, por sua vez, baixa um snapshot da lista de referência de patches que se aplica ao nó gerenciado. Esse snapshot da lista de referência de patches usa regras e listas de patches aprovados e bloqueados para conduzir o gerenciador de pacotes apropriado para cada tipo de nó:

- Nós gerenciados do Amazon Linux 1, Amazon Linux 2, CentOS, Oracle Linux e RHEL 7 usam YUM. Para operações YUM, o Patch Manager requer o `Python 2.6` ou uma versão posterior compatível (2.6-3.10).
- Os nós gerenciados do RHEL 8 usam o DNF. Para operações DNF, o Patch Manager requer uma versão compatível do `Python 2` ou `Python 3` (2.6-3.10). (Nenhuma versão é instalada por padrão no RHEL 8. É necessário instalar um deles manualmente.)
- As instâncias do Debian Server, do Raspberry Pi OS e do Ubuntu Server usam o APT. Para operações APT, o Patch Manager requer uma versão compatível do `Python 3` (3.0-3.10).
- Os nós gerenciados do SUSE Linux Enterprise Server usam o Zypper. Para operações Zypper, o Patch Manager requer o `Python 2.6` ou uma versão posterior compatível (2.6-3.10).

Windows Server

Em nós gerenciados do Windows Server, o documento `AWS-RunPatchBaselineWithHooks` baixa e invoca um módulo do PowerShell, que, por sua vez, baixa um snapshot da lista de referência de patches que se aplica ao nó gerenciado. Esse snapshot da lista de referência de patches contém uma lista de patches aprovados que é compilada consultando a lista de referência de patches em um servidor Windows Server Update Service (WSUS). Esse snapshot da lista de referência de patches é passado para a API do Windows Update, que controla o download e a instalação de patches aprovados, conforme apropriado.

Cada snapshot é específico para uma Conta da AWS, um grupo de patches, um sistema operacional e um ID de snapshot. O snapshot é entregue por meio de um URL pré-assinado do Amazon Simple Storage Service (Amazon S3), que expira 24 horas após a criação do snapshot. No entanto, depois

que o URL expirar, se você quiser aplicar o mesmo conteúdo de snapshot a outros nós gerenciados, você poderá gerar um novo URL pré-assinado do Amazon S3 até três dias após a criação do snapshot. Para fazer isso, use o comando get-deployable-patch-snapshot-for-instance.

Depois que todas as atualizações aprovadas e aplicáveis forem instaladas, e as reinicializações realizadas conforme a necessidade, as informações de conformidade do patch serão geradas em um nó gerenciado e relatadas ao Patch Manager.

Se o parâmetro `RebootOption` estiver definido como `NoReboot` no documento `AWS-RunPatchBaselineWithHooks`, o nó gerenciado não será reinicializado após a execução do Patch Manager. Para obter mais informações, consulte Nome do parâmetro: `RebootOption`.

> **Important**
>
> Embora a opção `NoReboot` impeça a reinicialização do sistema operacional, ela não impede as reinicializações no nível de serviço que podem ocorrer quando determinados pacotes são atualizados. Por exemplo, atualizar pacotes como o Docker pode acionar reinicializações automáticas de serviços dependentes (como serviços de orquestração de contêineres), mesmo quando `NoReboot` é especificado.

Para obter informações sobre como visualizar dados de conformidade de patches, consulte Sobre a conformidade de patches.

Etapas operacionais do **`AWS-RunPatchBaselineWithHooks`**

Quando o`AWS-RunPatchBaselineWithHook`sé executado, as seguintes etapas são executadas:

1. Scan - Uma operação `Scan` que usa o `AWS-RunPatchBaseline` é executada em um nó gerenciado, e um relatório de conformidade é gerado e carregado.
2. Verificar estados de patch locais- Um script é executado para determinar quais etapas serão executadas com base na operação selecionada e`Scan`Resultado da Etapa 1.
   a. Se a operação selecionada for `Scan`, ela será marcada como concluída. A operação termina.
   b. Se a operação selecionada for `Install`, o Patch Manager avalia o resultado `Scan` da Etapa 1 para determinar o que executar a seguir:
      i. Se nenhum patch ausente for detectado e nenhuma reinicialização pendente for necessária, a operação prosseguirá diretamente para a etapa final (Etapa 8), que inclui um hook fornecido por você. Todas as etapas entre elas são ignoradas.

ii. Se nenhum patch ausente for detectado, mas houver reinicializações pendentes necessárias e a opção de reinicialização selecionada forNoReboot, a operação prossegue diretamente para a etapa final (Etapa 8), que inclui um hook fornecido por você. Todas as etapas entre elas são ignoradas.

iii. Caso contrário, a operação prossegue para a próxima etapa.

3. Operação do hook pré-patch: o documento do SSM que você forneceu para o primeiro hook do ciclo de vida, `PreInstallHookDocName`, é executado no seu nó gerenciado.
4. Instalar com NoReboot - Uma operação `Install` com a opção de reinicialização do `NoReboot` usando `AWS-RunPatchBaseline` é executada em seu nó gerenciado, e um relatório de conformidade é gerado e carregado.
5. Operação do hook pós-instalação: o documento do SSM que você forneceu para o segundo hook do ciclo de vida, `PostInstallHookDocName`, é executado em seu nó gerenciado.
6. Verificar reinicializações - Um script é executado para determinar se uma reinicialização é necessária para o nó gerenciado e quais etapas executar:
   a. Se a opção de reinicialização selecionada for `NoReboot`, a operação prosseguirá diretamente para a etapa final (Etapa 8), que inclui um hook fornecido por você. Todas as etapas entre elas são ignoradas.
   b. Se a opção de reinicialização selecionada for `RebootIfNeeded`, o Patch Manager verifica se há reinicializações pendentes necessárias do inventário coletado na Etapa 4. Isso significa que a operação continua na etapa 7 e o nó gerenciado é reinicializado em um dos seguintes casos:
      i. O Patch Manager instalou um ou mais patches. (O Patch Manager não avalia se o patch exige uma reinicialização. O sistema é reinicializado mesmo que o patch não exija uma reinicialização.)
      ii. O Patch Manager detecta um ou mais patches com um status de `INSTALLED_PENDING_REBOOT` durante a operação de instalação. O status `INSTALLED_PENDING_REBOOT` pode significar que a opção `NoReboot` foi selecionada na última vez em que a operação Instalar foi executada ou que um patch foi instalado fora do Patch Manager na última vez que o nó gerenciado foi reinicializado.

   Se nenhum patch que atenda a esses critérios for encontrado, a operação de aplicação de patch no nó gerenciado será concluída e a operação prosseguirá diretamente para a etapa final (etapa 8), que inclui um hook fornecido por você. Todas as etapas entre elas são ignoradas.
7. Reinicializações e relatórios - Uma operação de instalação com a opção de reinicialização do `RebootIfNeeded` é executado em seu nó gerenciado usando o `AWS-RunPatchBaseline` e um relatório de conformidade é gerado e carregado.

8. Operação de hook pós-reinicialização: o documento do SSM que você forneceu para o terceiro hook do ciclo de vida, `OnExitHookDocName`, é executado em seu nó gerenciado.

Para uma operação de `Scan`, se a Etapa 1 falhar, o processo de execução do documento pára e a etapa é relatada como falha, embora as etapas subsequentes sejam relatadas como bem-sucedidas.

Para uma operação `Install`, se qualquer uma das etapas do `aws:runDocument` falharem durante a operação, essas etapas serão relatadas como falhas e a operação prosseguirá diretamente para a etapa final (Etapa 8), que inclui um hook fornecido por você. Todas as etapas entre elas são ignoradas. Esta etapa é relatada como falhou, a última etapa relata o status de seu resultado de operação e todas as etapas entre são relatadas como bem-sucedidas.

Parâmetros do **`AWS-RunPatchBaselineWithHooks`**

O `AWS-RunPatchBaselineWithHooks` oferece suporte a seis parâmetros.

O parâmetro `Operation` é obrigatório.

Os parâmetros `RebootOption`, `PreInstallHookDocName`, `PostInstallHookDocName` e `OnExitHookDocName` são opcionais.

O `Snapshot-ID` é tecnicamente opcional, mas recomendamos que você forneça um valor personalizado para ele quando executar o `AWS-RunPatchBaselineWithHooks` fora de uma janela de manutenção. Deixe o Patch Manager fornecer o valor automaticamente quando o documento é executado como parte de uma operação de janela de manutenção.

Parâmetros

- Nome do parâmetro: Operation
- Nome do parâmetro: Snapshot ID
- Nome do parâmetro: RebootOption
- Nome do parâmetro: PreInstallHookDocName
- Nome do parâmetro: PostInstallHookDocName
- Nome do parâmetro: OnExitHookDocName

Nome do parâmetro: **`Operation`**

Uso: obrigatório.

Opções: `Scan` | `Install`.

Verificar

Quando você escolhe a opção `Scan`, o sistema usa o documento `AWS-RunPatchBaseline` para determinar o estado de conformidade dos patches do nó gerenciado e retorna essas informações para o Patch Manager. A opção `Scan` não solicita que atualizações sejam instaladas ou que nós gerenciados sejam reinicializados. Em vez disso, a operação identifica onde existem atualizações ausentes que estão aprovadas e são aplicáveis ao nó.

Instalar

Quando você escolhe a opção `Install`, o `AWS-RunPatchBaselineWithHooks` tenta instalar as atualizações aprovadas e aplicáveis que estiverem ausentes em seu nó gerenciado. As informações de conformidade dos patches geradas como parte de uma operação `Install` não listam atualizações ausentes, mas poderão indicar atualizações em estado malsucedido se, por qualquer motivo, a instalação da atualização não tiver tido êxito. Sempre que uma atualização é instalada em um nó gerenciado, o nó é reinicializado para garantir que a atualização esteja instalada e ativa. (Exceção: Se o parâmetro `RebootOption` estiver definido como `NoReboot` no documento `AWS-RunPatchBaselineWithHooks`, o nó gerenciado não será reinicializado depois que o Patch Manager for executado. Para obter mais informações, consulte Nome do parâmetro: `RebootOption`).

> **Note**
>
> Se um patch especificado pelas regras da lista de referência for instalado antes que o Patch Manager atualize o nó gerenciado, o sistema poderá não ser reinicializado conforme esperado. Isso pode acontecer quando um patch é instalado manualmente por um usuário ou instalado automaticamente por outro programa, como o pacote `unattended-upgrades` no Ubuntu Server.

Nome do parâmetro: **`Snapshot ID`**

Uso: opcional.

O `Snapshot ID` é um ID exclusivo (GUID) usado pelo Patch Manager para garantir que um conjunto de nós gerenciados, corrigidos em uma única operação, tenham o mesmo conjunto de patches aprovados. Embora o parâmetro seja definido como opcional, nossa recomendação baseada em práticas recomendadas depende de estar executando ou não o `AWS-RunPatchBaselineWithHooks` em uma janela de manutenção, conforme descrito na tabela a seguir.

Práticas recomendadas do **`AWS-RunPatchBaselineWithHooks`**

Modo	Prática recomendada	Detalhes
Running`AWS-RunPatchBaselineWithHooks` Dentro de uma janela de manutenção	Não forneça um ID de snapshot. O Patch Manager fornecerá para você.	Se você usar uma janela de manutenção para executar o `AWS-RunPatchBaselineWithHooks`, não forneça seu próprio ID de snapshot gerado. Nesse cenário, o Systems Manager fornece um valor GUID com base no ID de execução da janela de manutenção. Isso garante que seja usado um ID correto para todas as invocações do `AWS-RunPatchBaselineWithHooks` nessa janela de manutenção. Se você especificar um valor nesse cenário, observe que talvez o snapshot da lista de referência de patches não permaneça vigente por mais de três dias. Depois disso, um novo snapshot será gerado, mesmo que você especificar o mesmo ID depois que o snapshot expirar.
Running`AWS-RunPatchBaselineWithHooks` Fora de uma janela de manutenção	Gere e especifique um valor de GUID personalizado para o ID do snapshot..[1]	Quando você não estiver usando uma janela de manutenção para executar o `AWS-RunPatchBaselineWithHooks`, recomendamos gerar e especificar um ID de snapshot exclusivo

Modo	Prática recomendada	Detalhes
		para cada lista de referência de patches, principalmente se estiver executando o documento `AWS-RunPatchBaselineWithHooks` em vários nós gerenciados na mesma operação. Se você não especificar um ID nesse cenário, o Systems Manager gerará um ID de snapshot diferente para cada nó gerenciado para o qual o comando for enviado. Em consequência disso, vários conjuntos de patches podem ser especificados entre os nós. Por exemplo, digamos que você esteja executando o documento `AWS-RunPatchBaselineWithHooks` diretamente do Run Command, uma ferramenta do AWS Systems Manager, e visando um grupo de 50 nós gerenciados. Ao especificar um ID de snapshot personalizado, é gerado um único snapshot da lista de referência, que é usado para avaliar e corrigir todos os nós gerenciados, garantindo que eles adquiram um estado consistente.

Modo	Prática recomendada	Detalhes

[1] Você pode usar qualquer ferramenta capaz de gerar um GUID para gerar um valor para o parâmetro ID do snapshot. Por exemplo, no PowerShell, você pode usar o cmdlet `New-Guid` para gerar um GUID no formato `12345699-9405-4f69-bc5e-9315aEXAMPLE` .

Nome do parâmetro: **`RebootOption`**

Uso: opcional.

Opções: `RebootIfNeeded` | `NoReboot`

Padrão: `RebootIfNeeded`

> **Warning**
>
> A opção padrão é `RebootIfNeeded`. Selecione a opção correta para o seu caso de uso. Por exemplo, se os nós gerenciados precisarem reinicializar imediatamente para concluir um processo de configuração, escolha `RebootIfNeeded`. Ou, se você precisar manter a disponibilidade do nó gerenciado até um horário de reinicialização agendado, escolha `NoReboot`.

> **Important**
>
> Não é recomendável usar o Patch Manager para aplicar patches em instâncias de cluster no Amazon EMR (antes chamado de Amazon Elastic MapReduce). Em particular, não selecione a opção `RebootIfNeeded` para o parâmetro `RebootOption`. (Essa opção está disponível nos documentos do SSM Command para aplicação de patches em `AWS-RunPatchBaseline`, `AWS-RunPatchBaselineAssociation` e `AWS-RunPatchBaselineWithHooks`.)
>
> Os comandos subjacentes para aplicação de patches usando o Patch Manager utilizam os comandos `yum` e `dnf`. Portanto, as operações resultam em incompatibilidades por causa da forma como os pacotes são instalados. Para obter informações sobre os métodos preferenciais para atualizar o software nos clusters do Amazon EMR, consulte Using the default AMI for Amazon EMR no Guia de gerenciamento do Amazon EMR.

RebootIfNeeded

Quando você escolhe a opção `RebootIfNeeded`, o nó gerenciado é reinicializado em um dos seguintes casos:

- O Patch Manager instalou um ou mais patches.

  O Patch Manager não avalia se uma reinicialização é exigida pelo patch. O sistema é reinicializado mesmo que o patch não exija uma reinicialização.

- O Patch Manager detecta um ou mais patches com um status de `INSTALLED_PENDING_REBOOT` durante a operação `Install`.

  O status `INSTALLED_PENDING_REBOOT` pode significar que a opção `NoReboot` foi selecionada na última vez em que a operação `Install` foi executada ou que um patch foi instalado fora do Patch Manager na última vez que o nó gerenciado foi reinicializado.

A reinicialização de nós gerenciados nesses dois casos garante que os pacotes atualizados sejam liberados da memória e mantenham o comportamento da aplicação de patches e da reinicialização consistente em todos os sistemas operacionais.

NoReboot

Quando você escolhe a opção `NoReboot`, o Patch Manager não reinicializa um nó gerenciado, mesmo que tenha instalado patches durante a operação `Install`. Essa opção é útil se você souber que os nós gerenciados não exigem reinicialização após a aplicação de patches, ou se houver aplicações ou processos em execução em um nó que não devam ser interrompidos por uma reinicialização da operação de aplicação de patch. Também é útil quando você deseja mais controle sobre o tempo de reinicializações de nós gerenciados, como, por exemplo, usando uma janela de manutenção.

> **Note**
>
> Se você escolher a opção `NoReboot` e um patch for instalado, o patch receberá um status de `InstalledPendingReboot`. O nó gerenciado em si, no entanto, é marcado como `Non-Compliant`. Depois que ocorre uma reinicialização e uma operação `Scan` é executada, o status do nó é atualizado para `Compliant`.

Arquivo de rastreamento de instalação de patches: para rastrear a instalação de patches, sobretudo os patches que foram instalados desde a última reinicialização do sistema, o Systems Manager mantém um arquivo em seu nó gerenciado.

> **Important**
>
> Não exclua nem modifique o arquivo de monitoramento. Se esse arquivo for excluído ou corrompido, o relatório de conformidade de patch do nó gerenciado será impreciso. Se isso acontecer, reinicialize o nó e execute uma operação de verificação de patch para restaurar o arquivo.

Esse arquivo de rastreamento é armazenado nos seguintes locais em seus nós gerenciados:

- Sistemas operacionais Linux:
  - `/var/log/amazon/ssm/patch-configuration/patch-states-configuration.json`
  - `/var/log/amazon/ssm/patch-configuration/patch-inventory-from-last-operation.json`
- Sistema operacional Windows Server:
  - `C:\ProgramData\Amazon\PatchBaselineOperations\State\PatchStatesConfiguration.json`
  - `C:\ProgramData\Amazon\PatchBaselineOperations\State\PatchInventoryFromLastOperation.json`

Nome do parâmetro: **`PreInstallHookDocName`**

Uso: opcional.

Padrão: `AWS-Noop`.

O valor a ser fornecido para o parâmetro `PreInstallHookDocName` é o nome ou o nome do recurso da Amazon (ARN) de um documento do SSM de sua escolha. Você pode fornecer o nome de um documento gerenciado pela AWS ou o nome ou ARN de um documento SSM personalizado que você criou ou que foi compartilhado com você. (Para um documento do SSM que foi compartilhado com você de uma Conta da AWS diferente, você deve especificar o ARN completo do recurso, como `arn:aws:ssm:us-east-2:123456789012:document/MySharedDocument`).

O documento SSM especificado é executado antes da operação `Install` e executa todas as ações suportadas pelo SSM Agent, como um script shell para executar a verificação de integridade da aplicação, antes que o patch seja executado em seu nó gerenciado. Para obter uma lista de ações consulte Referência de plug-ins de documentos de comando). O nome do documento SSM padrão é `AWS-Noop`, que não executa nenhuma operação em um nó gerenciado.

Para obter informações sobre como criar um documento SSM personalizado, consulte Criar conteúdo de documento do SSM.

Nome do parâmetro: **`PostInstallHookDocName`**

Uso: opcional.

Padrão: `AWS-Noop`.

O valor a ser fornecido para o parâmetro `PostInstallHookDocName` é o nome ou o nome do recurso da Amazon (ARN) de um documento do SSM de sua escolha. Você pode fornecer o nome de um documento gerenciado pela AWS ou o nome ou ARN de um documento SSM personalizado que você criou ou que foi compartilhado com você. (Para um documento do SSM que foi compartilhado com você de uma Conta da AWS diferente, você deve especificar o ARN completo do recurso, como `arn:aws:ssm:us-east-2:123456789012:document/MySharedDocument`).

O documento SSM especificado é executado após o`Install with NoReboote` executa todas as ações suportadas peloSSM Agent, como um script shell para instalar atualizações de terceiros antes da reinicialização. Para obter uma lista de ações consulte Referência de plug-ins de documentos de comando). O nome do documento SSM padrão é `AWS-Noop`, que não executa nenhuma operação em um nó gerenciado.

Para obter informações sobre como criar um documento SSM personalizado, consulte Criar conteúdo de documento do SSM.

Nome do parâmetro: **`OnExitHookDocName`**

Uso: opcional.

Padrão: `AWS-Noop`.

O valor a ser fornecido para o parâmetro `OnExitHookDocName` é o nome ou o nome do recurso da Amazon (ARN) de um documento do SSM de sua escolha. Você pode fornecer o nome de um documento gerenciado pela AWS ou o nome ou ARN de um documento SSM personalizado que você criou ou que foi compartilhado com você. (Para um documento do SSM que foi compartilhado

com você de uma Conta da AWS diferente, você deve especificar o ARN completo do recurso, como `arn:aws:ssm:us-east-2:123456789012:document/MySharedDocument`).

O documento SSM especificado é executado após a operação de reinicialização do nó gerenciado e executa todas as ações suportadas pelo SSM Agent, como um script shell para verificar a integridade do nó após a conclusão da operação de aplicação do patch. Para obter uma lista de ações consulte Referência de plug-ins de documentos de comando). O nome do documento SSM padrão é `AWS-Noop`, que não executa nenhuma operação em um nó gerenciado.

Para obter informações sobre como criar um documento SSM personalizado, consulte Criar conteúdo de documento do SSM.

Cenário de exemplo do uso do parâmetro InstallOverrideList em **`AWS-RunPatchBaseline`** ou **`AWS-RunPatchBaselineAssociation`**

Você poderá usar o parâmetro `InstallOverrideList` quando quiser substituir os patches especificados pela lista de referência de patches padrão atual no Patch Manager, uma ferramenta do AWS Systems Manager. Este tópico fornece exemplos que mostram como usar esse parâmetro para obter o seguinte:

- Aplique diferentes conjuntos de patches a um grupo de nós gerenciados de destino.
- Aplique esses conjuntos de patches em diferentes frequências.
- Use a mesma lista de referência de patches para as operações.

Digamos que você quer instalar duas categorias diferentes de patches em nós gerenciados do Amazon Linux 2. Você deseja instalar esses patches em programações diferentes usando janelas de manutenção. Deseja que uma janela de manutenção seja executada todas as semanas e instale todos os patches `Security`. Deseja que outra janela de manutenção seja executada uma vez por mês e instale todos os patches disponíveis ou categorias de patches que não sejam `Security`.

No entanto, só é possível definir uma lista de referência de patches por vez como o padrão para um sistema operacional. Esse requisito ajuda a evitar situações em que uma lista de referência de patches aprova um patch enquanto outro o bloqueia, o que pode levar a problemas entre versões conflitantes.

A estratégia a seguir permite que você use o parâmetro `InstallOverrideList` para aplicar tipos de patches diferentes a um grupo de destino, em diferentes programações, enquanto ainda usa a mesma lista de referência de patches.

1. Na lista de referência de patches padrão, confirme se apenas as atualizações `Security` estão especificadas.
2. Crie uma janela de manutenção que execute o `AWS-RunPatchBaseline` ou `AWS-RunPatchBaselineAssociation` toda semana. Não especifique uma lista de substituição.
3. Crie uma lista de substituição dos patches de todos os tipos que você deseja aplicar mensalmente e armazene-a em um bucket do Amazon Simple Storage Service (Amazon S3).
4. Crie uma segunda janela de manutenção que é executada uma vez por mês. No entanto, para a tarefa do Run Command registrada para essa janela de manutenção, especifique o local da lista de substituição.

O resultado: somente patches `Security`, conforme definido na lista de referência de patches padrão, são instalados toda semana. Todos os patches disponíveis, ou qualquer subconjunto de patches que você definir, são instalados todos os meses.

Para obter mais informações e listas de exemplo, consulte Nome do parâmetro: `InstallOverrideList`.

Usar o parâmetro BaselineOverride

Defina as preferências da aplicação de patches no runtime usando o recurso de substituição da lista de referência no Patch Manager, uma ferramenta do AWS Systems Manager. Faça isso especificando um bucket do Amazon Simple Storage Service (Amazon S3) que contenha um objeto JSON com uma lista de referência de patches. A operação de patch usa as linhas de base fornecidas no objeto JSON que correspondem ao sistema operacional host em vez de aplicar as regras da linha de base do patch padrão.

> **Important**
>
> O nome do arquivo `BaselineOverride` não pode conter os seguintes caracteres: crase (`), aspas simples ('), aspas duplas (") e cifrão ($).

Exceto quando uma operação de patch usa uma política de patch, usar o parâmetro `BaselineOverride` não substitui a conformidade do patch da linha de base fornecida no parâmetro. Os resultados de saída são registrados nos logs de Stdout do Run Command, uma ferramenta do AWS Systems Manager. Os resultados apenas imprimem

pacotes marcados comoNON_COMPLIANT. Isso significa que o pacote está marcado comoMissing,Failed,InstalledRejected, ouInstalledPendingReboot.

No entanto, quando uma operação de patch usa uma política de patch, o sistema passa o parâmetro de substituição do bucket do S3 associado e o valor de conformidade é atualizado para o nó gerenciado. Para obter mais informações sobre comportamentos de políticas de patch, consulte Configurações de políticas de patches em Quick Setup.

Usando a substituição da linha de base do patch com parâmetros Id de Snapshot ou Lista de Sobreposição de Instalação

Há dois casos em que a substituição da linha de base do patch tem um comportamento digno de nota.

Usando substituição de linha de base e ID de Snapshot ao mesmo tempo

Os IDs do snapshot garantem que todos os nós gerenciados em um determinado comando de patch apliquem a mesma coisa. Por exemplo, se você corrigir 1.000 nós gerenciados ao mesmo tempo, os patches serão os mesmos.

Ao usar um ID de Snapshot e uma substituição de linha de base de patch, o ID de Snapshot tem precedência sobre a substituição de linha de base do patch. As regras de substituição de linha de base ainda serão usadas, mas elas serão avaliadas apenas uma vez. No exemplo anterior, os patches em seus 1.000 nós gerenciados continuarão sempre os mesmos. Se, no meio da operação de patch, você alterou o arquivo JSON no bucket S3 referenciado para ser algo diferente, os patches aplicados continuarão sendo os mesmos. Isso ocorre porque o ID do Snapshot foi fornecido.

Usando substituição de linha de base e Lista de Substituição de Instalação ao mesmo tempo

Você não pode usar esses dois parâmetros ao mesmo tempo. O documento de patch falhará se ambos os parâmetros forem fornecidos e não executará nenhuma verificação ou instalação em seu nó gerenciado.

Exemplos de código

O exemplo de código a seguir para Python mostra como gerar a substituição de linha de base do patch.

```
import boto3
import json
```

```
ssm = boto3.client('ssm')
s3 = boto3.resource('s3')
s3_bucket_name = 'my-baseline-override-bucket'
s3_file_name = 'MyBaselineOverride.json'
baseline_ids_to_export = ['pb-0000000000000000', 'pb-0000000000000001']

baseline_overrides = []
for baseline_id in baseline_ids_to_export:
    baseline_overrides.append(ssm.get_patch_baseline(
        BaselineId=baseline_id
    ))

json_content = json.dumps(baseline_overrides, indent=4, sort_keys=True, default=str)
s3.Object(bucket_name=s3_bucket_name, key=s3_file_name).put(Body=json_content)
```

Isso produz uma substituição de linha de base de patch, como a seguir.

```
[
    {
        "ApprovalRules": {
            "PatchRules": [
                {
                    "ApproveAfterDays": 0,
                    "ComplianceLevel": "UNSPECIFIED",
                    "EnableNonSecurity": false,
                    "PatchFilterGroup": {
                        "PatchFilters": [
                            {
                                "Key": "PRODUCT",
                                "Values": [
                                    "*"
                                ]
                            },
                            {
                                "Key": "CLASSIFICATION",
                                "Values": [
                                    "*"
                                ]
                            },
                            {
                                "Key": "SEVERITY",
                                "Values": [
                                    "*"
```

```
                            ]
                        }
                    ]
                }
            }
        ]
    },
    "ApprovedPatches": [],
    "ApprovedPatchesComplianceLevel": "UNSPECIFIED",
    "ApprovedPatchesEnableNonSecurity": false,
    "GlobalFilters": {
        "PatchFilters": []
    },
    "OperatingSystem": "AMAZON_LINUX_2",
    "RejectedPatches": [],
    "RejectedPatchesAction": "ALLOW_AS_DEPENDENCY",
    "Sources": []
},
{
    "ApprovalRules": {
        "PatchRules": [
            {
                "ApproveUntilDate": "2021-01-06",
                "ComplianceLevel": "UNSPECIFIED",
                "EnableNonSecurity": true,
                "PatchFilterGroup": {
                    "PatchFilters": [
                        {
                            "Key": "PRODUCT",
                            "Values": [
                                "*"
                            ]
                        },
                        {
                            "Key": "CLASSIFICATION",
                            "Values": [
                                "*"
                            ]
                        },
                        {
                            "Key": "SEVERITY",
                            "Values": [
                                "*"
                            ]
```

```
                            }
                        ]
                    }
                }
            ]
        },
        "ApprovedPatches": [
            "open-ssl*"
        ],
        "ApprovedPatchesComplianceLevel": "UNSPECIFIED",
        "ApprovedPatchesEnableNonSecurity": false,
        "GlobalFilters": {
            "PatchFilters": []
        },
        "OperatingSystem": "CENTOS",
        "RejectedPatches": [
            "python*"
        ],
        "RejectedPatchesAction": "ALLOW_AS_DEPENDENCY",
        "Sources": []
    }
]
```

Linhas de base de patch

Os tópicos nesta seção fornecem informações sobre como as listas de referência de patches funcionam no Patch Manager, uma ferramenta do AWS Systems Manager, quando você executa uma operação `Scan` ou `Install` em seus nós gerenciados.

Tópicos

- Listas de referência de patches predefinidas e personalizadas
- Sobre formatos de nomes de pacotes para listas de patches aprovados e rejeitados
- Grupos de patches
- Aplicações de patches lançadAs pela Microsoft no Windows Server

Listas de referência de patches predefinidas e personalizadas

O Patch Manager, uma ferramenta do AWS Systems Manager, fornece uma lista de referência de patches predefinida para cada sistema operacional compatível com o Patch Manager. É possível usar essas listas de referência como estão configuradas no momento (não é possível personalizá-

las) ou criar suas próprias listas de referência de patches personalizadas. As listas de referência de patches personalizadas permitem um controle maior sobre quais patches são aprovados ou rejeitados para o ambiente. Além disso, as listas de referência predefinidas atribuem um nível de conformidade de `Unspecified` a todos os patches instalados usando essas listas de referência. Para que os valores de conformidade sejam atribuídos, é possível criar uma cópia de uma lista de referência predefinida e especificar os valores de conformidade que deseja atribuir aos patches. Para obter mais informações, consulte Linhas de base personalizadas e Trabalhando com linhas de base de patch personalizadas.

> **Note**
>
> As informações neste tópico se aplicam independentemente do método ou tipo de configuração que você esteja usando para suas operações de aplicação de patch:
>
> - Uma política de patch configurada no Quick Setup
> - Uma opção do Host Management configurada no Quick Setup
> - Uma janela de manutenção para executar um patch `Scan` ou tarefa `Install`
> - Uma operação Patch now (Aplicar patch agora) sob demanda

Tópicos

- Linhas de base predefinidas
- Linhas de base personalizadas

Linhas de base predefinidas

A tabela a seguir descreve as linhas de base de patch predefinidas fornecidas com o Patch Manager.

Para obter informações sobre quais versões de cada sistema operacional são compatíveis com o Patch Manager, consulte Pré-requisitos da Patch Manager.

Name	Sistema operacional com suporte	Detalhes
`AWS-AlmaLinuxDefaultPatchBaseline`	AlmaLinux	Aprova todos os patches do sistema operacional classificados como "Security

Name	Sistema operacional com suporte	Detalhes
		" (Segurança) e com nível de gravidade "Critical" (Crítico) ou "Important" (Importante). Também aprova todos os patches classificados como “Bugfix” (Correção de erros). Os patches são aprovados automaticamente sete dias após serem lançados ou atualizados.[1]
`AWS-AmazonLinuxDefaultPatchBaseline`	Amazon Linux 1	Aprova todos os patches do sistema operacional classificados como "Security " (Segurança) e com nível de gravidade "Critical" (Crítico) ou "Important" (Importante). Também aprova automatic amente todos os patches com uma classificação de “Bugfix” (Correção de erros). Os patches são aprovados automaticamente sete dias após serem lançados ou atualizados.[1]

Name	Sistema operacional com suporte	Detalhes
`AWS-AmazonLinux2DefaultPatchBaseline`	Amazon Linux 2	Aprova todos os patches do sistema operacional classificados como "Security" (Segurança) e com nível de gravidade "Critical" (Crítico) ou "Important" (Importante). Também aprova todos os patches com uma classificação de “Bugfix” (Correção de erros). Os patches são aprovados automaticamente sete dias após o lançamento.[1]
`AWS-AmazonLinux2022DefaultPatchBaseline`	Amazon Linux 2022	Aprova todos os patches do sistema operacional classificados como "Security" (Segurança) e com nível de gravidade "Critical" (Crítico) ou "Important" (Importante). Os patches são aprovados automaticamente sete dias após a liberação. Também aprova todos os patches com uma classificação de "Bugfix" (Correção de erros) sete dias após o lançamento.

Name	Sistema operacional com suporte	Detalhes
`AWS-AmazonLinux2023DefaultPatchBaseline`	Amazon Linux 2023	Aprova todos os patches do sistema operacional classificados como "Security" (Segurança) e com nível de gravidade "Critical" (Crítico) ou "Important" (Importante). Os patches são aprovados automaticamente sete dias após a liberação. Também aprova todos os patches com uma classificação de "Bugfix" (Correção de erros) sete dias após o lançamento.
`AWS-CentOSDefaultPatchBaseline`	CentOS e CentOS Stream	Aprova todas as atualizações sete dias após elas serem disponibilizadas, incluindo atualizações que não sejam de segurança.
`AWS-DebianDefaultPatchBaseline`	Debian Server	Aprova imediatamente todos os patches relacionados à segurança do sistema operacional com prioridade "Required", (Obrigatório), "Standard" (Padrão) ou "Extra". Não há espera antes da aprovação, pois as datas de lançamento confiáveis não estão disponíveis nos repositórios.

Name	Sistema operacional com suporte	Detalhes
`AWS-MacOSDefaultPatchBaseline`	macOS	Aprova todos os patches do sistema operacional classificados como “Security ” (Segurança). Também aprova todos os pacotes com uma atualização atual.
`AWS-OracleLinuxDefaultPatchBaseline`	Oracle Linux	Aprova todos os patches do sistema operacional classificados como "Security " (Segurança) e com nível de gravidade "Importan t" (Importante) ou "Moderate " (Moderado). Também aprova todos os patches classific ados como “Bugfix” (Correção de erros) sete dias após o lançamento. Os patches são aprovados automatic amente sete dias após serem lançados ou atualizados.[1]
`AWS-DefaultRaspbianPatchBaseline`	Raspberry Pi OS	Aprova imediatamente todos os patches relacionados à segurança do sistema operacional com prioridad e "Required", (Obrigató rio), "Standard" (Padrão) ou "Extra". Não há espera antes da aprovação, pois as datas de lançamento confiáveis não estão disponíveis nos repositórios.

Name	Sistema operacional com suporte	Detalhes
`AWS-RedHatDefaultPatchBaseline`	Red Hat Enterprise Linux (RHEL)	Aprova todos os patches do sistema operacional classificados como "Security" (Segurança) e com nível de gravidade "Critical" (Crítico) ou "Important" (Importante). Também aprova todos os patches classificados como “Bugfix” (Correção de erros). Os patches são aprovados automaticamente sete dias após serem lançados ou atualizados.[1]
`AWS-RockyLinuxDefaultPatchBaseline`	Rocky Linux	Aprova todos os patches do sistema operacional classificados como "Security" (Segurança) e com nível de gravidade "Critical" (Crítico) ou "Important" (Importante). Também aprova todos os patches classificados como “Bugfix” (Correção de erros). Os patches são aprovados automaticamente sete dias após serem lançados ou atualizados.[1]

Name	Sistema operacional com suporte	Detalhes
`AWS-SuseDefaultPatchBaseline`	SUSE Linux Enterprise Server (SLES)	Aprova todos os patches do sistema operacional classificados como "Security" (Segurança) e com gravidade "Critical" (Crítico) ou "Important" (Importante). Os patches são aprovados automaticamente sete dias após serem lançados ou atualizados.[1]
`AWS-UbuntuDefaultPatchBaseline`	Ubuntu Server	Aprova imediatamente todos os patches relacionados à segurança do sistema operacional com prioridade "Required", (Obrigatório), "Standard" (Padrão) ou "Extra". Não há espera antes da aprovação, pois as datas de lançamento confiáveis não estão disponíveis nos repositórios.

Name	Sistema operacional com suporte	Detalhes
`AWS-DefaultPatchBaseline`	Windows Server	Aprova todos os patches do sistema operacional Windows Server classificados como "CriticalUpdates" (Atualizações críticas) ou "SecurityUpdates" (Atualizações de segurança) e com um nível de gravidade MSRC "Critical" (Crítico) ou "Important" (Importante). Os patches são aprovados automaticamente sete dias após serem lançados ou atualizados.[2]
`AWS-WindowsPredefinedPatchBaseline-OS`	Windows Server	Aprova todos os patches do sistema operacional Windows Server classificados como "CriticalUpdates" (Atualizações críticas) ou "SecurityUpdates" (Atualizações de segurança) e com um nível de gravidade MSRC "Critical" (Crítico) ou "Important" (Importante). Os patches são aprovados automaticamente sete dias após serem lançados ou atualizados.[2]

Name	Sistema operacional com suporte	Detalhes
`AWS-WindowsPredefinedPatchBaseline-OS-Applications`	Windows Server	Para o sistema operacional Windows Server, aprova todos os patches classificados como "CriticalUpdates" (Atualizações críticas) ou "SecurityUpdates" (Atualizações de segurança) e com um nível de gravidade MSRC "Critical" (Crítico) ou "Important" (Importante). Para aplicações lançadas pela Microsoft, aprova todos os patches. Os patches para sistemas operacionais e aplicações são aprovados automaticamente sete dias após serem lançados ou atualizados.[2]

[1] Para Amazon Linux 1 e Amazon Linux 2, a espera de 7 dias antes da aprovação automática dos patches é calculada a partir de um valor `Updated Date` em `updateinfo.xml` e não um valor `Release Date`. Vários fatores podem afetar o valor `Updated Date`. Outros sistemas operacionais processam datas de lançamento e atualização de modo diferente. Para obter informações que podem ajudar você a evitar resultados inesperados com atrasos na aprovação automática, consulte Como as datas de lançamento e atualização de pacotes são calculadas.

[2] Para o Windows Server, as listas de referência padrão incluem um atraso de sete dias para a aprovação automática. Para instalar um patch em até sete dias após o lançamento, você deve criar uma lista de referência personalizada.

Linhas de base personalizadas

Use as informações a seguir para ajudar você a criar listas de referência de patches personalizadas para atender às suas metas de patches.

Tópicos

- Usar aprovações automáticas em listas de referência personalizadas
- Informações adicionais para criar listas de referência de patches

Usar aprovações automáticas em listas de referência personalizadas

Se você criar sua própria linha de base para patch, poderá escolher quais patches serão automaticamente aprovados, usando as seguintes categorias.

- Sistema operacional: Windows Server, Amazon Linux, Ubuntu Server e assim por diante.
- Nome do produto (para sistemas operacionais): RHEL 6.5, Amazon Linux 2014.09, Windows Server 2012, Windows Server 2012 R2 e assim por diante.
- Nome do produto (somente para aplicações lançadas pela Microsoft no Windows Server): por exemplo, Word 2016, BizTalk Server e assim por diante.
- Classificação: por exemplo, atualizações críticas, atualizações de segurança e assim por diante.
- Gravidade: por exemplo, crítica, importante e assim por diante.

Para cada regra de aprovação criada, é possível optar por especificar um atraso de aprovação automática ou especificar uma data-limite de aprovação de patch.

> **Note**
>
> Como não é possível determinar de forma confiável as datas de lançamento dos pacotes de atualização do Ubuntu Server, as opções de aprovação automática não são compatíveis com esse sistema operacional.

Um atraso na aprovação automática é o número de dias para aguardar após o lançamento ou a última atualização do patch, antes que a aplicação do patch seja aprovada automaticamente. Por exemplo, se você criar uma regra usando a classificação `CriticalUpdates` e configurá-la para atraso de aprovação automática de 7 dias, um novo patch crítico lançado em 7 de julho será automaticamente aprovado em 14 de julho.

Se um repositório do Linux não fornecer informações de data de lançamento para pacotes, o Patch Manager usará o horário de compilação do pacote como a data para especificações de data de aprovação automática para Amazon Linux 1, Amazon Linux 2, SUSE Linux Enterprise Server

(SLES), Red Hat Enterprise Linux (RHEL) e CentOS. Se o horário de compilação do pacote não puder ser determinado, o Patch Manager usa uma data padrão de 1º de janeiro de 1970. Isso resulta no Patch Manager ignorar quaisquer especificações de data de aprovação automática nas lista de referência de patches que estão configuradas para aprovar patches para qualquer data após 1º de janeiro de 1970.

Quando você especifica uma data-limite de aprovação automática, o Patch Manager aplica automaticamente todos os patches lançados ou com a última atualização nessa data ou antes dela. Por exemplo, se você especificar 7 de julho de 2023 como a data-limite, nenhum patch lançado ou com a última atualização em ou após 8 de julho de 2023 será instalado automaticamente.

Ao criar uma linha de lista de referência personalizada, é possível especificar um nível de gravidade de conformidade para patches aprovados por essa lista de referência de patches, como `Critical` ou `High`. Se o estado do patch de qualquer patch aprovado for relatado como `Missing`, a gravidade geral da conformidade relatada pela lista de referência de patches será o nível de gravidade especificado.

Informações adicionais para criar listas de referência de patches

Lembre-se do seguinte ao criar uma linha de base de patch:

- O Patch Manager fornece uma linha de base de patch predefinida para cada sistema operacional compatível. Essas linhas de base de patch predefinidas são usadas como as linhas de base de patch padrão para cada tipo de sistema operacional, a menos que você crie sua própria linha de base de patch e a designe como a padrão para o tipo de sistema operacional correspondente.

  > **Note**
  >
  > Para o Windows Server, três linhas de base de patch predefinidas são fornecidas. As listas de referência de patches `AWS-DefaultPatchBaseline` e `AWS-WindowsPredefinedPatchBaseline-OS` oferecem suporte apenas às atualizações do sistema operacional Windows em si. O `AWS-DefaultPatchBaseline` é usado como lista de referência de patches para nós gerenciados do Windows Server, a menos que você especifique outra lista de referência de patches. As definições de configuração nestas duas linhas de base de patch são as mesmas. O mais novo dos dois,`AWS-WindowsPredefinedPatchBaseline-OS`, foi criado para distingui-lo da terceira linha de base de patch predefinida paraWindows Server. Essa lista de referência do patch, `AWS-WindowsPredefinedPatchBaseline-OS-Applications`, pode ser usada

para aplicar patches tanto ao sistema operacional Windows Server quanto a aplicações compatíveis lançadas pela Microsoft.

- Por padrão, o Windows Server 2019 e o Windows Server 2022 removem atualizações que são substituídas por atualizações posteriores. Como resultado, se você usar o parâmetro `ApproveUntilDate` em uma lista de referência de patches do Windows Server, mas a data selecionada no parâmetro `ApproveUntilDate` for anterior à data do patch mais recente, o novo patch não será instalado quando a operação de patch for executada. Para obter mais informações sobre como criar regras de aplicação de patches no Windows Server, consulte a guia Windows Server em Como os patches de segurança são selecionados.

  Isso significa que o nó gerenciado é compatível em termos de operações do Systems Manager, mesmo que um patch crítico do mês anterior possa não ter sido instalado. Esse mesmo cenário pode ocorrer ao usar o parâmetro `ApproveAfterDays`. Devido ao comportamento do patch substituído pela Microsoft, é possível definir um número (geralmente maior que 30 dias) para que os patches do Windows Server nunca sejam instalados se o patch mais recente disponível da Microsoft for lançado antes de o número de dias em `ApproveAfterDays` tiver passado.

- Apenas para Windows Server, um patch de atualização de segurança disponível que não seja aprovado pela lista de referência de patches pode ter um valor de conformidade de `Compliant` ou `Non-Compliant`, conforme definido em uma lista de referência de patches personalizada.

  Ao criar ou atualizar uma lista de referência de patches, você escolhe o status que deseja atribuir aos patches de segurança que estão disponíveis, mas não aprovados, porque não atendem aos critérios de instalação especificados na lista de referência de patches. Por exemplo, os patches de segurança que você talvez queira instalar poderão ser ignorados se você tiver especificado um longo período de espera após o lançamento de um patch antes da instalação. Se uma atualização do patch for lançada durante o período de espera especificado, o período de espera para instalação do patch recomeçará. Se o período de espera for muito longo, várias versões do patch poderão ser lançadas, mas nunca instaladas.

  Usando o console para criar ou atualizar uma lista de referência de patches, você especifica essa opção no campo Status de conformidade das atualizações de segurança disponíveis. Usando a AWS CLI para executar o comando create-patch-baseline ou update-patch-baseline, você especifica essa opção no parâmetro `available-security-updates-compliance-status`.

- Para servidores on-premises e máquinas virtuais (VMs), o Patch Manager tenta usar a lista de referência de patches padrão personalizada. Se não existir uma linha de base de patch padrão

personalizada, o sistema usará a linha de base de patch predefinida para o sistema operacional correspondente.

- Se um patch estiver listado como aprovado e rejeitado na mesma linha de base de patch, o patch será rejeitado.
- Um nó gerenciado pode ter apenas uma lista de referência de patches definida para ele.
- Os formatos de nomes de pacotes que você pode adicionar a listas de patches aprovados e rejeitados para uma linha de base de patch dependem do tipo de sistema operacional no qual você está aplicando patches.

  Para obter mais informações sobre os formatos aceitos de listas de patches aprovados e rejeitados, consulte Sobre formatos de nomes de pacotes para listas de patches aprovados e rejeitados.
- Se você estiver usando uma configuração de política de patch em Quick Setup, as atualizações feitas nas listas de referência de patches personalizadas serão sincronizadas com Quick Setup uma vez por hora.

  Se uma lista de referência de patches personalizada que foi referenciada em uma política de patch for excluída, um banner será exibido na página Configuration details (Detalhes da configuração) do Quick Setup da sua política de patch. O banner informa que a política de patch faz referência a uma lista de referência de patches que não existe mais e que as operações de aplicação de patches subsequentes falharão. Nesse caso, retorne à página Configurations (Configurações) do Quick Setup, selecione a configuração Patch Manager e escolha Actions (Ações), Edit configuration (Editar configuração). O nome da lista de referência de patches excluída será destacado, e você deverá selecionar uma nova lista de referência de patches para o sistema operacional afetado.

Para obter mais informações sobre a linha de base de patch, consulte Trabalhando com linhas de base de patch personalizadas e Tutorial: Aplicar patches a um ambiente de servidor usando a AWS CLI.

Sobre formatos de nomes de pacotes para listas de patches aprovados e rejeitados

Os formatos de nomes de pacotes que você pode adicionar a listas de patches aprovados e rejeitados dependem do tipo de sistema operacional no qual você está aplicando patches.

Formatos de nomes de pacotes para sistemas operacionais Linux

Os formatos que você pode especificar para patches aprovados e rejeitados em sua linha de base de patches variam de acordo com o tipo do Linux. Mais especificamente, os formatos compatíveis dependem do gerenciador de pacotes usado pelo tipo de sistema operacional Linux.

Tópicos

- Amazon Linux 1, Amazon Linux 2, Amazon Linux 2022, Amazon Linux 2023, CentOS, Oracle Linux e Red Hat Enterprise Linux (RHEL)
- Debian Server, Raspberry Pi OS (anteriormente Raspbian) e Ubuntu Server
- SUSE Linux Enterprise Server (SLES)

Amazon Linux 1, Amazon Linux 2, Amazon Linux 2022, Amazon Linux 2023, CentOS, Oracle Linux e Red Hat Enterprise Linux (RHEL)

Gerenciador de pacotes: YUM, exceto no Amazon Linux 2022, no Amazon Linux 2023, no RHEL 8 e no CentOS 8, que usam o DNF como gerenciador de pacotes

Patches aprovados: para patches aprovados, você pode especificar qualquer um dos seguintes:

- IDs Bugzilla, no formato `1234567` (o sistema processa somente números de sequências como IDs Bugzilla).
- IDs CVE, no formato `CVE-2018-1234567`
- IDs de consultoria em formatos como `RHSA-2017:0864` e `ALAS-2018-123`
- Nomes de pacotes criados usando um ou mais dos componentes disponíveis para nomeação de pacotes. Para ilustrar, para o pacote chamado `dbus.x86_64:1:1.12.28-1.amzn2023.0.1`, os componentes são os seguintes:
  - `name`: `dbus`
  - `architecture`: `x86_64`
  - `epoch`: `1`
  - `version`: `1.12.28`
  - `release`: `1.amzn2023.0.1`

  Nomes de pacotes com as seguintes construções são aceitos:
  - `name`
  - `name.arch`

- `name-version`
- `name-version-release`
- `name-version-release.arch`
- `version`
- `version-release`
- `epoch:version-release`
- `name-epoch:version-release`
- `name-epoch:version-release.arch`
- `epoch:name-version-release.arch`
- `name.arch:epoch:version-release`

  Alguns exemplos:

  - `dbus.x86_64`
  - `dbus-1.12.28`
  - `dbus-1.12.28-1.amzn2023.0.1`
  - `dbus-1:1.12.28-1.amzn2023.0.1.x86_64`
- Também oferecemos suporte a componentes de nome de pacote com um único curinga nos formatos acima, como os seguintes:
  - `dbus*`
  - `dbus-1.12.2*`
  - `dbus-*:1.12.28-1.amzn2023.0.1.x86_64`

Patches rejeitados: para patches rejeitados, você pode especificar qualquer um dos seguintes:

- Nomes de pacotes criados usando um ou mais dos componentes disponíveis para nomeação de pacotes. Para ilustrar, para o pacote chamado `dbus.x86_64:1:1.12.28-1.amzn2023.0.1`, os componentes são os seguintes:
  - `name`: `dbus`
  - `architecture`; `x86_64`
  - `epoch`: `1`
  - `version`: `1.12.28`
  - `release`: `1.amzn2023.0.1`

Nomes de pacotes com as seguintes construções são aceitos:

- `name`
- `name.arch`
- `name-version`
- `name-version-release`
- `name-version-release.arch`
- `version`
- `version-release`
- `epoch:version-release`
- `name-epoch:version-release`
- `name-epoch:version-release.arch`
- `epoch:name-version-release.arch`
- `name.arch:epoch:version-release`

Alguns exemplos:

- `dbus.x86_64`
- `dbus-1.12.28`
- `dbus-1.12.28-1.amzn2023.0.1`
- `dbus-1:1.12.28-1.amzn2023.0.1.x86_64`

- Também oferecemos suporte a componentes de nome de pacote com um único curinga nos formatos acima, como os seguintes:
  - `dbus*`
  - `dbus-1.12.2*`
  - `dbus-*:1.12.28-1.amzn2023.0.1.x86_64`

Debian Server, Raspberry Pi OS (anteriormente Raspbian) e Ubuntu Server

Gerenciador de pacotes: APT

Patches aprovados e patches rejeitados: para patches aprovados e rejeitados, especifique o seguinte:

- Nomes de pacotes no formato `ExamplePkg33`

> **Note**
>
> Para listas do Debian Server, Raspberry Pi OS e Ubuntu Server, não inclua elementos como arquitetura ou versões. Por exemplo, você especifica o nome do pacote `ExamplePkg33` para incluir todos os seguintes em uma lista de patches:
>
> - `ExamplePkg33.x86.1`
> - `ExamplePkg33.x86.2`
> - `ExamplePkg33.x64.1`
> - `ExamplePkg33.3.2.5-364.noarch`

SUSE Linux Enterprise Server (SLES)

Gerenciador de pacotes: Zypper

Patches aprovados e patches rejeitados: para listas de patches aprovados e rejeitados, você pode especificar o seguinte:

- Nomes de pacotes completos em formatos como:
  - `SUSE-SLE-Example-Package-12-2018-123`
  - `example-pkg-2018.11.4-46.17.1.x86_64.rpm`
- Nomes de pacotes com um único caractere curinga, como:
  - `SUSE-SLE-Example-Package-12-2018-*`
  - `example-pkg-2018.11.4-46.17.1.*.rpm`

Formatos de nomes de pacotesmacOS

Gerenciadores de pacote compatíveis: atualização de software, instalador, Brew, Brew Cask

Patches aprovados e patches rejeitados: para listas de patches aprovados e rejeitados, especifique os nomes completos do pacote, em formatos como:

- `XProtectPlistConfigData`
- `MRTConfigData`

Curingas não são compatíveis com listas de patches aprovados e rejeitados para macOS.

Formatos de nomes de pacotes para sistemas operacionais Windows

Para sistemas operacionais Windows, especifique os patches usando IDs da Base de Dados de Conhecimento Microsoft e IDs do boletim de segurança da Microsoft; por exemplo:

```
KB2032276,KB2124261,MS10-048
```

Grupos de patches

> **Note**
>
> Os grupos de patches não são usados em operações de aplicação de patches em políticas de patch. Para obter informações sobre como trabalhar com políticas de patch, consulte Configurações de políticas de patches em Quick Setup.
> Não há suporte no console à funcionalidade de grupos de patches para pares de conta-região que ainda não usavam grupos de patches antes do lançamento do suporte à política de patches em 22 de dezembro de 2022. A funcionalidade de grupo de patches ainda está disponível em pares de conta-região que começaram a usar grupos de patches antes dessa data.

Você pode usar um grupo de patches para associar os nós gerenciados a uma lista de referência de patches específica no Patch Manager, uma ferramenta do AWS Systems Manager. Grupos de patches ajudam a garantir que você esteja implantando os patches apropriados, com base nas regras de lista de referência de patches associadas, para corrigir o conjunto de nós. Grupos de patches também podem ajudar você a evitar a implantação de patches antes que eles tenham sido adequadamente testados. Por exemplo, você pode criar grupos de patches para diferentes ambientes (como Desenvolvimento, Teste e Produção) e registrar cada grupo de patches em uma linha de base de patch apropriada.

Ao executar o `AWS-RunPatchBaseline` ou outros documentos do comando do SSM para aplicar patches, você pode direcionar os nós gerenciados usando o ID ou as tags do nó. Em seguida, o SSM Agent e o Patch Manager avaliam qual lista de referência de patches deve ser usada, com base no valor do grupo de patches que você adicionou ao nó gerenciado.

Usando tags para definir grupos de patches

É possível criar um grupo de patches ao usar tags aplicadas as instâncias do Amazon Elastic Compute Cloud (Amazon EC2) e nós que não são do EC2 em seu ambiente híbrido e multinuvem. Observe os seguintes detalhes sobre o uso de tags para grupos de patches:

- Um grupo de patches deve ser definido usando a chave de tag `Patch Group` ou `PatchGroup` aplicada aos seus nós gerenciados. Ao registrar um grupo de patches para uma lista de referência de patches, quaisquer valores idênticos especificados para essas duas chaves são interpretados como parte do mesmo grupo. Por exemplo, digamos que você tenha marcado cinco nós com o primeiro dos seguintes pares de valores-chave e cinco com o segundo:
  - `key=PatchGroup,value=DEV`
  - `key=Patch Group,value=DEV`

  O comando do Patch Manager para criar uma linha de base combina esses 10 nós gerenciados em um único grupo com base no valor `DEV`. A AWS CLI equivalente para o comando criar uma lista de referência de patches para grupos de patches é o seguinte:

  ```
  aws ssm register-patch-baseline-for-patch-group \
      --baseline-id pb-0c10e65780EXAMPLE \
      --patch-group DEV
  ```

  A combinação de valores de chaves diferentes em um único destino é exclusiva desse comando do Patch Manager para criar um novo grupo de patches e não é suportada por outras ações da API. Por exemplo, se você executar ações send-command usando chaves do `Patch Group` e do `PatchGroup` com os mesmos valores, terá como alvo dois conjuntos de nós completamente diferentes:

  ```
  aws ssm send-command \
      --document-name AWS-RunPatchBaseline \
      --targets "Key=tag:PatchGroup,Values=DEV"
  ```

  ```
  aws ssm send-command \
      --document-name AWS-RunPatchBaseline \
      --targets "Key=tag:Patch Group,Values=DEV"
  ```
- Há limites na segmentação baseada em tags. Cada matriz de alvos para `SendCommand` pode conter no máximo cinco pares de chave-valor.

- Recomendamos que você escolha apenas uma dessas convenções de chave de tag, seja `PatchGroup` (sem espaço) ou `Patch Group` (com espaço). No entanto, se você tiver tags permitidas nos metadados da instância do EC2, em uma instância, você deve usar o `PatchGroup`.
- Observe que a chave faz distinção entre maiúsculas e minúsculas. Você pode especificar qualquer valor para ajudar você a identificar e direcionar os recursos desse grupo, por exemplo, “servidores web” ou “US-EAST-PROD”, mas a chave deve ser `Patch Group` ou `PatchGroup`.

Depois de criar um grupo de patches e marcar os nós gerenciados, você poderá registrar o grupo de patches com uma lista de referência de patches. Registrar o grupo de patches em uma lista de referência de patches garante que os nós nesse grupo usem as regras definidas na lista de referência de patches associada.

Para obter mais informações sobre como criar um grupo de patches e associá-lo a uma lista de referência dos patches, consulte Como criar e gerenciar grupos de patches e Adicionar um grupo de patches a uma lista de referência de patches.

Para visualizar um exemplo de como criar uma lista de referência de patches e grupos de patches usando a AWS Command Line Interface (AWS CLI), consulte Tutorial: Aplicar patches a um ambiente de servidor usando a AWS CLI. Para obter mais informações, consulte Marcar recursos do Amazon EC2 no Guia do usuário do Amazon EC2 para instâncias do Linux.

Como funciona

Quando o sistema executar a tarefa para aplicar uma lista de referência de patches a um nó gerenciado, o SSM Agent verificará se um grupo de patches está definido para esse nó. Se o nó estiver atribuído a um grupo de patches, o Patch Manager verificará se a lista de referência de patches está registrada nesse grupo. Se uma lista de referência de patches for encontrada para esse grupo, o Patch Manager notificará o SSM Agent para usar a lista de referência de patches associada. Se um nó não estiver configurado para um grupo de patches, o Patch Manager notificará automaticamente o SSM Agent para usar a lista de referência de patches padrão atualmente configurada.

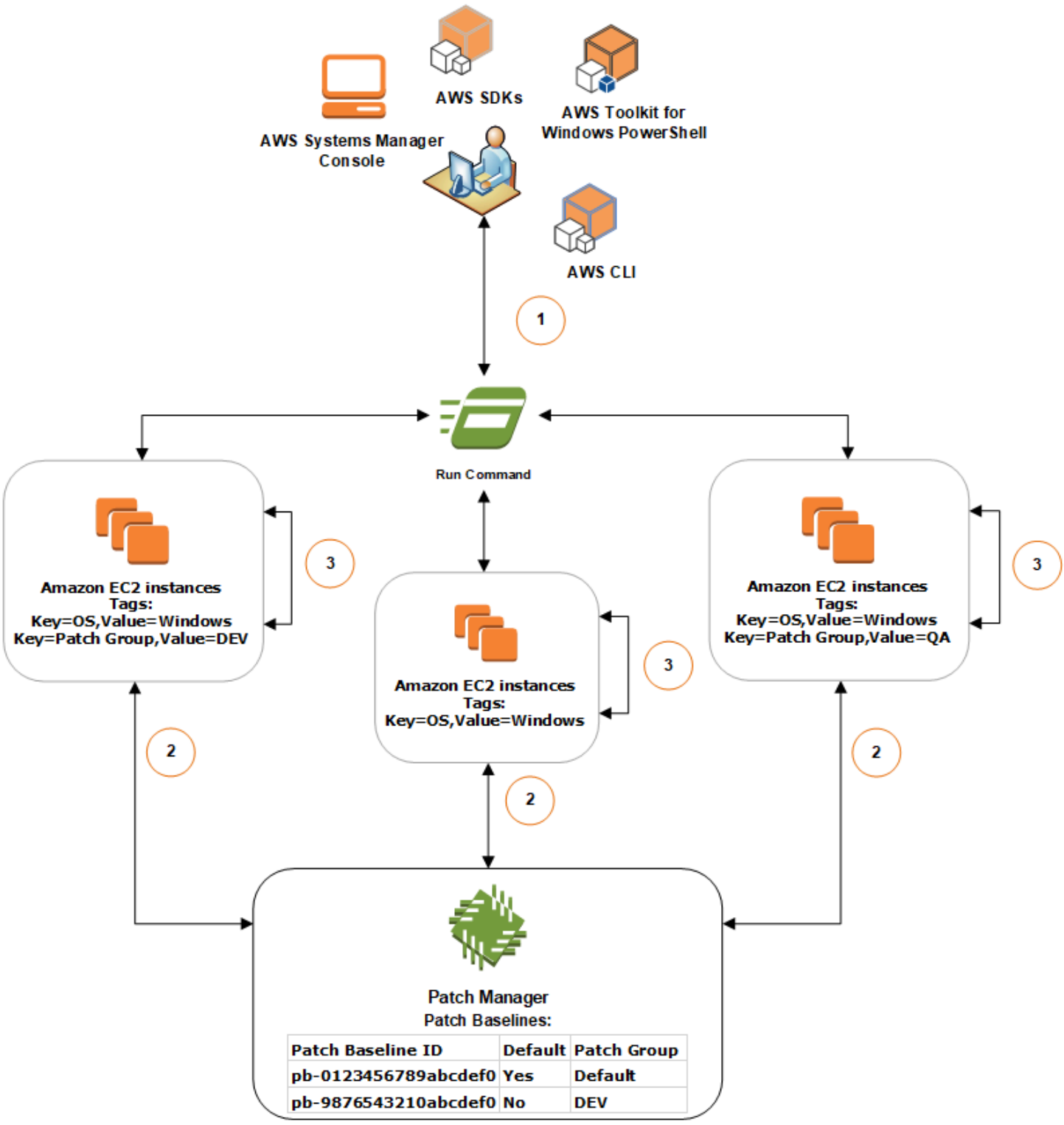
> **Important**
>
> Um nó gerenciado só pode estar em um grupo de patches.
> Um grupo de patches pode ser registrado em somente uma linha de base de patch para cada tipo de sistema operacional.

Não é possível aplicar a tag `Patch Group` (com um espaço) a uma instância do Amazon EC2 se a opção Allow tags in instance metadata (Permitir tags em metadados de instância) estiver habilitada na instância. Permitir tags em metadados de instância impede que nomes de chaves de tag contenham espaços. Se você tiver permissão para tags nos metadados da instância do EC2, é necessário usar a chave de tag `PatchGroup` (sem um espaço).

Diagrama 1: exemplo geral do fluxo de processos de operações da aplicação de patches

A ilustração a seguir mostra um exemplo geral dos processos que o Systems Manager realiza ao enviar uma tarefa do Run Command à sua frota de servidores para aplicar patches usando o Patch Manager. Esses processos determinam quais listas de referência de patches usar nas operações de aplicação de patches. (Um processo semelhante é usado quando uma janela de manutenção estiver configurada para enviar um comando para aplicar patches usando o Patch Manager.)

O processo completo é explicado abaixo da ilustração.



Neste exemplo, temos três grupos de instâncias do EC2 para Windows Server, com as seguintes tags aplicadas:

Grupo de instâncias do EC2	Tags
Grupo 1	`key=OS,value=Windows` `key=PatchGroup,value=DEV`
Grupo 2	`key=OS,value=Windows`
Grupo 3	`key=OS,value=Windows` `key=PatchGroup,value=QA`

Para este exemplo, também temos essas duas listas de referência de patches do Windows Server:

ID da linha de base do patch	Padrão	Grupo de patches associado
`pb-0123456789abcdef0`	Sim	`Default`
`pb-9876543210abcdef0`	Não	`DEV`

O processo geral para verificar ou instalar patches usando o Run Command, uma ferramenta do AWS Systems Manager, e o Patch Manager é o seguinte:

1. Enviar um comando para patch: use o console do Systems Manager, o SDK, a AWS Command Line Interface (AWS CLI) ou o AWS Tools for Windows PowerShell para enviar uma tarefa de Run Command usando o documento `AWS-RunPatchBaseline`. O diagrama mostra uma tarefa de Run Command para aplicar patch a instâncias gerenciadas especificando a etiqueta `key=OS,value=Windows`.
2. Determinação da lista de referência de patches: o SSM Agent verifica as etiquetas de grupo de patches aplicadas à instância do EC2 e consulta o Patch Manager em busca da lista de referência de patches correspondente.
   - Valor de grupo de patches correspondente associado à linha de base de patch:

     1. O SSM Agent, que está instalado em instâncias do EC2 no grupo 1, recebe o comando emitido na Etapa 1 para iniciar uma operação de aplicação de patch. O SSM Agent valida se as instâncias do EC2 têm o valor de tag `DEV` aplicado ao grupo de patches e consulta o Patch Manager em busca de uma linha de base de patch associada.

2. O Patch Manager verifica se a linha de base do patch `pb-9876543210abcdef0` tem o grupo de patches `DEV` associado e notifica o SSM Agent.
3. O SSM Agent recupera um snapshot de linha de base de patch do Patch Manager com base nas regras de aprovação e nas exceções configuradas em `pb-9876543210abcdef0` e prossegue para a próxima etapa.

- Nenhuma tag de grupo de patches adicionada à instância:

1. O SSM Agent, que está instalado em instâncias do EC2 no grupo 2, recebe o comando emitido na Etapa 1 para iniciar uma operação de aplicação de patch. O SSM Agent valida se as instâncias do EC2 não têm uma tag `Patch Group` ou `PatchGroup` aplicada e, como resultado, o SSM Agent consulta o Patch Manager em busca da lista de referência de patches do Windows.
2. O Patch Manager verifica se a lista de referência de patches padrão do Windows Server é `pb-0123456789abcdef0` e notifica o SSM Agent.
3. O SSM Agent recupera um snapshot de linha de base de patch do Patch Manager com base nas regras de aprovação e nas exceções configuradas na linha de base de patch padrão `pb-0123456789abcdef0` e prossegue para a próxima etapa.

- Nenhum valor de grupo de patches correspondente associado a uma linha de base de patch:

1. O SSM Agent, que está instalado nas instâncias do EC2 no grupo 3, recebe o comando emitido na Etapa 1 para iniciar uma operação de aplicação de patch. O SSM Agent valida se as instâncias do EC2 têm o valor de tag `QA` aplicado ao grupo de patches e consulta o Patch Manager em busca de uma linha de base de patch associada.
2. O Patch Manager não encontra uma linha de base do patch que tenha o grupo de patches `QA` associado.
3. O Patch Manager notifica o SSM Agent para usar a linha de base de patch padrão do Windows, `pb-0123456789abcdef0`.
4. O SSM Agent recupera um snapshot de linha de base de patch do Patch Manager com base nas regras de aprovação e nas exceções configuradas na linha de base de patch padrão `pb-0123456789abcdef0` e prossegue para a próxima etapa.

3. Verificação ou instalação de patches: depois de determinar a linha de base de patch apropriada para uso, o SSM Agent começa a sondar ou instalar patches com base no valor de operação especificado na Etapa 1. Os patches sondados ou instalados são determinados pelas regras de aprovação e pelas exceções de patches definidas no snapshot de lista de referência de patches fornecido pelo Patch Manager.

Mais informações

- Valores de estados de conformidade de patches

Aplicações de patches lançadAs pela Microsoft no Windows Server

Use as informações neste tópico para ajudar você a preparar a aplicação de patches no Windows Server usando o Patch Manager, uma ferramenta do AWS Systems Manager.

Aplicação de patches

O suporte à aplicação de patches em instâncias gerenciadas do Windows Server é limitado a aplicações lançadas pela Microsoft.

> **Note**
>
> Em alguns casos, a Microsoft lança patches para aplicações que não especificam data e hora atualizadas. Nesses casos, uma data e hora atualizadas de `01/01/1970` são fornecidas por padrão.

Lista de referência de patches para aplicações de patches lançados pela Microsoft.

Para o Windows Server, três linhas de base de patch predefinidas são fornecidas. As listas de referência de patches `AWS-DefaultPatchBaseline` e `AWS-WindowsPredefinedPatchBaseline-OS` oferecem suporte apenas às atualizações do sistema operacional Windows em si. O `AWS-DefaultPatchBaseline` é usado como lista de referência de patches para nós gerenciados do Windows Server, a menos que você especifique outra lista de referência de patches. As definições de configuração nestas duas linhas de base de patch são as mesmas. O mais novo dos dois,`AWS-WindowsPredefinedPatchBaseline-OS`, foi criado para distingui-lo da terceira linha de base de patch predefinida paraWindows Server. Essa lista de referência do patch, `AWS-WindowsPredefinedPatchBaseline-OS-Applications`, pode ser usada para aplicar patches tanto ao sistema operacional Windows Server quanto a aplicações compatíveis lançadas pela Microsoft.

Você também pode criar uma lista de referência de patch personalizada para atualizar aplicações da Microsoft em máquinas do Windows Server.

Compatibilidade com a aplicação de patches em aplicativos lançados pela Microsoft em servidores on-premises, dispositivos de borda, VMs e outros nós que não pertençam ao EC2

Para aplicar patches em aplicações lançadas pela Microsoft em máquinas virtuais (VMs) e outros gerenciados que não são do EC2, é necessário ativar o nível de instâncias avançadas. Há uma cobrança para o uso do nível de instâncias avançadas. No entanto, não há custo adicional para aplicar patches em aplicativos lançados pela Microsoft em instâncias do Amazon Elastic Compute Cloud (Amazon EC2). Para obter mais informações, consulte Configurar níveis de instâncias.

Opção de atualização do Windows para “outros produtos da Microsoft”

Para que o Patch Manager possa aplicar patches em aplicações lançadas pela Microsoft nos nós gerenciados pelo Windows Server, a opção Give me updates for other Microsoft products when I update Windows (Fornecer atualizações para outros produtos Microsoft quando atualizo o Windows), para a atualização do Windows, deverá estar ativada em seu nó gerenciado.

Para obter informações sobre como permitir essa opção em um único nó gerenciado, consulte Update Office with Microsoft Update (Atualizar o Office com o Microsoft Update) no site de Suporte da Microsoft.

Para uma frota de nós gerenciados que executam o Windows Server 2016 e posterior, você pode usar um objeto de política de grupo (GPO) para ativar a configuração. No editor de gerenciamento de políticas de grupo, acesse Configuração do computador, Templates Administrativos, Componentes do Windows, Atualizações do Windows e escolha Instalar atualizações para outros produtos da Microsoft. Recomendamos também configurar o GPO com parâmetros adicionais que impedem atualizações automáticas não planejadas e reinicializações fora doPatch Manager. Para obter mais informações, consulte Configurar atualizações automáticas em um ambiente não Active Directory, no site de documentação técnica da Microsoft.

Para uma frota de nós gerenciados que executam o Windows Server 2012 ou 2012 R2, você pode ativar a opção usando um script, conforme descrito em Enabling and Disabling Microsoft Update in Windows 7 via Script (Ativar e desativar o Microsoft Update no Windows 7 via Script) no site do Blog do Microsoft Docs. Por exemplo, você pode fazer o seguinte:

1. Salve o script da postagem do blog em um arquivo.
2. Faça upload do arquivo para um bucket do Amazon Simple Storage Service (Amazon S3) ou outro local acessível.
3. Usar o Run Command, uma ferramenta do AWS Systems Manager, para executar o script em seus nós gerenciados usando o documento do Systems Manager (documento do SSM) `AWS-RunPowerShellScript` com um comando semelhante ao mostrado a seguir.

```
Invoke-WebRequest `
```

```
    -Uri "https://s3.aws-api-domain/amzn-s3-demo-bucket/script.vbs" `
    -Outfile "C:\script.vbs" cscript c:\script.vbs
```

Requisitos mínimos de parâmetros

Para incluir aplicações da Microsoft em sua lista de referência do patch personalizada, especifique pelo menos o produto ao qual deseja aplicar os patches. O comando da AWS Command Line Interface (AWS CLI) a seguir demonstra os requisitos mínimos para aplicar patch a um produto, como o Office 2016.

Linux & macOS

```
aws ssm create-patch-baseline \
    --name "My-Windows-App-Baseline" \
    --approval-rules
 "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=PRODUCT,Values='Office 2016'},
{Key=PATCH_SET,Values='APPLICATION'}]},ApproveAfterDays=5}]"
```

Windows Server

```
aws ssm create-patch-baseline ^
    --name "My-Windows-App-Baseline" ^
    --approval-rules
 "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=PRODUCT,Values='Office 2016'},
{Key=PATCH_SET,Values='APPLICATION'}]},ApproveAfterDays=5}]"
```

Se você especificar a família de produtos de aplicativos Microsoft, cada produto especificado deverá ser compatível com o membro da família de produtos selecionada. Por exemplo, para aplicar patch ao produto "Active Directory Rights Management Services Client 2.0", você deve especificar sua família de produtos como "Active Directory" e não, por exemplo, "Office" ou "SQL Server". O seguinte comando da AWS CLI demonstra um emparelhamento de correspondência entre família e produto:

Linux & macOS

```
aws ssm create-patch-baseline \
    --name "My-Windows-App-Baseline" \
    --approval-rules
 "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=PRODUCT_FAMILY,Values='Active
```

```
 Directory'},{Key=PRODUCT,Values='Active Directory Rights Management Services Client
 2.0'},{Key=PATCH_SET,Values='APPLICATION'}]},ApproveAfterDays=5}]"
```

Windows Server

```
aws ssm create-patch-baseline ^
    --name "My-Windows-App-Baseline" ^
    --approval-rules
 "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=PRODUCT_FAMILY,Values='Active
 Directory'},{Key=PRODUCT,Values='Active Directory Rights Management Services Client
 2.0'},{Key=PATCH_SET,Values='APPLICATION'}]},ApproveAfterDays=5}]"
```

> **Note**
>
> Se você receber uma mensagem de erro sobre um emparelhamento de produto e família incompatíveis, consulteProblema: família de produtos/pares de produtos incompatíveispara obter ajuda para resolver o problema.

Usar o Kernel Live Patching em nós gerenciados do Amazon Linux 2

O Kernel Live Patching para Amazon Linux 2 permite que você aplique patches para vulnerabilidades de segurança e erros críticos a um kernel do Linux em execução, sem reinicializações ou interrupções às aplicações em execução. Isso permite que você aproveite a maior disponibilidade de serviços e aplicações, mantendo sua infraestrutura segura e atualizada. O Kernel Live Patching é compatível com instâncias do Amazon EC2, com dispositivos principais do AWS IoT Greengrass e com máquinas virtuais on-premises que executam o Amazon Linux 2.

Para obter informações gerais sobre o Kernel Live Patching, consulte Kernel Live Patching no AL2 no Guia do usuário do Amazon Linux 2.

Depois de ativar Kernel Live Patching em um nó gerenciado do Amazon Linux 2, você poderá usar o Patch Manager, uma ferramenta do AWS Systems Manager, para aplicar patches dinâmicos do kernel ao nó gerenciado. Usar o Patch Manager é uma alternativa ao uso de fluxos de trabalho do yum existentes em seu nó para aplicar as atualizações.

Antes de começar

Para usar o Patch Manager para aplicar patches dinâmicos do kernel aos nós gerenciados do Amazon Linux 2, verifique se os nós são baseados na arquitetura e na versão do kernel corretas.

Para obter mais informações, consulte Configurações e pré-requisitos compatíveis no Guia do usuário do Amazon EC2.

Tópicos

- Kernel Live Patching usando o Patch Manager
- Como funciona o Patch Manager usando o Kernel Live Patching
- Ativar o Kernel Live Patching usando Run Command
- Aplicar patches dinâmicos do kernel usando o Run Command
- Desativar o Kernel Live Patching usando Run Command

Kernel Live Patching usando o Patch Manager

Atualizar a versão do kernel

Não é necessário reinicializar um nó gerenciado depois de aplicar uma atualização de patch dinâmico do kernel. No entanto, a AWS fornece patches dinâmicos do kernel para uma versão do kernel do Amazon Linux 2 por até três meses após seu lançamento. Após o período de três meses, é necessário fazer a atualização para uma versão posterior do kernel para continuar a receber patches dinâmicos do kernel. Recomendamos usar uma janela de manutenção para agendar uma reinicialização do nó pelo menos uma vez a cada três meses para solicitar a atualização da versão do kernel.

Desinstalar patches dinâmicos do kernel

Os patches dinâmicos do kernel não podem ser desinstalados usando o Patch Manager. Em vez disso, é possível desabilitar o Kernel Live Patching, o que remove os pacotes RPM para os patches dinâmicos do kernel aplicados. Para ter mais informações, consulte Desativar o Kernel Live Patching usando Run Command.

Conformidade com o kernel

Em alguns casos, instalar todas as correções CVE dos patches dinâmicos para a versão atual do kernel pode trazer esse kernel para o mesmo estado de conformidade de uma versão mais recente do kernel. Quando isso acontece, a versão mais recente é relatada como `Installed` e o nó gerenciado é relatado como `Compliant`. No entanto, nenhum tempo de instalação é relatado para a versão mais recente do kernel.

Um patch dinâmico do kernel, vários CVEs

Se um patch dinâmico do kernel aborda vários CVEs e estes tiverem vários valores de classificação e gravidade, somente a classificação e a gravidade mais altas entre os CVEs serão relatadas ao patch.

O restante desta seção descreve como usar o Patch Manager para aplicar patches dinâmicos do kernel aos nós gerenciados que atendem a esses requisitos.

Como funciona o Patch Manager usando o Kernel Live Patching

A AWS lança dois tipos de patches dinâmicos do kernel para o Amazon Linux 2: atualizações de segurança e correções de bugs. Para aplicar esses tipos de patches, use um documento de linha de base de patch que visa somente as classificações e severidades listadas na tabela a seguir.

Classificação	Gravidade
`Security`	`Critical`, `Important`
`Bugfix`	`All`

É possível criar uma lista de referência de patches personalizada destinada apenas a esses patches ou usar a lista de referência de patches `AWS-AmazonLinux2DefaultPatchBaseline` predefinida. Em outras palavras, é possível usar o `AWS-AmazonLinux2DefaultPatchBaseline` com nós gerenciados do Amazon Linux 2 nos quais o Kernel Live Patching estiver ativado, e as atualizações ao vivo do kernel serão aplicadas.

> **Note**
>
> A configuração `AWS-AmazonLinux2DefaultPatchBaseline` especifica um período de espera de sete dias após o lançamento ou última atualização de um patch antes que ele seja instalado automaticamente. Se você não quiser aguardar sete dias para que os patches dinâmicos do kernel sejam aprovados automaticamente, é possível criar e usar uma lista de referência de patches personalizada. Na linha de base do patch, você pode especificar nenhum período de espera de aprovação automática ou especificar um período mais curto ou mais longo. Para ter mais informações, consulte Trabalhando com linhas de base de patch personalizadas.

Recomendamos a seguinte estratégia para aplicar patches aos nós gerenciados com atualizações ao vivo do kernel:

1. Ativar o Kernel Live Patching em nós gerenciados do Amazon Linux 2.
2. Use o Run Command, uma ferramenta do AWS Systems Manager, para executar uma operação do `Scan` em seus nós gerenciados, usando a lista de referência de patches `AWS-AmazonLinux2DefaultPatchBaseline` predefinida ou a personalizada que também é destinada somente às atualizações `Security` classificadas como `Critical` e `Important` e a gravidade `Bugfix` de `All`.
3. Use o Compliance, uma ferramenta do AWS Systems Manager para verificar se uma não conformidade para aplicação de patches é relatada para qualquer um dos nós gerenciados que foram verificados. Em caso afirmativo, visualize os detalhes de conformidade do nó para determinar se algum patch dinâmico do kernel está ausente em seu nó gerenciado.
4. Para instalar patches dinâmicos do kernel ausentes, use o Run Command com a mesma lista de referência de patches especificada anteriormente, mas desta vez execute uma operação `Install` em vez de uma operação `Scan`.

   Como os patches dinâmicos do kernel são instalados sem a necessidade de reinicialização, é possível escolher a opção de reinicialização `NoReboot` para esta operação.

   > **Note**
   >
   > Você pode ainda reinicializar o nó gerenciado se necessário para outros tipos de patches instalados nele ou se quiser atualizar para um kernel mais recente. Nesses casos, escolha a opção de reinicialização `RebootIfNeeded`.

5. Retorne à conformidade do para verificar se os patches dinâmicos do kernel foram instalados.

Ativar o Kernel Live Patching usando Run Command

Para ativar o Kernel Live Patching, você pode executar o `yum` em seus nós gerenciados ou usar o Run Command e um documento personalizado do Systems Manager (documento SSM) que você criar.

Para obter informações sobre como ativar o Kernel Live Patching executando comandos `yum` diretamente em seu nó gerenciado, consulte Habilitar o Kernel Live Patching no Guia do usuário do Amazon EC2.

> **Note**
>
> Quando você ativa o Kernel Live Patching, se o kernel já executado em seu nó gerenciado for anterior ao `kernel-4.14.165-131.185.amzn2.x86_64` (a versão mínima suportada), o processo instala a versão mais recente do kernel disponível e reinicializa o nó gerenciado. Se o nó já estiver executando o `kernel-4.14.165-131.185.amzn2.x86_64` ou posterior, o processo não instalará uma versão mais recente e não reinicializará o nó.

Para ativar oKernel Live PatchingusandoRun Command(console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Run Command.
3. Selecione Run command.
4. No Documento de comando, escolha o documento SSM personalizado`AWS-ConfigureKernelLivePatching`.
5. Na seção Command parameters (Parâmetros de comando), especifique se deseja que os nós gerenciados sejam reinicializados como parte desta operação.
6. Para obter informações sobre como trabalhar com os controles restantes nesta página, consulte Executar comandos no console.
7. Escolha Executar.

Para ativar Kernel Live Patching (AWS CLI)

- Execute o seguinte comando na máquina local.

  Linux & macOS

  ```
  aws ssm send-command \
      --document-name "AWS-ConfigureKernelLivePatching" \
      --parameters "EnableOrDisable=Enable" \
      --targets "Key=instanceids,Values=instance-id"
  ```

  Windows Server

  ```
  aws ssm send-command ^
      --document-name "AWS-ConfigureKernelLivePatching" ^
  ```

```
    --parameters "EnableOrDisable=Enable" ^
    --targets "Key=instanceids,Values=instance-id"
```

Substitua *instance-id* pelo ID do nó gerenciado do Amazon Linux 2 na qual você deseja ativar o recurso, como i-02573cafcfEXAMPLE. Para ativar o recurso em vários nós gerenciados, é possível usar um dos formatos a seguir.

- `--targets "Key=instanceids,Values=instance-id1,instance-id2"`
- `--targets "Key=tag:tag-key,Values=tag-value"`

Para obter informações sobre outras opções que você pode usar no comando, consulte send-command na AWS CLI Command Reference.

Aplicar patches dinâmicos do kernel usando o Run Command

Para aplicar patches dinâmicos do kernel, é possível executar os comandos do `yum` em seus nós gerenciados ou usar o Run Command e o documento `AWS-RunPatchBaseline` do SSM.

Para obter informações sobre como aplicar patches dinâmicos do kernel executando os comandos do `yum` diretamente em seu nó gerenciado, consulte Aplicar patches dinâmicos do kernel no Guia do usuário do Amazon EC2.

Como aplicar patches dinâmicos do kernel usando o Run Command (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Run Command.
3. Selecione Run command.
4. Na lista Documento do comando, escolha o documento `AWS-RunPatchBaseline` do SSM.
5. Na seção Parâmetros de comando, siga um destes procedimentos:

   - Se você estiver verificando se novos patches dinâmicos do kernel estão disponíveis, em Operação, escolha `Scan`. Em Reboot Option (Opção de reinicialização), se você não quiser que os nós gerenciados sejam reinicializados após essa operação, escolha `NoReboot`. Após a conclusão da operação, será possível verificar se há novos patches e status de conformidade em Compliance (Conformidade).

- Se você já verificou a conformidade de patches e está pronto para aplicar patches dinâmicos do kernel disponíveis, em Operação, escolha `Install`. Em Reboot Option (Opção de reinicialização), se não quiser que os nós gerenciados sejam reinicializados após essa operação, escolha `NoReboot`.

6. Para obter informações sobre como trabalhar com os controles restantes nesta página, consulte Executar comandos no console.
7. Escolha Executar.

Como aplicar patches dinâmicos do kernel usando o Run Command (AWS CLI)

1. Para executar uma operação `Scan` antes de verificar os resultados no Compliance, execute o comando a seguir na máquina local.

   Linux & macOS

   ```
   aws ssm send-command \
       --document-name "AWS-RunPatchBaseline" \
       --targets "Key=InstanceIds,Values=instance-id" \
       --parameters '{"Operation":["Scan"],"RebootOption":["RebootIfNeeded"]}'
   ```

   Windows Server

   ```
   aws ssm send-command ^
       --document-name "AWS-RunPatchBaseline" ^
       --targets "Key=InstanceIds,Values=instance-id" ^
       --parameters {\"Operation\":[\"Scan\"],\"RebootOption\":[\"RebootIfNeeded
   \"]}
   ```

   Para obter informações sobre outras opções que você pode usar no comando, consulte send-command na AWS CLI Command Reference.

2. Para executar uma operação `Install` após verificar os resultados no Compliance, execute o comando a seguir na máquina local.

   Linux & macOS

   ```
   aws ssm send-command \
       --document-name "AWS-RunPatchBaseline" \
   ```

```
    --targets "Key=InstanceIds,Values=instance-id" \
    --parameters '{"Operation":["Install"],"RebootOption":["NoReboot"]}'
```

Windows Server

```
aws ssm send-command ^
    --document-name "AWS-RunPatchBaseline" ^
    --targets "Key=InstanceIds,Values=instance-id" ^
    --parameters {\"Operation\":[\"Install\"],\"RebootOption\":[\"NoReboot\"]}
```

Nos comandos anteriores, substitua *instance-id* pelo ID do nó gerenciado do Amazon Linux 2 no qual você deseja aplicar patches dinâmicos do kernel, como i-02573cafcfEXAMPLE. Para ativar o recurso em vários nós gerenciados, é possível usar um dos formatos a seguir.

- `--targets "Key=instanceids,Values=instance-id1,instance-id2"`
- `--targets "Key=tag:tag-key,Values=tag-value"`

Para obter informações sobre outras opções que você pode usar nesses comandos, consulte send-command na AWS CLI Command Reference.

Desativar o Kernel Live Patching usando Run Command

Para habilitar o Kernel Live Patching, é possível executar os comandos do `yum` em seus nós gerenciados ou usar o Run Command e um documento personalizado do `AWS-ConfigureKernelLivePatching` criado por você.

> **Note**
>
> Se não precisar mais usar o Kernel Live Patching, você pode desabilitá-lo a qualquer momento. Na maioria dos casos, não é necessário desativar o recurso.

Para obter informações sobre como desativar o Kernel Live Patching executando comandos `yum` diretamente em seu nó gerenciado, consulte Habilitar o Kernel Live Patching no Guia do usuário do Amazon EC2.

> **Note**
>
> Quando você desativa o Kernel Live Patching, o processo desinstala o plugin Kernel Live Patching e, em seguida, reinicializa o nó gerenciado.

Para desativar oKernel Live PatchingusandoRun Command(console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Run Command.
3. Selecione Run command.
4. Na lista Documento do comando, escolha o documento `AWS-ConfigureKernelLivePatching` do SSM.
5. Na seção Command parameters, especifique valores para os parâmetros necessários.
6. Para obter informações sobre como trabalhar com os controles restantes nesta página, consulte Executar comandos no console.
7. Escolha Executar.

Para desativar oKernel Live Patching(AWS CLI)

- Execute um comando semelhante ao seguinte.

  Linux & macOS

  ```
  aws ssm send-command \
      --document-name "AWS-ConfigureKernelLivePatching" \
      --targets "Key=instanceIds,Values=instance-id" \
      --parameters "EnableOrDisable=Disable"
  ```

  Windows Server

  ```
  aws ssm send-command ^
      --document-name "AWS-ConfigureKernelLivePatching" ^
      --targets "Key=instanceIds,Values=instance-id" ^
      --parameters "EnableOrDisable=Disable"
  ```

Substitua *instance-id* pelo ID do nó gerenciado do Amazon Linux 2 na qual você quer desativar o recurso, como i-02573cafcfEXAMPLE. Para desativar o recurso em vários nós gerenciados, é possível usar um dos formatos a seguir.

- `--targets "Key=instanceids,Values=instance-id1,instance-id2"`
- `--targets "Key=tag:tag-key,Values=tag-value"`

Para obter informações sobre outras opções que você pode usar no comando, consulte send-command na AWS CLI Command Reference.

Trabalhar com recursos e conformidade do Patch Manager usando o console

Para usar o Patch Manager, uma ferramenta do AWS Systems Manager, conclua as tarefas a seguir. Essas tarefas estão descritas com mais detalhes nesta seção.

1. Verifique se a linha de base de patch predefinida da AWS para cada tipo de sistema operacional que você usa atende às suas necessidades. Se não atender, crie uma lista de referência de patches que defina um conjunto padrão de patches para esse tipo de nó gerenciado e a defina como o padrão.
2. Organize os nós gerenciados em grupos de patches usando etiquetas do Amazon Elastic Compute Cloud (Amazon EC2) (opcional, mas recomendado).
3. Execute um destes procedimentos:
   - (Recomendado) Configure uma política de patch no Quick Setup, uma ferramenta do Systems Manager que permite instalar patches que estão faltando de acordo com uma agenda para uma organização inteira, um subconjunto de unidades organizacionais ou uma única Conta da AWS. Para ter mais informações, consulte Configurar a aplicação de patches para instâncias em uma organização usando uma política de patch do Quick Setup.
   - Crie uma janela de manutenção que use o documento Systems Manager (SSM document) `AWS-RunPatchBaseline` em um tipo de tarefa Run Command. Para ter mais informações, consulte Tutorial: Create a maintenance window for patching using the console.
   - Executar manualmente`AWS-RunPatchBaseline`em umRun Commandoperação. Para ter mais informações, consulte Executar comandos no console.
   - Aplique manualmente os patches nos nós sob demanda usando o recurso Patch now. Para ter mais informações, consulte Aplicação de patches em nós gerenciados sob demanda.

4. Monitore a aplicação de patches para verificar a conformidade e investigar falhas.

Tópicos

- Criação de uma política de patch
- Visualizar resumos do painel de patches
- Trabalhando com relatórios de conformidade de patch
- Aplicação de patches em nós gerenciados sob demanda
- Trabalhar com linhas de base de patches
- Visualizar patches disponíveis
- Como criar e gerenciar grupos de patches
- Integrar o Patch Manager ao AWS Security Hub

Criação de uma política de patch

Uma política de patch é uma configuração que você define usando o Quick Setup, uma ferramenta do AWS Systems Manager. As políticas de patch fornecem um controle mais amplo e centralizado sobre suas operações de aplicação de patches do que o disponível com os outros métodos de configuração de patches. Uma política de patch define a programação e a lista de referência de patches a serem usados na aplicação automática de patches nos seus nós gerenciados.

Para obter mais informações, consulte os tópicos a seguir.

- Configurações de políticas de patches em Quick Setup
- Configurar a aplicação de patches para instâncias em uma organização usando uma política de patch do Quick Setup

Visualizar resumos do painel de patches

A guia Painel no Patch Manager oferece uma visualização de resumo no console que você pode usar para monitorar suas operações de aplicação de patches em uma visualização consolidada. O Patch Manager é uma ferramenta do AWS Systems Manager. Na guia Dashboard (Painel), você pode visualizar o seguinte:

- Um snapshot de quantos nós gerenciados são ou não são compatíveis com as regras de aplicação de patches.

- Um snapshot de quando os resultados de conformidade dos patches para os nós gerenciados foram gerados.
- Uma contagem vinculada de quantos nós gerenciados não compatíveis existem para cada um dos motivos mais comuns para a não conformidade.
- Uma lista vinculada das operações de patch mais recentes.
- Uma lista vinculada das tarefas de patch recorrentes que foram configuradas.

Para visualizar resumos do painel de patches

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Patch Manager.
3. Selecione a guia Dashboard (Painel).
4. Role até a seção que contém dados de resumo que você deseja visualizar:
    - Gerenciamento de instâncias do Amazon EC2
    - Resumo de conformidade
    - Contagens de não conformidade
    - Relatórios de conformidade
    - Operações baseadas em políticas que não sejam de patches
    - Tarefas recorrentes baseadas em políticas que não sejam de patches

Trabalhando com relatórios de conformidade de patch

Use as informações nos tópicos a seguir para ajudar você a gerar e trabalhar com relatórios de conformidade de patches no Patch Manager, uma ferramenta do AWS Systems Manager.

As informações nos tópicos a seguir se aplicam independentemente do método ou tipo de configuração que você estiver usando para suas operações de aplicação de patch:

- Uma política de patch configurada no Quick Setup
- Uma opção do Host Management configurada no Quick Setup
- Uma janela de manutenção para executar um patch `Scan` ou tarefa `Install`
- Uma operação Patch now (Aplicar patch agora) sob demanda

> **Important**
>
> Se você tiver vários tipos de operações em andamento para verificar a conformidade dos patches em suas instâncias, observe que cada verificação substitui os dados de conformidade de patches de verificações anteriores. Como resultado, você pode obter resultados inesperados em seus dados de conformidade de patch. Para ter mais informações, consulte Prevenção de substituições não intencionais de dados de conformidade de patches.
> Para verificar qual lista de referência de patches foi usada para gerar as informações de conformidade mais recentes, navegue até a guia Relatórios de conformidade no Patch Manager, localize a linha do nó gerenciado sobre o qual deseja obter informações e escolha o ID da lista de referência na coluna ID da lista de referência usada.

Tópicos

- Visualizar resultados de conformidade de patches
- Gere relatórios .csv de conformidade de patches
- Corrigir nós gerenciados fora de conformidade com o Patch Manager
- Prevenção de substituições não intencionais de dados de conformidade de patches

Visualizar resultados de conformidade de patches

Use esses procedimentos para exibir as informações de conformidade de patches sobre seus nós gerenciados.

Este procedimento se aplica a operações de patch que usam o`AWS-RunPatchBaseline`document. Para obter mais informações sobre como visualizar informações de conformidade de patches para operações de patch que usam o documento `AWS-RunPatchBaselineAssociation`, consulte Identificar nós gerenciados fora de conformidade.

> **Note**
>
> As operações de digitalização de patches para Quick Setup e Explorer usam o documento `AWS-RunPatchBaselineAssociation`. Quick Setup e Explorer são ambos ferramentas do AWS Systems Manager.

Identificar a solução de patch para um problema específico de CVE (Linux)

Para muitos sistemas operacionais baseados em Linux, os resultados de conformidade de patch indicam quais problemas do boletim Common Vulnerabilities and Exposure (CVE) serão resolvidos por quais patches. Essas informações podem ajudar você a determinar com que urgência você precisa instalar um patch ausente ou com falha.

Os detalhes do CVE estão incluídos nas versões com suporte dos seguintes tipos de sistema operacional:

- AlmaLinux
- Amazon Linux 1
- Amazon Linux 2
- Amazon Linux 2022
- Amazon Linux 2023
- Oracle Linux
- Red Hat Enterprise Linux (RHEL)
- Rocky Linux
- SUSE Linux Enterprise Server (SLES)

> **Note**
>
> Por padrão, CentOS e CentOS Stream não fornecem informações do CVE sobre atualizações. No entanto, você pode permitir esse suporte usando repositórios de terceiros como o repositório Extra Packages for Enterprise Linux (EPEL) publicado pelo Fedora. Para obter mais informações, consulte EPEL no Fedora Wiki.
> No momento, os valores de ID de CVE são reportados somente para patches com um status de `Missing` ou `Failed`.

Você também pode adicionar IDs de CVE às listas de patches aprovados ou rejeitados em suas linhas de base de patch, conforme a situação e suas metas de patch o justificarem.

Para obter mais informações sobre como trabalhar com listas de patches aprovados e rejeitados, consulte os seguintes tópicos:

- Trabalhando com linhas de base de patch personalizadas

- Sobre formatos de nomes de pacotes para listas de patches aprovados e rejeitados
- Como funcionam as regras de linha de base de patch em sistemas baseados no Linux
- Como os patches são instalados

> **Note**
>
> Em alguns casos, a Microsoft lança patches para aplicações que não especificam data e hora atualizadas. Nesses casos, uma data e hora atualizadas de `01/01/1970` são fornecidas por padrão.

Visualizar resultados de conformidade de patches

Use o procedimento a seguir para visualizar dados de conformidade do patch no console do AWS Systems Manager.

> **Note**
>
> Para obter informações sobre como gerar relatórios de conformidade de patches que foram baixados em um bucket do Amazon Simple Storage Service (Amazon S3), consulte Gere relatórios .csv de conformidade de patches.

Para visualizar resultados de conformidade de patch do

1. Execute um destes procedimentos:

   Opção 1 (recomendada): navegar desde o Patch Manager, uma ferramenta do AWS Systems Manager:

   - No painel de navegação, escolha Patch Manager.
   - Selecione a guia Compliance reporting (Relatório de conformidade).
   - Na área Detalhes de patch do nó, escolha o ID de nó do nó gerenciado para o qual deseja analisar os resultados da conformidade de patches.
   - Na área Detalhes, na lista Propriedades, escolha Patches.

   Opção 2: navegar desde o Compliance, uma ferramenta do AWS Systems Manager:

- No painel de navegação, selecione Compliance (Conformidade).
- Para Compliance resources summary (Resumo dos recursos de conformidade), escolha um número na coluna para os tipos de recursos de patch que você deseja revisar, como Non-Compliant resources (Recursos sem conformidade).
- Na lista Recurso abaixo, escolha o ID do nó gerenciado do qual você deseja rever os resultados de conformidade do patch.
- Na área Detalhes, na lista Propriedades, escolha Patches.

Opção 3: navegar desde o Fleet Manager, uma ferramenta do AWS Systems Manager.

- No painel de navegação, escolha Fleet Manager.
- Na área Instâncias gerenciadas, escolha o ID do nó gerenciado do qual você deseja rever os resultados da conformidade do patch.
- Na área Detalhes, na lista Propriedades, escolha Patches.

2. (Opcional) Na caixa de pesquisa (), escolha um ou mais dos filtros disponíveis.

   Por exemplo, para o Red Hat Enterprise Linux (RHEL), escolha uma das seguintes opções:

   - Name
   - Classificação
   - Estado
   - Gravidade

   Para o Windows Server, escolha uma das seguintes opções:

   - KB
   - Classificação
   - Estado
   - Gravidade

3. Escolha um dos valores disponíveis para o tipo de filtro escolhido. Por exemplo, se você escolheu State (Estado), agora escolha um estado de conformidade como InstalledPendingReboot, Failed (Com falha) ou Missing (Ausente).

> **Note**
>
> No momento, os valores de ID de CVE são reportados somente para patches com um status de `Missing` ou `Failed`.

4. Dependendo do estado de conformidade do nó gerenciado, você poderá escolher qual ação tomar para corrigir qualquer nó não compatível.

   Por exemplo, você pode optar por corrigir os nós gerenciados não compatíveis imediatamente. Para obter informações sobre a aplicação de patches em nós gerenciados sob demanda, consulte Aplicação de patches em nós gerenciados sob demanda.

   Para obter informações sobre dados de conformidade dos patches, consulte Valores de estados de conformidade de patches.

Gere relatórios .csv de conformidade de patches

Você pode usar o console do AWS Systems Manager para gerar relatórios de conformidade de patch que são salvos como um arquivo.csv em um bucket do Amazon Simple Storage Service (Amazon S3) de sua escolha. Você pode gerar um único relatório sob demanda ou especificar uma programação para gerar os relatórios automaticamente.

Os relatórios podem ser gerados para um único nó gerenciado ou para todos os nós gerenciados na sua Conta da AWS e Região da AWS selecionadas. Para um único nó, um relatório contém detalhes abrangentes, incluindo os IDs de patches relacionados a um nó que não esteja compatível. Para obter um relatório sobre todos os nós gerenciados, apenas informações resumidas e contagens de patches de nós não compatíveis são fornecidas.

Depois que um relatório é gerado, você pode usar uma ferramenta, como o Amazon QuickSight, para importar e analisar os dados. O QuickSight é um serviço de business intelligence (BI) que você pode usar para explorar e interpretar informações em um ambiente visual interativo. Para obter mais informações, consulte o Manual do usuário do Amazon QuickSight.

> **Note**
>
> Ao criar uma linha de lista de referência personalizada, é possível especificar um nível de gravidade de conformidade para patches aprovados por essa lista de referência de patches, como `Critical` ou `High`. Se o estado do patch de qualquer patch aprovado for relatado

como `Missing`, a gravidade geral da conformidade relatada pela lista de referência de patches será o nível de gravidade especificado.

Você também pode especificar um tópico do Amazon Simple Notification Service (Amazon SNS) a ser usado para enviar notificações quando um relatório for gerado.

Funções de serviço para gerar relatórios de conformidade de patch

Na primeira vez que você gera um relatório, o Systems Manager cria um perfil de admissão do Automation chamado `AWS-SystemsManager-PatchSummaryExportRole` para usar com o processo de exportação para o S3.

Note

Se você estiver exportando dados de conformidade para um bucket criptografado do S3, será necessário atualizar a política de chave do AWS KMS associada para fornecer as permissões necessárias para `AWS-SystemsManager-PatchSummaryExportRole`. Por exemplo, adicione uma permissão semelhante a essa à política do AWS KMS de seu bucket do S3:

```
{
    "Effect": "Allow",
    "Action": [
        "kms:GenerateDataKey"
    ],
    "Resource": "role-arn"
}
```

Substitua *`role-arn`* pelo nome do recurso da Amazon (ARN) criado em sua conta no formato `arn:aws:iam::111222333444:role/service-role/AWS-SystemsManager-PatchSummaryExportRole`.
Para obter mais informações, consulte Políticas de chaves no AWS KMS no Guia do desenvolvedor do AWS Key Management Service.

Na primeira vez que você gerar um relatório em um agendamento, o Systems Manager cria outra função de serviço chamada`AWS-EventBridge-Start-SSMAutomationRole`, juntamente com a função de serviço`AWS-SystemsManager-PatchSummaryExportRole`(se ainda

não tiver sido criado) para usar para o processo de exportação.`AWS-EventBridge-Start-SSMAutomationRole`permite que o Amazon EventBridge inicie uma automação usando o runbookAWS-ExportPatchReporttos3.

Recomendamos que não tente modificar essas políticas e funções. Isso pode causar falha na geração do relatório de conformidade do patch. Para obter mais informações, consulte Solução de problemas da geração de relatórios de conformidade de patches.

Tópicos

- O que está em um relatório de conformidade de patch gerado?
- Gerar relatórios de conformidade de patch para um único nó gerenciado.
- Gerar relatórios de conformidade de patch para todos os nós gerenciados.
- Visualizar histórico de relatórios de conformidade de patches
- Visualizar as programações de relatórios de conformidade de patches
- Solução de problemas da geração de relatórios de conformidade de patches

O que está em um relatório de conformidade de patch gerado?

Este tópico fornece informações sobre os tipos de conteúdo incluídos nos relatórios de conformidade de patch que são gerados e baixados para um bucket do S3 especificado.

Formato de relatório para um único nó gerenciado

Um relatório gerado para um único nó gerenciado fornece informações resumidas e detalhadas.

Baixar um relatório de exemplo (nó único)

Informações resumidas para um único nó gerenciado incluem o seguinte:

- Índice
- ID da instância
- Nome da instância
- Instance IP
- nome-da-plataforma
- Versão da plataforma

- Versão do SSM Agent
- Lista de referência de patches
- Grupo de patches
- Compliance status (Status de conformidade)
- Gravidade da conformidade
- Contagem de patches de severidade crítica não compatível
- Contagem de patches de alta gravidade não compatível
- Contagem de patches de gravidade média não compatível
- Contagem de patches de baixa gravidade não compatível
- Contagem de patches de severidade informativa não compatível
- Contagem de patches de gravidade não especificada não compatível

Informações detalhadas para um único nó gerenciado incluem o seguinte:

- Índice
- ID da instância
- Nome da instância
- Nome do patch
- ID do BC/ID do patch
- Estado do patch
- Hora do último relatório
- Nível de conformidade
- Gravidade do patch
- Classificação de patch
- CVE ID
- Lista de referência de patches
- URL de logs
- Instance IP
- nome-da-plataforma
- Versão da plataforma

- Versão do SSM Agent

> **Note**
>
> Ao criar uma linha de lista de referência personalizada, é possível especificar um nível de gravidade de conformidade para patches aprovados por essa lista de referência de patches, como `Critical` ou `High`. Se o estado do patch de qualquer patch aprovado for relatado como `Missing`, a gravidade geral da conformidade relatada pela lista de referência de patches será o nível de gravidade especificado.

Formato de relatório para todos os nós gerenciados

Um relatório gerado para todos os nós gerenciados fornece apenas informações resumidas.

Baixar um relatório de amostra (todos os nós gerenciados)

Informações resumidas para todos os nós gerenciados incluem o seguinte:

- Índice
- ID da instância
- Nome da instância
- Instance IP
- nome-da-plataforma
- Versão da plataforma
- Versão do SSM Agent
- Lista de referência de patches
- Grupo de patches
- Compliance status (Status de conformidade)
- Gravidade da conformidade
- Contagem de patches de severidade crítica não compatível
- Contagem de patches de alta gravidade não compatível
- Contagem de patches de gravidade média não compatível
- Contagem de patches de baixa gravidade não compatível

- Contagem de patches de severidade informativa não compatível
- Contagem de patches de gravidade não especificada não compatível

Gerar relatórios de conformidade de patch para um único nó gerenciado.

Use o procedimento a seguir para gerar um relatório de resumo de patches para um único nó gerenciado na sua Conta da AWS. O relatório de um único nó gerenciado fornece detalhes sobre cada patch que estiver fora de conformidade, incluindo nomes de patches e IDs.

Para gerar relatórios de conformidade de patch para um único nó gerenciado.

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Patch Manager.
3. Selecione a guia Compliance reporting (Relatório de conformidade).
4. Selecione o botão na linha do nó gerenciado para o qual você deseja gerar um relatório e escolha View detail (Visualizar detalhes).
5. Na seção Patch summary (Resumo do patch), escolha Export to S3 (Exportar para o S3).
6. Para Report name (Nome do relatório), insira um nome para ajudar você a identificar o relatório mais tarde.
7. Para Reporting frequency (Frequência dos relatórios), escolha uma das seguintes opções:
   - Sob demanda— Cria um relatório único. Vá para a etapa 9.
   - Em um horário— Especifique uma programação recorrente para gerar relatórios automaticamente. Continue na etapa 8.
8. Para Schedule type (Tipo de programação), especifique uma expressão de taxa, como a cada 3 dias, ou forneça uma expressão cron para definir a frequência do relatório.

   Para obter informações sobre expressões cron, consulte Referência: Expressões cron e rate para o Systems Manager.
9. Para o Bucket name (Nome do bucket), selecione o nome de um bucket do S3 onde você deseja armazenar os arquivos .csv de relatório

> **Important**
>
> Se você estiver trabalhando em uma Região da AWS que foi lançada após 20 de março de 2019, selecione um bucket do S3 na mesma região. As regiões iniciadas após essa

data foram desativadas por padrão. Para obter mais informações e uma lista dessas regiões, consulte Enabling a Region no Referência geral da Amazon Web Services.

10. (Opcional) Para enviar notificações quando o relatório for gerado, expanda a seção Tópico do SNS e escolha um tópico existente do Amazon SNS em SNS topic Amazon Resource Name (ARN) (Nome do recurso da Amazon (ARN) do tópico do SNS).
11. Selecione Enviar.

Para obter informações sobre como exibir um histórico de relatórios gerados, consulte Visualizar histórico de relatórios de conformidade de patches.

Para obter informações sobre como visualizar detalhes das agendas de relatórios que você criou, consulte Visualizar as programações de relatórios de conformidade de patches.

Gerar relatórios de conformidade de patch para todos os nós gerenciados.

Use o procedimento a seguir para gerar um relatório de resumo de patches para todos os nós gerenciados na sua Conta da AWS. O relatório de todos os nós gerenciados indica quais nós estão fora de conformidade e os números de patches fora de conformidade. Ele não fornece os nomes ou outros identificadores dos patches. Para esses detalhes adicionais, você pode gerar um relatório de conformidade de patches para um único nó gerenciado. Para obter mais informações, consulte Gerar relatórios de conformidade de patch para um único nó gerenciado. já abordado neste tópico.

Para gerar relatórios de conformidade de patch para todos os nós gerenciados

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Patch Manager.
3. Selecione a guia Compliance reporting (Relatório de conformidade).
4. Selecione Export to S3 (Exportar para o S3). (Não selecione um ID de nó primeiro.)
5. Para Report name (Nome do relatório), insira um nome para ajudar você a identificar o relatório mais tarde.
6. Para Reporting frequency (Frequência dos relatórios), escolha uma das seguintes opções:
   - Sob demanda— Cria um relatório único. Vá para a etapa 8.
   - Em um horário— Especifique uma programação recorrente para gerar relatórios automaticamente. Continue na etapa 7.

7. Para Schedule type (Tipo de programação), especifique uma expressão de taxa, como a cada 3 dias, ou forneça uma expressão cron para definir a frequência do relatório.

   Para obter informações sobre expressões cron, consulte Referência: Expressões cron e rate para o Systems Manager.
8. Para o Bucket name (Nome do bucket), selecione o nome de um bucket do S3 onde você deseja armazenar os arquivos .csv de relatório

   > **Important**
   >
   > Se você estiver trabalhando em uma Região da AWS que foi lançada após 20 de março de 2019, selecione um bucket do S3 na mesma região. As regiões iniciadas após essa data foram desativadas por padrão. Para obter mais informações e uma lista dessas regiões, consulte Enabling a Region no Referência geral da Amazon Web Services.
9. (Opcional) Para enviar notificações quando o relatório for gerado, expanda a seção Tópico do SNS e escolha um tópico existente do Amazon SNS em SNS topic Amazon Resource Name (ARN) (Nome do recurso da Amazon (ARN) do tópico do SNS).
10. Selecione Enviar.

Para obter informações sobre como exibir um histórico de relatórios gerados, consulte Visualizar histórico de relatórios de conformidade de patches.

Para obter informações sobre como visualizar detalhes das agendas de relatórios que você criou, consulte Visualizar as programações de relatórios de conformidade de patches.

Visualizar histórico de relatórios de conformidade de patches

Use as informações deste tópico para ajudar você a exibir detalhes sobre os relatórios de conformidade de patches gerados em seu Conta da AWS.

Para visualizar o histórico dos relatórios de conformidade dos patches

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Patch Manager.
3. Selecione a guia Compliance reporting (Relatório de conformidade).
4. Selecione View all S3 exports (Exibir todas as exportações do S3) e, em seguida, selecione a guia Export history (Histórico de exportação).

Visualizar as programações de relatórios de conformidade de patches

Use as informações deste tópico para ajudar você a exibir detalhes sobre as agendas de relatórios de conformidade de patch criadas em seu Conta da AWS.

Para visualizar o histórico dos relatórios de conformidade dos patches

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Patch Manager.
3. Selecione a guia Compliance reporting (Relatório de conformidade).
4. Selecione View all S3 exports (Exibir todas as exportações do S3) e, em seguida, selecione a guia Report schedule rules (Regras para o agendamento de relatórios).

Solução de problemas da geração de relatórios de conformidade de patches

Use as seguintes informações para ajudar você a solucionar problemas com a geração de relatórios de conformidade de patches no Patch Manager, uma ferramenta do AWS Systems Manager.

Tópicos

- Uma mensagem informa que a política AWS-SystemsManager-PatchManagerExportRolePolicy está corrompida
- Depois de excluir políticas ou funções de conformidade de patch, os relatórios agendados não são gerados com êxito

Uma mensagem informa que a política **`AWS-SystemsManager-PatchManagerExportRolePolicy`** está corrompida

Problema: Você recebe uma mensagem de erro semelhante à seguinte, indicando a`AWS-SystemsManager-PatchManagerExportRolePolicy`está corrompido:

```
An error occurred while updating the AWS-SystemsManager-PatchManagerExportRolePolicy
policy. If you have edited the policy, you might need to delete the policy, and any
role that uses it, then try again. Systems Manager recreates the roles and policies
you have deleted.
```

- Solução: use o console do Patch Manager ou a AWS CLI para excluir as políticas e os perfis afetados antes de gerar um novo relatório de conformidade de patches.

Para excluir a política corrompida usando o console

1. Abra o console do IAM, em https://console.aws.amazon.com/iam/.
2. Execute um destes procedimentos:

   Relatórios sob demanda: se o problema ocorreu durante a geração de um relatório por solicitação único, na navegação à esquerda, escolha Políticas, pesquise por `AWS-SystemsManager-PatchManagerExportRolePolicy` e exclua a política. Depois, escolha Perfis, pesquise por `AWS-SystemsManager-PatchSummaryExportRole` e, em seguida, exclua o perfil.

   Relatórios programados: se o problema ocorreu durante a geração de um relatório em um agendamento, na navegação à esquerda, escolha Políticas, pesquise um de cada vez por`AWS-EventBridge-Start-SSMAutomationRolePolicy` e `AWS-SystemsManager-PatchManagerExportRolePolicy` e exclua cada política. Depois, escolhaPerfil, pesquise um de cada vez por `AWS-EventBridge-Start-SSMAutomationRole` e `AWS-SystemsManager-PatchSummaryExportRole`, e exclua cada perfil.

Para excluir a política corrompida usando a AWS CLI

Substitua os *`valores dos espaços reservados`* pelo ID de sua conta.

- Se o problema ocorreu ao gerar um relatório único sob demanda, execute estes comandos:

```
aws iam delete-policy --policy-arn arn:aws:iam::account-id:policy/AWS-SystemsManager-PatchManagerExportRolePolicy
```

```
aws iam delete-role --role-name AWS-SystemsManager-PatchSummaryExportRole
```

Se o problema ocorreu ao gerar um relatório sobre um agendamento, execute estes comandos:

```
aws iam delete-policy --policy-arn arn:aws:iam::account-id:policy/AWS-EventBridge-Start-SSMAutomationRolePolicy
```

```
aws iam delete-policy --policy-arn arn:aws:iam::account-id:policy/AWS-
SystemsManager-PatchManagerExportRolePolicy
```

```
aws iam delete-role --role-name AWS-EventBridge-Start-SSMAutomationRole
```

```
aws iam delete-role --role-name AWS-SystemsManager-PatchSummaryExportRole
```

Após concluir todos os procedimento, siga as etapas para gerar ou agendar um novo relatório de conformidade de patch.

Depois de excluir políticas ou funções de conformidade de patch, os relatórios agendados não são gerados com êxito

Problema: A primeira vez que você gerar um relatório, o Systems Manager cria uma função de serviço e uma política a ser usada para o processo de exportação (`AWS-SystemsManager-PatchSummaryExportRoleeAWS-SystemsManager-PatchManagerExportRolePolicy`). Na primeira vez que você gerar um relatório em uma agenda, o Systems Manager cria outra função de serviço e uma política (`AWS-EventBridge-Start-SSMAutomationRoleeAWS-EventBridge-Start-SSMAutomationRolePolicy`). Eles permitem que o Amazon EventBridge inicie uma automação usando o runbookAWS-ExportPatchReporttos3.

Se você excluir qualquer uma dessas políticas ou funções, as conexões entre o agendamento e o bucket do S3 especificado e o tópico do Amazon SNS poderão ser perdidas.

- Solução: para contornar esse problema, recomendamos excluir o agendamento anterior e criar um novo agendamento para substituir o que estava enfrentando problemas.

Corrigir nós gerenciados fora de conformidade com o Patch Manager

Os tópicos desta seção fornecem visões gerais de como identificar nós gerenciados que estão fora da conformidade de patch e como colocar esses nós em conformidade.

Tópicos

- Identificar nós gerenciados fora de conformidade
- Valores de estados de conformidade de patches

- Aplicação de patches em nós gerenciados fora de conformidade

Identificar nós gerenciados fora de conformidade

Os nós gerenciados fora de conformidade são identificados quando um dos dois documentos do AWS Systems Manager (documentos SSM) são executados. Esses documentos do SSM fazem referência à lista de referência de patches apropriada para cada nó gerenciado do Patch Manager, uma ferramenta do AWS Systems Manager. Em seguida, eles avaliam o estado do patch do nó gerenciado e, em seguida, disponibilizam os resultados de conformidade para você.

Há dois documentos do SSM que são usados para identificar ou atualizar nós gerenciados fora de conformidade: `AWS-RunPatchBaseline` e `AWS-RunPatchBaselineAssociation`. Cada um é usado por diferentes processos, e seus resultados de conformidade estão disponíveis em diferentes canais. A tabela a seguir descreve as diferenças entre esses documentos.

> **Note**
>
> Dados de conformidade de patches do Patch Manager podem ser enviados para o AWS Security Hub. O Security Hub oferece uma visão abrangente dos alertas de segurança de alta prioridade e do status de conformidade. Também monitora o estado de aplicação de patches da sua frota. Para obter mais informações, consulte Integrar o Patch Manager ao AWS Security Hub.

	`AWS-RunPatchBaseline`	**`AWS-RunPatchBaselineAssociation`**
Processos que usam o documento	Patch sob demanda - Você pode verificar ou corrigir os nós gerenciados sob demanda usando a opção Patch now (Aplicar patch agora). Para mais informações, consulte Aplicação de patches em nós gerenciados sob demanda.	Host Management do Quick Setup do Systems Manager: é possível ativar uma opção de configuração do Host Management no Quick Setup para verificar as instâncias gerenciadas quanto à conformidade de patches todos os dias. Para mais informações, consulte

	AWS-RunPatchBaseline	AWS-RunPatchBaselineAssociation
	Políticas de patch do Quick Setup do Systems Manager: é possível criar uma configuração de patches no Quick Setup, uma ferramenta do AWS Systems Manager que pode verificar ou instalar patches que estão faltando em programações separadas para uma organização inteira, um subconjunto de unidades organizacionais ou uma única Conta da AWS. Para mais informações, consulte Configurar a aplicação de patches para instâncias em uma organização usando uma política de patch do Quick Setup. Execute um comando: é possível executar `AWS-RunPatchBaseline` manualmente em uma operação no Run Command, uma ferramenta do AWS Systems Manager. Para mais informações, consulte Executar comandos no console. Maintenance window (Janela de manutenção)— Você pode criar uma janela de manutenção que use o	Configurar o gerenciamento de host do Amazon EC2 usando a Quick Setup. Systems Manager Explorer: quando você habilita o Explorer, uma ferramenta do AWS Systems Manager, ele verifica suas instâncias gerenciadas regularmente no que diz respeito à conformidade de patches e relata os resultados no painel do Explorer.

	AWS-RunPatchBaseline	**AWS-RunPatchBaselineAssociation**
	documento do SSMAWS-RunPatchBaseline em umRun CommandTipo de tarefa. Para mais informações, consulte Tutorial: Create a maintenance window for patching using the console.	
Formato dos dados do resultado da verificação do patch	Depois que o AWS-RunPatchBaseline é executado, o Patch Manager envia um objeto do AWS:PatchSummary ao Inventory, uma ferramenta do AWS Systems Manager.	Depois que o AWS-RunPatchBaselineAssociation é executado, o Patch Manager envia um objeto do AWS:ComplianceItem ao Systems Manager Inventory.

	AWS-RunPatchBaseline	**AWS-RunPatchBaselineAssociation**
Visualizar relatórios da conformidade de patches no console	Você pode visualizar as informações de conformidade do patch para processos que usam o `AWS-RunPatchBaseline` em Conformidade da configuração do Systems Manager e em Trabalhar com nós gerenciados. Para obter mais informações, consulte Visualizar resultados de conformidade de patches.	Se você usa o Quick Setup para verificar as instâncias gerenciadas quanto à conformidade de patches, você pode ver o relatório de conformidade no Fleet Manager do Systems Manager. No console do Fleet Manager, escolha o ID do seu nó gerenciado. No menu Geral, escolha Conformidade de configuração. Se você usa o Explorer para verificar as instâncias gerenciadas quanto à conformidade de patches, você pode ver o relatório de conformidade no Explorer e no OpsCenter do Systems Manager.

	AWS-RunPatchBaseline	AWS-RunPatchBaselineAssociation
Comandos da AWS CLI para visualizar os resultados de conformidade de patches	Para processos que usam o AWS-RunPatchBaseline, você pode usar o seguinte comando da AWS CLI para visualizar informações resumidas sobre patches em um nó gerenciado. • describe-instance-patch-states • describe-instance-patch-states-for-patch-group • describe-patch-group-state	Para processos que usam o AWS-RunPatchBaselineAssociation, você pode usar o seguinte comando da AWS CLI para visualizar informações resumidas sobre patches em uma instância. • list-compliance-items
Sobre operações de aplicação de patches	Para processos que usam o AWS-RunPatchBaseline, você especifica se deseja que a operação execute uma Scan somente operação ou uma operação Scan and install. Se o objetivo for identificar nós gerenciados fora de conformidade e não os corrigir, execute apenas uma operação Scan.	Os processos do Quick Setup e Explorer, que usam o AWS-RunPatchBaselineAssociation, executam apenas uma operação de Scan.
Mais informações	Documento de comando do SSM para a aplicação de patches: AWS-RunPatchBaseline	Documento de comando do SSM para a aplicação de patches: AWS-RunPatchBaselineAssociation

Para obter informações sobre os vários estados de conformidade dos patches que você poderá ver relatados, consulte Valores de estados de conformidade de patches

Para obter informações sobre como corrigir nós gerenciados que não estiverem em conformidade com os patches, consulte Aplicação de patches em nós gerenciados fora de conformidade.

Valores de estados de conformidade de patches

As informações sobre patches de um nó gerenciado incluem um relatório do estado, ou status, de cada patch individual.

> **Tip**
>
> Para atribuir um estado de conformidade de patch específico a um nó gerenciado, você pode usar o comando put-compliance-items da AWS Command Line Interface (AWS CLI) ou a operação de API PutComplianceItems. A atribuição do estado da conformidade não tem suporte no console.

Use as informações nas tabelas a seguir para ajudar você a identificar por que um nó gerenciado pode estar fora de conformidade com patches.

Valores de conformidade de patch para Debian Server, Raspberry Pi OS e Ubuntu Server

No Debian Server, Raspberry Pi OS e Ubuntu Server, as regras de classificação de pacotes em diferentes estados de conformidade estão descritas na tabela abaixo.

> **Note**
>
> Tenha em mente o seguinte ao avaliar os valores de status `INSTALLED`, `INSTALLED_OTHER`, e `MISSING`: se você não marcar a caixa de seleção Incluir atualizações não relacionadas à segurança ao criar ou atualizar uma linha de base de patch, as versões candidatas a patches serão limitadas a patches nos repositórios a seguir.
>
> - Ubuntu Server 14.04 LTS: `trusty-security`
> - Ubuntu Server 16.04 LTS: `xenial-security`
> - Ubuntu Server 18.04 LTS: `bionic-security`
> - Ubuntu Server 20.04 LTS: `focal-security`
> - Ubuntu Server 20.10 STR: `groovy-security`

- Ubuntu Server 22.04 LTS (`jammy-security`)
- Ubuntu Server 23.04 (`lunar-security`)
- Ubuntu Server 23.10 (`mantic-security`)
- Ubuntu Server 24.04 LTS (`noble-security`)
- Ubuntu Server 24.10 (`oracular-security`)
- Ubuntu Server 25.04 (`plucky-security`)
- `debian-security` (Debian Server e Raspberry Pi OS)

Se você selecionar a caixa de seleção Incluir atualizações não relacionadas à segurança, os patches de outros repositórios também são considerados.

Estado do patch	Descrição	Compliance status (Status de conformidade)
`INSTALLED`	O patch está listado na lista de referência de patches e está instalado em seu nó gerenciado. Ele poderia ter sido instalado manualmente por um indivíduo, ou automatic amente pelo Patch Manager, quando o documento `AWS-RunPatchBaseline` foi executado em seu nó gerenciado.	Conforme
`INSTALLED_OTHER`	O patch não está incluído na lista de referência ou não é aprovado por ela, mas está instalado em seu nó gerenciado. O patch pode ter sido instalado manualmen te, o pacote pode ser uma dependência necessária	Conforme

Estado do patch	Descrição	Compliance status (Status de conformidade)
	de outro patch aprovado ou o patch pode ter sido incluído em uma operação InstallOverrideList. Se você não especificar `Block` como a ação Rejected patches (Patches rejeitados), os patches `INSTALLED_OTHER` também incluirão os patches instalados, porém rejeitados.	

Estado do patch	Descrição	Compliance status (Status de conformidade)
INSTALLED_PENDING_REBOOT	INSTALLED_PENDING_REBOOT pode significar uma de duas possibilidades: • A operação Patch Manager Install aplicou o patch ao nó gerenciado, mas o nó não foi reinicializado desde que o patch foi aplicado. Isso geralmente significa que a opção NoReboot foi selecionada para o parâmetro RebootOption quando o documento AWS-RunPatchBaseline foi executado pela última vez em seu nó gerenciado. Para obter mais informações, consulte Nome do parâmetro: RebootOption. • Um patch foi instalado fora do Patch Manager desde a última vez que o nó gerenciado foi reinicializado. Em nenhum dos casos isso significa que um patch com esse status exige uma reinicialização, mas apenas que o nó não foi reinicializado desde que o patch foi instalado.	incompatível:

Estado do patch	Descrição	Compliance status (Status de conformidade)
INSTALLED_REJECTED	O patch está instalado em seu nó gerenciado, mas está especificado em uma lista de patches rejeitados. Isso geralmente significa que o patch foi instalado antes de ser adicionado a uma lista de patches rejeitados.	incompatível:
MISSING	Pacotes que são filtrados através da linha de base e ainda não instalados.	incompatível:
FAILED	Failed: pacotes cuja instalaçã o na operação de patch foi malsucedida.	incompatível:

Valores de conformidade de patches para outros sistemas operacionais

Para todos os sistemas operacionais além do Debian Server, Raspberry Pi OS e Ubuntu Server, as regras de classificação de pacotes em diferentes estados de conformidade estão descritas na tabela abaixo.

Estado do patch	Descrição	Valor da conformidade
INSTALLED	O patch está listado na lista de referência de patches e está instalado em seu nó gerenciado. Ele poderia ter sido instalado manualmente por um indivíduo, ou automatic amente pelo Patch Manager, quando o documento `AWS-`	Conforme

Estado do patch	Descrição	Valor da conformidade
	`RunPatchBaseline` foi executado em seu nó.	

Estado do patch	Descrição	Valor da conformidade
`INSTALLED_OTHER` [1]	O patch não está na lista de referência, mas está instalado em seu nó gerenciado. Há dois motivos possíveis para isso: 1. O patch pode ter sido instalado manualmente. 2. Somente Linux: o pacote pode ter sido instalado como uma dependência necessária de um patch diferente aprovado. Se você especificar `Allow as dependency` como a ação Patches rejeitados, os patches instalados como dependências receberão o status `INSTALLED_OTHER` de relatório. **Note** O Windows Server não é compatíve l com o conceito de dependências de patches. Para obter informações sobre como o Patch Manager lida com patches na lista de patches rejeitado sWindows Server, consulte Opções	Conforme

Estado do patch	Descrição	Valor da conformidade
	de lista de patches rejeitados em listas de referência de patches personalizados.	
INSTALLED_REJECTED	O patch está instalado em seu nó gerenciado, mas está especificado em uma lista de patches rejeitados. Isso geralmente significa que o patch foi instalado antes de ser adicionado a uma lista de patches rejeitados.	incompatível:

Estado do patch	Descrição	Valor da conformidade
INSTALLED_PENDING_REBOOT	INSTALLED_PENDING_REBOOT pode significar uma de duas possibilidades: • A operação Patch Manager Install aplicou o patch ao nó gerenciado, mas o nó não foi reinicializado desde que o patch foi aplicado. Isso geralmente significa que a opção NoReboot foi selecionada para o parâmetro RebootOption quando o documento AWS-RunPatchBaseline foi executado pela última vez em seu nó gerenciado. Para obter mais informações, consulte Nome do parâmetro: RebootOption . • Um patch foi instalado fora do Patch Manager desde a última vez que o nó gerenciado foi reinicializado. Em nenhum dos casos isso significa que um patch com esse status exige uma reinicialização, mas apenas que o nó não foi reinicializado desde que o patch foi instalado.	incompatível:

Estado do patch	Descrição	Valor da conformidade
MISSING	O patch foi aprovado na lista de referência, mas não está instalado em seu nó gerenciado. Se você configurar a tarefa do documento `AWS-RunPatchBaseline` para verificar (em vez de instalar), o sistema relatará esse status para os patches que foram localizados durante a verificação, mas que não foram instalados.	incompatível:
FAILED	FAILED: o patch foi aprovado na linha de base, mas não pôde ser instalado. Para solucionar essa situação, reveja o resultado do comando para obter informações que possam ajudar você a entender o problema.	incompatível:

Estado do patch	Descrição	Valor da conformidade
NOT_APPLICABLE [1]	Este estado de conformidade é apenas indicado em sistemas operacionais do Windows Server. O patch está aprovado na lista de referência, mas o serviço ou recurso que o utiliza não está instalado em seu nó gerenciado. Por exemplo, um patch para o serviço do servidor Web, como os Serviços de Informações da Internet (IIS), mostraria `NOT_APPLICABLE` se fosse aprovado na linha de base, mas o serviço da Web não está instalado em seu nó gerenciado. Um patch também pode ser marcado como `NOT_APPLICABLE` se tiver sido substituído por uma atualização subsequente. Isso significa que a atualização posterior está instalada e a atualização `NOT_APPLICABLE` não é mais necessária.	Não aplicável

Estado do patch	Descrição	Valor da conformidade
AVAILABLE_SECURITY_UPDATES	Este estado de conformidade é apenas indicado em sistemas operacionais do Windows Server. Um patch de atualização de segurança disponível que não é aprovado pela lista de referência de patches pode ter um valor de conformidade `Compliant` ou `Non-Compliant`, conforme definido em uma lista de referência de patches personalizada. Ao criar ou atualizar uma lista de referência de patches, você escolhe o status que deseja atribuir aos patches de segurança que estão disponíveis, mas não aprovados, porque não atendem aos critérios de instalação especificados na lista de referência de patches. Por exemplo, os patches de segurança que você talvez queira instalar poderão ser ignorados se você tiver especificado um longo período de espera após o lançamento de um patch antes da instalação. Se uma atualização do patch for lançada durante o período de	Compatível ou Não compatível, dependendo da opção selecionada para as atualizações de segurança disponíveis. **Note** Usando o console para criar ou atualizar uma lista de referência de patches, você especifica essa opção no campo Status de conformidade das atualizações de segurança disponíveis. Usando a AWS CLI para executar o comando create-patch-baseline ou update-patch-baseline, você especifica essa opção no parâmetro `available-security-updates-compliance-status`.

Estado do patch	Descrição	Valor da conformidade
	espera especificado, o período de espera para instalação do patch recomeçará. Se o período de espera for muito longo, várias versões do patch poderão ser lançadas, mas nunca instaladas. Para contagens resumidas de patches, quando um patch é informado como `Available SecurityUpdate` , ele sempre será incluído em `AvailableSecurityU pdateCount` . Se a lista de referência estiver configurada para informar esses patches como `NonCompliant` , eles também serão incluídos em `SecurityNonComplia ntCount` . Se a lista de referência estiver configura da para informar esses patches como `Compliant` , eles não serão incluídos em `SecurityNonComplia ntCount` . Esses patches são sempre informados com uma severidade não especific ada e nunca são incluídos em `CriticalNonComplia ntCount` .	

¹ Para patches com o estado `INSTALLED_OTHER` e `NOT_APPLICABLE`, o Patch Manager omite alguns dados dos resultados de consultas com base no comando describe-instance-patches, como os valores para `Classification` e `Severity`. Isso é feito para ajudar a evitar ultrapassar o limite de dados para nós individuais no Inventory, uma ferramenta do AWS Systems Manager. Para visualizar todos os detalhes do patch, você pode usar o comando describe-available-patches.

Aplicação de patches em nós gerenciados fora de conformidade

Muitas das mesmas ferramentas e processos do AWS Systems Manager que você pode usar para conferir a conformidade do patch em nós gerenciados podem ser usadas para colocar os nós em conformidade com as regras de patch que se aplicam atualmente a eles. Para promover a conformidade com patches em nós gerenciados, o Patch Manager, uma ferramenta do AWS Systems Manager, deve executar uma operação `Scan and install`. (Se o objetivo for apenas identificar nós gerenciados fora de conformidade e não os corrigir, execute uma operação `Scan`. Para obter mais informações, consulte Identificar nós gerenciados fora de conformidade.)

Instalar patches usando o Systems Manager

Você pode escolher entre várias ferramentas para executar um`Scan and install`operação:

- (Recomendado) Configure uma política de patch no Quick Setup, uma ferramenta do Systems Manager que permite instalar patches que estão faltando de acordo com uma agenda para uma organização inteira, um subconjunto de unidades organizacionais ou uma única Conta da AWS. Para ter mais informações, consulte Configurar a aplicação de patches para instâncias em uma organização usando uma política de patch do Quick Setup.
- Crie uma janela de manutenção que use o documento Systems Manager (SSM document) `AWS-RunPatchBaseline` em um tipo de tarefa Run Command. Para ter mais informações, consulte Tutorial: Create a maintenance window for patching using the console.
- Executar manualmente`AWS-RunPatchBaseline`em umRun Commandoperação. Para ter mais informações, consulte Executar comandos no console.

- Instale patches sob demanda usando oPatch agoraopção. Para ter mais informações, consulte Aplicação de patches em nós gerenciados sob demanda.

Prevenção de substituições não intencionais de dados de conformidade de patches

Se você tiver vários tipos de operações em andamento para verificar a conformidade dos patches em suas instâncias, cada verificação substitui os dados de conformidade de patches de

verificações anteriores. Como resultado, você pode obter resultados inesperados em seus dados de conformidade de patch.

Por exemplo, suponha que você crie uma política de patch que verifique a conformidade dos patches todos os dias às 2h, horário local. Essa política de patch usa uma lista de referência de patches que visa patches com gravidade marcada como `Critical`, `Important` e `Moderate`. Essa lista de referência de patches também define alguns patches especificamente rejeitados.

Suponha também que você já tenha uma janela de manutenção configurada para realizar a verificação do mesmo conjunto de nós gerenciados todos os dias às 4h da manhã, no horário local, que você não exclui nem desativa. A tarefa dessa janela de manutenção usa uma lista de referência de patches diferente, que visa apenas patches com gravidade `Critical` e não exclui nenhum patch específico.

Quando essa segunda verificação é executada pela janela de manutenção, os dados de conformidade do patch da primeira verificação são excluídos e substituídos pela conformidade do patch da segunda verificação.

Portanto, é altamente recomendável usar apenas um método automatizado para verificar e instalar em suas operações de aplicação de patch. Se você estiver configurando políticas de patch, exclua ou desative outros métodos de verificação de conformidade de patches. Para obter mais informações, consulte os tópicos a seguir.

- Para remover uma tarefa de operação de aplicação de patch de uma janela de manutenção: Atualização ou cancelamento de registros de tarefas da janela de manutenção usando o console
- Para excluir uma associação de State Manager: Excluir associações.

Para desativar as verificações diárias de conformidade de patches em uma configuração de gerenciamento de host, faça o seguinte em Quick Setup:

1. No painel de navegação, escolha Quick Setup.
2. Selecione a configuração do Host Management a ser atualizada.
3. Escolha Actions, Edit configuration (Ações, Editar configuração).
4. Desmarque a caixa de seleção Scan instances for missing patches daily (Verificar instâncias em busca de patches ausentes diariamente).
5. Selecione Atualizar.

> **Note**
>
> O uso da opção Patch now (Aplicar patch agora) para verificar a conformidade de um nó gerenciado também resulta na substituição dos dados de conformidade do patch.

Aplicação de patches em nós gerenciados sob demanda

Usar a opção Patch agora no Patch Manager, uma ferramenta do AWS Systems Manager, é possível executar operações de patch sob demanda no console do Systems Manager. Isso significa que você não precisa criar um agendamento para atualizar o status de conformidade dos os nós gerenciados ou instalar patches em nós não compatíveis. Você também não precisa alternar o console do Systems Manager entre o Patch Manager e a Maintenance Windows, uma ferramenta do AWS Systems Manager para configurar ou modificar uma janela de patch programada.

A opção Patch now (Aplicar patch agora) é especialmente útil quando você tiver que aplicar atualizações de dia zero ou instalar outros patches críticos aos nós gerenciados o mais rápido possível.

> **Note**
>
> Há suporte para a aplicação de patches sob demanda para um único par Conta da AWS-Região da AWS por vez. Ela não pode ser usada com operações de aplicação de patches baseadas em políticas de patch. Recomendamos o uso de políticas de patch para manter todos os seus nós gerenciados em conformidade. Para mais informações sobre como trabalhar com políticas de patch, consulte Configurações de políticas de patches em Quick Setup.

Tópicos

- Como funciona o comando 'Patch now' (Aplicar patches agora)
- Executar 'Patch agora'

Como funciona o comando 'Patch now' (Aplicar patches agora)

Para executar o Patch agora, você especifica apenas duas configurações necessárias:

- Verificar somente se há patches ausentes ou verificar e instalar patches em nós gerenciados

- Em quais nós gerenciados executar a operação

Quando a operação Patch now (Aplicar patch agora) é executada, ela determina qual lista de referência de patches usar, da mesma forma que uma é selecionada para outras operações de aplicação de patches. Se um nó gerenciado estiver associado a um grupo de patches, a lista de referência de patches especificada para esse grupo será usada. Se o nó gerenciado não estiver associado a um grupo de patches, a operação usará a lista de referência de patches atualmente definida como padrão para o tipo de sistema operacional do nó gerenciado. Esta pode ser uma linha de base predefinida ou a linha de base personalizada que você definiu como padrão. Para obter mais informações sobre a seleção da lista de referência de patches, consulte Grupos de patches.

As opções que você pode especificar para Patch now (Aplicar patches agora) incluem escolher quando, ou se, reinicializar nós gerenciados após a aplicação do patch, especificar um bucket do Amazon Simple Storage Service (Amazon S3) para armazenar dados de log para a operação de patch e executar documentos do Systems Manager (documentos do SSM) como hooks de ciclo de vida durante a aplicação de patches.

Limites de erro e simultaneidade para 'Patch now' (Aplicar patch agora)

Para as operações Patch now (Aplicar patch agora), a simultaneidade e as opções de limite de erro são resolvidas pelo Patch Manager. Você não precisa especificar quantos nós gerenciados devem ser corrigidos de uma só vez, nem quantos erros são permitidos antes que a operação falhe. O Patch Manager aplica as configurações de limite de simultaneidade e erro descritas nas tabelas a seguir quando você aplica patches sob demanda.

> **Important**
>
> Os seguintes limites se aplicam somente a operações `Scan and install`. Para operações `Scan`, o Patch Manager tenta verificar até 1.000 nós simultaneamente e continuar a varredura até que ele tenha encontrado até 1.000 erros.

Simultaneidade: operações de instalação

Número total de nós gerenciados na operação Patch now (Aplicar patch agora)	Número de nós gerenciados verificados ou corrigidos de cada vez
Menos de 25	1

Número total de nós gerenciados na operação Patch now (Aplicar patch agora)	Número de nós gerenciados verificados ou corrigidos de cada vez
25-100	5%
101 a 1.000	8%
Mais de 1.000	10%

Limite de erro: operações de instalação

Número total de nós gerenciados na operação Patch now (Aplicar patch agora)	Número de erros permitidos antes da falha da operação
Menos de 25	1
25-100	5
101 a 1.000	10
Mais de 1.000	10

Usando hooks de ciclo de vida "Patch agora"

Patch agora oferece a capacidade de executar documentos de Comando do SSM como hook s de ciclo de vida durante uma operação de aplicação de patches de `Install`. É possível usar esses hooks para tarefas como desligar aplicações antes de aplicar patches ou executar verificações de integridade em aplicações após a aplicação de patches ou após uma reinicialização.

Para obter mais informações sobre hooks de ciclo de vida, consulte Documento de comando do SSM para a aplicação de patches: `AWS-RunPatchBaselineWithHooks`.

A tabela a seguir lista os hooks de ciclo de vida disponíveis para cada uma das três opções de reinicialização Patch agora, além de exemplos de usos para cada hook.

Hooks de ciclo de vida e usos de amostra

Opção de reinicializações	Hook: antes da instalação	Hook: após a instalação	Hook: na saída	Hook: após a reinicialização agendada
Reinicializar, se necessário	Execute um documento do SSM antes do início do patch. Exemplo de uso: encerre aplicações com segurança antes do início do processo de aplicação de patches.	Execute um documento SSM no final da operação de patch e antes da reinicialização do nó gerenciado. Exemplo de uso: execute operações como a instalação de aplicações de terceiros antes de uma possível reinicialização.	Execute um documento do SSM após concluir a operação de aplicação de patches e reinicializar as instâncias. Exemplo de uso: verifique se as aplicações estão sendo executadas conforme esperado após a aplicação de patches.	Não disponível
Não reinicialize minhas instâncias	O mesmo que acima.	Execute um documento do SSM no final da operação de patch. Exemplo de uso: verifique se as aplicações estão sendo executadas conforme esperado após	Não disponível	Não disponível

Opção de reinicializações	Hook: antes da instalação	Hook: após a instalação	Hook: na saída	Hook: após a reinicialização agendada
		a aplicação de patches.		
Agendar um tempo de reinicialização	O mesmo que acima.	Igual a de Não reinicialize minhas instâncias.	Não disponível	Execute um documento do SSM imediatamente após uma reinicialização programada estar concluída. Exemplo de uso: verifique se as aplicações estão sendo executadas conforme esperado após a reinicialização.

Executar 'Patch agora'

Use o procedimento a seguir para corrigir os nós gerenciados sob demanda.

Para executar 'Patch agora'

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Patch Manager.
3. Selecione Patch now (Patch agora).
4. Para a Patching operation (Operação de aplicação de patches), escolha uma das seguintes opções:

   - Scan (Verificar): o Patch Manager localiza quais patches estão faltando em seus nós gerenciados, mas não os instala. Você pode visualizar os resultados no painel Compliance

(Conformidade) ou em outras ferramentas que você usar para exibir a conformidade dos patches.

- Scan e install (Verificar e instalar): o Patch Manager localiza quais patches estão faltando em seus nós gerenciados e os instala.

5. Use esta etapa se escolher Scan e instalação na etapa anterior. Em Reboot Option (Opção de reinicialização), escolha uma das seguintes opções:

   - Reboot if needed (Reinicializar se necessário): após a instalação, o Patch Manager reinicializa os nós gerenciados somente se necessário para concluir uma instalação de patch.
   - Não reinicialize minhas instâncias: após a instalação, Patch Manager não reinicializará os nós. Você pode reinicializar nós manualmente ao escolher ou gerenciar reinicializações fora do Patch Manager.
   - Schedule a reboot time (Agendar um tempo de reinicialização): especifique a data, a hora e o fuso horário UTC para o Patch Manager para reiniciar os nós gerenciados. Depois de executar a operação Patch now (Aplicar o patch agora), a reinicialização agendada será listada como uma associação no State Manager com o nome `AWS-PatchRebootAssociation`.

6. Em Instances to patch (Instâncias que receberão os patches) escolha uma das seguintes opções:

   - Patch all instances (Aplicar patch a todas as instâncias): o Patch Manager executa a operação especificada em todos os nós gerenciados na sua Conta da AWS, na Região da AWS atual.
   - Patch only the target instances I specify (Aplicar patch somente aos nós gerenciados de destino que eu especificar): você especifica com quais nós gerenciados você deve trabalhar, na próxima etapa.

7. Use esta etapa se escolher Patch apenas as instâncias de destino que eu especificar na etapa anterior. Na seção Target selection (Seleção de destino), identifique os nós onde você deseja executar essa operação especificando etiquetas, selecionando nós manualmente ou especificando um grupo de recursos.

   > **Note**
   >
   > Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.
   > Se você optar por segmentar um grupo de recursos, observe que os grupos de recursos que são baseados em um AWS CloudFormation ainda devem ser marcados com a

tag `aws:cloudformation:stack-id` padrão. Se ele tiver sido removido, o Patch Manager poderá não conseguir determinar quais nós gerenciados pertencem ao grupo de recursos.

8. (Opcional) Em Patching log storage (Aplicação de patches ao armazenamento de logs), se você quiser criar e salvar logs dessa operação de patch, selecione o bucket S3 para armazenar os logs.

   > **Note**
   >
   > As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil de instância (para instâncias do EC2) ou perfil de serviço do IAM (máquinas ativadas para ambientes híbridos) atribuído à instância, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar o perfil de serviço do IAM obrigatório para o Systems Manager em ambientes híbridos e multinuvem. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço do IAM associado ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

9. (Opcional) Se você deseja executar documentos do SSM como hooks de ciclo de vida durante pontos específicos da operação de patch, faça o seguinte:

   - Selecione Usar hooks de ciclo de vida.
   - Para cada hook disponível, selecione o documento do SSM a ser executado no ponto especificado da operação:
     - Antes da instalação
     - Após a instalação
     - Na saída
     - Após a reinicialização agendada

     > **Note**
     >
     > O documento padrão,`AWS-Noop`, não executa operações.

10. Selecione Patch now (Patch agora).

A página Association execution targets (Destinos de execução da associação) é aberta. (O Patch agora usa associações no State Manager, uma ferramenta do AWS Systems Manager, para suas operações). Na área Operation summary (Resumo da operação), você pode monitorar o status da verificação ou a aplicação de patches em seus nós gerenciados.

Trabalhar com linhas de base de patches

Uma lista de referência de patches no Patch Manager, uma ferramenta do AWS Systems Manager, define quais patches são aprovados para instalação em nós gerenciados. Você pode especificar individualmente os patches aprovados ou rejeitados. Pode também criar regras de aprovação automática para especificar que determinados tipos de atualização (por exemplo, atualizações essenciais) devem ser aprovados automaticamente. A lista se rejeitados substitui as regras e a lista de aprovados. Para usar uma lista de patches aprovados para instalar pacotes específicos, primeiro remova todas as regras de aprovação automática. Se você identificar explicitamente um patch como rejeitado, ele não será aprovado ou instalado, mesmo que ele corresponda a todos os critérios em uma regra de aprovação automática. Além disso, um patch é instalado em um nó gerenciado somente se ele se aplicar ao software do nó gerenciado, mesmo que tenha sido aprovado para o mesmo.

Tópicos

- Visualizar as listas de referência de patches predefinidas da AWS
- Trabalhando com linhas de base de patch personalizadas
- Definir uma linha de base de patches existente como padrão

Mais informações

- Linhas de base de patch

Visualizar as listas de referência de patches predefinidas da AWS

O Patch Manager, uma ferramenta do AWS Systems Manager, inclui uma lista de referência de patches predefinida para cada sistema operacional compatível com o Patch Manager. Você pode usar essas linhas de base de patch (não pode personalizá-las) ou pode criar suas próprias. O procedimento a seguir descreve como visualizar uma linha de base de patch predefinida para ver se ela atende às suas necessidades. Para saber mais sobre linhas de base de patch, consulte Listas de referência de patches predefinidas e personalizadas.

Para visualizar linhas de base de patch predefinida da AWS

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Patch Manager.
3. Na lista de linhas de base de patch, escolha o ID da linha de base de uma das linhas de base de patch predefinidas.

   - ou -

   Se estiver acessando o Patch Manager pela primeira vez na Região da AWS atual, escolha Iniciar com uma visão geral, escolha a guia Listas de referência de patches e escolha o ID da lista de referência de uma das listas de referência de patches predefinidas.

   > **Note**
   >
   > Para o Windows Server, três linhas de base de patch predefinidas são fornecidas. As listas de referência de patches `AWS-DefaultPatchBaseline` e `AWS-WindowsPredefinedPatchBaseline-OS` oferecem suporte apenas às atualizações do sistema operacional Windows em si. O `AWS-DefaultPatchBaseline` é usado como lista de referência de patches para nós gerenciados do Windows Server, a menos que você especifique outra lista de referência de patches. As definições de configuração nestas duas linhas de base de patch são as mesmas. O mais novo dos dois,`AWS-WindowsPredefinedPatchBaseline-OS`, foi criado para distingui-lo da terceira linha de base de patch predefinida paraWindows Server. Essa lista de referência do patch, `AWS-WindowsPredefinedPatchBaseline-OS-Applications`, pode ser usada para aplicar patches tanto ao sistema operacional Windows Server quanto a aplicações compatíveis lançadas pela Microsoft.
   > Para obter mais informações, consulte Definir uma linha de base de patches existente como padrão.

4. Escolha a guia Regras de aprovação e analise a configuração da lista de referência de patches.
5. Se a configuração for aceitável para os nós gerenciados, você poderá passar para o procedimento Como criar e gerenciar grupos de patches.

   - ou -

   Para criar sua própria linha de base de patch padrão, prossiga para o tópico Trabalhando com linhas de base de patch personalizadas.

Trabalhando com linhas de base de patch personalizadas

O Patch Manager, uma ferramenta do AWS Systems Manager, inclui uma lista de referência de patches predefinida para cada sistema operacional compatível com o Patch Manager. Você pode usar essas linhas de base de patch (não pode personalizá-las) ou pode criar suas próprias.

Os procedimentos a seguir descrevem como criar sua própria lista de referência de patches personalizada. Para saber mais sobre linhas de base de patch, consulte Listas de referência de patches predefinidas e personalizadas.

Tópicos

- Criar uma lista de referência de patches personalizada para o Linux
- Criar uma lista de referência de patches personalizada para o macOS
- Criar uma lista de referência de patches personalizada para o Windows Server
- Atualizar ou excluir uma linha de base de patches personalizada

Criar uma lista de referência de patches personalizada para o Linux

Use o procedimento a seguir para criar uma lista de referência de patches personalizada para os nós gerenciados do Linux no Patch Manager, uma ferramenta do AWS Systems Manager.

Para obter informações sobre como criar uma lista de referência de patches para nós gerenciados do macOS, consulte Criar uma lista de referência de patches personalizada para o macOS. Para obter informações sobre como criar uma lista de referência de patches para nós gerenciados do Windows, consulte Criar uma lista de referência de patches personalizada para o Windows Server.

Para criar uma lista de referência de patches personalizada para os nós gerenciados do Linux

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Patch Manager.
3. Escolha a guia Listas de referência do patches e, em seguida, escolha Criar lista de referência de patches.

   - ou -

   Se estiver acessando o Patch Manager pela primeira vez na Região da AWS atual, escolha Iniciar com uma visão geral, escolha a guia Listas de referência de patches e depois escolha Criar lista de referência de patches.

4. Em Nome, insira um nome para a nova lista de referência de patches, como `MyRHELPatchBaseline`.
5. (Opcional) Em Description (Descrição), insira uma descrição para essa linha de base de patch.
6. Em Operating System (Sistema operacional), escolha um sistema operacional, como `Red Hat Enterprise Linux`.
7. Se você quiser começar a usar essa lista de referência de patch como o padrão para o sistema operacional assim que a criar, selecione a opção Set this patch baseline as the default patch baseline for ***`operating system name`*** instances (Definir esta lista de referência de patch como a padrão para instâncias do [nome do sistema operacional]).

   > **Note**
   >
   > Essa opção está disponível somente se você acessou pela primeira vez Patch Manager antes do lançamento das políticas de patch em 22 de dezembro de 2022.
   > Para obter informações sobre como definir uma linha de base de patch existente como padrão, consulte Definir uma linha de base de patches existente como padrão.

8. Na seção Approval rules for operating-systems (Regras de aprovação para sistemas operacionais), use os campos para criar uma ou mais regras de aprovação automática.

   - Produtos: a versão dos sistemas operacionais à qual a regra de aprovação se aplica, como `RedhatEnterpriseLinux7.4`. A seleção padrão é `All`.
   - Classificação: o tipo de patch ao qual a regra de aprovação se aplica, como `Security` ou `Enhancement`. A seleção padrão é `All`.

     > **Tip**
     >
     > É possível configurar uma lista de referência de patches para controlar se atualizações secundárias de versão do Linux são instaladas, como o RHEL 7.8. As atualizações de versões secundárias podem ser instaladas automaticamente pelo Patch Manager, desde que a atualização esteja disponível no repositório apropriado.
     > Para sistemas operacionais Linux, atualizações de versões secundárias não são classificadas de forma consistente. Elas podem ser classificadas como correções de bugs ou atualizações de segurança, ou não classificadas, mesmo dentro da mesma versão do kernel. Veja a seguir algumas opções para controlar se uma lista de referência de patches fará a instalação.

- Opção 1: a regra de aprovação mais ampla para garantir que atualizações de versões secundárias sejam instaladas quando disponíveis é especificar Classificação como `All` (*) e escolher a opção Incluir atualizações não relacionadas a segurança.
- Opção 2: para garantir que os patches para uma versão do sistema operacional estejam instalados, é possível usar um curinga (*) para especificar o formato de kernel na seção Exceções de patch da lista de referência. Por exemplo, o formato do kernel para o RHEL 7.* é `kernel-3.10.0-*.el7.x86_64`.

  Insira `kernel-3.10.0-*.el7.x86_64` na lista Patches aprovados na lista de referência de patches para garantir que todos os patches, inclusive atualizações de versões secundárias, sejam aplicados aos nós gerenciados do RHEL 7.*. (Se você souber o nome exato do pacote de um patch de versão secundária, poderá inseri-lo.)
- Opção 3: é possível obter o máximo de controle sobre quais patches são aplicados aos nós gerenciados, inclusive atualizações de versões secundárias, usando o parâmetro InstallOverrideList no documento `AWS-RunPatchBaseline`. Para obter mais informações, consulte Documento de comando do SSM para a aplicação de patches: `AWS-RunPatchBaseline`.

- Severity (Gravidade): o valor da gravidade dos patches aos quais a regra se aplica, como `Critical`. A seleção padrão é `All`.
- Auto-approval (Aprovação automática): o método para selecionar patches para aprovação automática.

> **Note**
>
> Como não é possível determinar de forma confiável as datas de lançamento dos pacotes de atualização do Ubuntu Server, as opções de aprovação automática não são compatíveis com esse sistema operacional.

- Approve patches after a specified number of days (Aprovar patches após um número específico de dias): o número de dias que o Patch Manager deve aguardar para aprovar automaticamente um patch após o lançamento ou última atualização de um patch. Insira

qualquer número inteiro de zero (0) a 360. Para a maioria dos casos, recomendamos que não aguarde mais de 100 dias.

- Approve patches released up to a specific date (Aprovar patches lançados até uma data específica): a data de lançamento do patch para a qual o Patch Manager aplica automaticamente todos os patches lançados ou com a última atualização nessa data ou antes dela. Por exemplo, se você especificar 7 de julho de 2023, nenhum patch lançado ou com última atualização em ou após 8 de julho de 2023 será instalado automaticamente.

- (Opcional) Relatórios de conformidade: o nível de gravidade que você deseja atribuir aos patches aprovados pela lista de referência, como `Critical` ou `High`.

> **Note**
>
> Se você especificar um nível de relatório de conformidade e o estado do patch de qualquer patch aprovado for relatado como `Missing`, a gravidade geral da conformidade relatada pela lista de referência de patches será o nível de gravidade especificado.

- Include non-security updates (Incluir atualizações não relacionadas a segurança): marque a caixa de seleção para instalar patches do sistema operacional Linux não relacionados a segurança que estão disponíveis no repositório de origem, além de patches de segurança.

> **Note**
>
> No SUSE Linux Enterprise Server, (SLES) não é necessário marcar a caixa de seleção, pois os patches relacionados ou não a segurança são instalados por padrão em nós gerenciados do SLES. Para obter mais informações, consulte o conteúdo sobre o SLES em Como os patches de segurança são selecionados.

Para obter mais informações sobre como trabalhar com regras de aprovação em uma linha de base de patch personalizada, consulte Linhas de base personalizadas.

9. Se quiser explicitamente aprovar qualquer patch, além das suas regras de aprovação, faça o seguinte na seção Patch exceptions (Exceções de patch):

   - Em Approved patches (Patches aprovados), insira uma lista separada por vírgulas dos patches que você deseja aprovar.

Para obter mais informações sobre os formatos aceitos de listas de patches aprovados e rejeitados, consulte Sobre formatos de nomes de pacotes para listas de patches aprovados e rejeitados.

- (Opcional) Em Approved patches compliance level (Nível de conformidade dos patches aprovados), atribua um nível de conformidade aos patches na lista.
- Se algum dos patches aprovados que você especificar não for relacionado a segurança, marque a caixa Incluir atualizações não relacionadas a segurança para que esse patch também seja instalado no sistema operacional Linux.

10. Se quiser explicitamente rejeitar qualquer patch que de outra forma atendem às suas regras de aprovação, faça o seguinte na seção Patch exceptions (Exceções de patch):

    - Em Rejected patches (Patches rejeitados), insira uma lista separada por vírgulas dos patches que você deseja rejeitar.

      Para obter mais informações sobre os formatos aceitos de listas de patches aprovados e rejeitados, consulte Sobre formatos de nomes de pacotes para listas de patches aprovados e rejeitados.
    - Em Rejected patches action (Ação para patches rejeitados), selecione a ação que o Patch Manager deve realizar para patches incluídos na lista Rejected patches (Patches rejeitados).
      - Permitir como dependência: um pacote na lista Rejected patches (Patches rejeitados) é instalado apenas se for uma dependência de outro. Ele é considerado compatível com a linha de base de patch e seu status é relatado como InstalledOther. Essa é a ação padrão se nenhuma opção for especificada.
      - Bloco: os pacotes na lista Patches rejeitados e pacotes que os incluem como dependências não são instalados pelo Patch Manager em nenhuma circunstância. Se um pacote tiver sido instalado antes de ser adicionado à lista Patches rejeitados ou instalado fora do Patch Manager, ele será considerado como fora conformidade com a lista de referência de patches e seu status será indicado como InstalledRejected.

> **Note**
>
> O Patch Manager pesquisa dependências de patches recursivamente.

11. (Opcional) Se quiser especificar repositórios de patches alternativos para versões diferentes de um sistema operacional, como AmazonLinux2016.03 e AmazonLinux2017.09, faça o seguinte para cada produto na seção Patch sources (Origens de patches):

    - Em Name (Nome), insira um nome para ajudar você a identificar a configuração de origem.
    - Em Product (Produto), selecione a versão dos sistemas operacionais para a qual o repositório de origem de patches é destinado, como `RedhatEnterpriseLinux7.4`.
    - Em Configuração, insira o valor da configuração do repositório a ser usado formato apropriado:

      Example for yum repositories

      ```
      [main]
      name=MyCustomRepository
      baseurl=https://my-custom-repository
      enabled=1
      ```

      > **Tip**
      >
      > Para obter informações sobre outras opções disponíveis para a configuração do repositório yum, consulte dnf.conf (5).

      Examples for Ubuntu Server and Debian Server

      ```
      deb http://security.ubuntu.com/ubuntu jammy main
      ```

      ```
      deb https://site.example.com/debian distribution component1
      component2 component3
      ```

      As informações de repositório para repositórios do Ubuntu Server devem ser especificadas em uma única linha. Para obter mais exemplos e informações, consulte jammy (5) sources.list.5.gz no site Ubuntu Server Manuals e sources.list format no Debian Wiki.

    Selecione Add another source (Adicionar outra origem) para especificar um repositório de origem para cada versão do sistema operacional adicional, até um máximo de 20.

    Para obter mais informações sobre repositórios de origem de patches alternativos, consulte Como especificar um repositório de origem de patches alternativo (Linux).

12. (Opcional) Em Manage tags (Gerenciar tags), aplique um ou mais pares de nome/valor de chave de tag à linha de base de patch.

    Tags são metadados opcionais que você atribui a um recurso. As tags permitem categorizar um recurso de diferentes formas, como por finalidade, proprietário ou ambiente. Por exemplo, você pode querer marcar uma linha de base de patch para identificar o nível de gravidade dos patches especificados, a família de sistemas operacionais aos quais ela se aplica e o tipo de ambiente. Nesse caso, você pode especificar tags semelhantes aos seguintes pares de nome/valor:

    - `Key=PatchSeverity,Value=Critical`
    - `Key=OS,Value=RHEL`
    - `Key=Environment,Value=Production`
13. Escolha Create patch baseline.

Criar uma lista de referência de patches personalizada para o macOS

Use o procedimento a seguir para criar uma lista de referência de patches personalizada para os nós gerenciados do macOS no Patch Manager, uma ferramenta do AWS Systems Manager.

Para obter informações sobre como criar uma lista de referência de patches para nós gerenciados do Windows Server, consulte Criar uma lista de referência de patches personalizada para o Windows Server. Para obter informações sobre como criar uma lista de referência de patches para nós gerenciados do Linux, consulte Criar uma lista de referência de patches personalizada para o Linux.

> **Note**
>
> Não há suporte para macOS em todas as Regiões da AWS. Para obter mais informações sobre o suporte a instâncias do EC2 para macOS, consulte Instâncias Mac do Amazon EC2 no Guia do usuário do Amazon EC2.

Para criar uma lista personalizada de referência de patches para os nós gerenciados do macOS

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Patch Manager.
3. Escolha a guia Listas de referência do patches e, em seguida, escolha Criar lista de referência de patches.

- ou -

Se estiver acessando o Patch Manager pela primeira vez na Região da AWS atual, escolha Iniciar com uma visão geral, escolha a guia Listas de referência de patches e depois escolha Criar lista de referência de patches.

4. Em Nome, insira um nome para a nova lista de referência de patches, como `MymacOSPatchBaseline`.
5. (Opcional) Em Description (Descrição), insira uma descrição para essa linha de base de patch.
6. Em Operating system (Sistema operacional), escolha macOS.
7. Se você quiser começar a usar essa lista de referência de patch como o padrão para o macOS assim que a criar, selecione a opção Set this patch baseline as the default patch baseline for macOS instances (Definir esta lista de referência de patch como a padrão para instâncias do Windows Server).

> **Note**
>
> Essa opção está disponível somente se você acessou pela primeira vez Patch Manager antes do lançamento das políticas de patch em 22 de dezembro de 2022.
> Para obter informações sobre como definir uma linha de base de patch existente como padrão, consulte Definir uma linha de base de patches existente como padrão.

8. Na seção Approval rules for operating-systems (Regras de aprovação para sistemas operacionais), use os campos para criar uma ou mais regras de aprovação automática.

   - Produto: a versão dos sistemas operacionais à qual a regra de aprovação se aplica, como `Mojave10.14.1` ou `Catalina10.15.1`. A seleção padrão é `All`.

   > **Note**
   >
   > O sistema de gerenciamento de pacotes de software de código aberto Homebrew descontinuou o suporte para macOS 10.14.x (Mojave) e 10.15.x (Catalina). Como resultado, as operações de aplicação de patches usando o Patch Manager não são compatíveis nessas versões.

   - Classificação: o gerenciador ou gerenciadores de pacotes aos quais você deseja aplicar pacotes durante o processo de aplicação de patches. Você pode escolher entre as seguintes opções:

- atualização de software
- installer
- brew
- brew cask

A seleção padrão é `All`.

- (Opcional) Relatórios de conformidade: o nível de gravidade que você deseja atribuir aos patches aprovados pela lista de referência, como `Critical` ou `High`.

> **Note**
>
> Se você especificar um nível de relatório de conformidade e o estado do patch de qualquer patch aprovado for relatado como `Missing`, a gravidade geral da conformidade relatada pela lista de referência de patches será o nível de gravidade especificado.

Para obter mais informações sobre como trabalhar com regras de aprovação em uma linha de base de patch personalizada, consulte Linhas de base personalizadas.

9. Se quiser explicitamente aprovar qualquer patch, além das suas regras de aprovação, faça o seguinte na seção Patch exceptions (Exceções de patch):

   - Em Approved patches (Patches aprovados), insira uma lista separada por vírgulas dos patches que você deseja aprovar.

     Para obter mais informações sobre os formatos aceitos de listas de patches aprovados e rejeitados, consulte Sobre formatos de nomes de pacotes para listas de patches aprovados e rejeitados.

   - (Opcional) Em Approved patches compliance level (Nível de conformidade dos patches aprovados), atribua um nível de conformidade aos patches na lista.

10. Se quiser explicitamente rejeitar qualquer patch que de outra forma atendem às suas regras de aprovação, faça o seguinte na seção Patch exceptions (Exceções de patch):

    - Em Rejected patches (Patches rejeitados), insira uma lista separada por vírgulas dos patches que você deseja rejeitar.

Para obter mais informações sobre os formatos aceitos de listas de patches aprovados e rejeitados, consulte Sobre formatos de nomes de pacotes para listas de patches aprovados e rejeitados.

- Em Rejected patches action (Ação para patches rejeitados), selecione a ação que o Patch Manager deve realizar para patches incluídos na lista Rejected patches (Patches rejeitados).
  - Permitir como dependência: um pacote na lista Rejected patches (Patches rejeitados) é instalado apenas se for uma dependência de outro. Ele é considerado compatível com a linha de base de patch e seu status é relatado como InstalledOther. Essa é a ação padrão se nenhuma opção for especificada.
  - Bloco: os pacotes na lista Patches rejeitados e pacotes que os incluem como dependências não são instalados pelo Patch Manager em nenhuma circunstância. Se um pacote tiver sido instalado antes de ser adicionado à lista Patches rejeitados ou instalado fora do Patch Manager, ele será considerado como fora conformidade com a lista de referência de patches e seu status será indicado como InstalledRejected.

11. (Opcional) Em Manage tags (Gerenciar tags), aplique um ou mais pares de nome/valor de chave de tag à linha de base de patch.

    Tags são metadados opcionais que você atribui a um recurso. As tags permitem categorizar um recurso de diferentes formas, como por finalidade, proprietário ou ambiente. Por exemplo, você pode querer marcar uma linha de base de patch para identificar o nível de gravidade dos patches especificados, a família de sistemas operacionais aos quais ela se aplica e o tipo de ambiente. Nesse caso, você pode especificar tags semelhantes aos seguintes pares de nome/valor:

    - `Key=PatchSeverity,Value=Critical`
    - `Key=PackageManager,Value=softwareupdate`
    - `Key=Environment,Value=Production`

12. Escolha Create patch baseline.

Criar uma lista de referência de patches personalizada para o Windows Server

Use o procedimento a seguir para criar uma lista de referência de patches personalizada para os nós gerenciados do Windows no Patch Manager, uma ferramenta do AWS Systems Manager.

Para obter informações sobre como criar uma lista de referência de patches para nós gerenciados do Linux, consulte Criar uma lista de referência de patches personalizada para o Linux. Para obter

informações sobre como criar uma lista de referência de patches para nós gerenciados do macOS, consulte Criar uma lista de referência de patches personalizada para o macOS.

Para obter um exemplo de criação de uma lista de referência de patches limitada à instalação apenas do Windows Service Packs, consulte Tutorial: Criar uma lista de referência de patches para instalar o Windows Service Packs usando o console.

Para criar uma linha de base de patch personalizada (Windows)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Patch Manager.
3. Escolha a guia Listas de referência do patches e, em seguida, escolha Criar lista de referência de patches.

   - ou -

   Se estiver acessando o Patch Manager pela primeira vez na Região da AWS atual, escolha Iniciar com uma visão geral, escolha a guia Listas de referência de patches e depois escolha Criar lista de referência de patches.
4. Em Nome, insira um nome para a nova lista de referência de patches, como `MyWindowsPatchBaseline`.
5. (Opcional) Em Description (Descrição), insira uma descrição para essa linha de base de patch.
6. Em Operating system (Sistema operacional), escolha `Windows`.
7. Em Status de conformidade de atualizações de segurança disponíveis, escolha o status que você deseja atribuir aos patches de segurança que estão disponíveis, mas não aprovados, porque não atendem aos critérios de instalação especificados na lista de referência de patches, Não compatível ou Compatível.

   Cenário de exemplo: os patches de segurança que você talvez queira instalar poderão ser ignorados se você tiver especificado um longo período de espera após o lançamento de um patch antes da instalação. Se uma atualização do patch for lançada durante o período de espera especificado, o período de espera para instalação do patch recomeçará. Se o período de espera for muito longo, várias versões do patch poderão ser lançadas, mas nunca instaladas.
8. Se você deseja começar a usar essa linha de base de patch como o padrão para o Windows assim que a criar, selecione Set this patch baseline as the default patch baseline for Windows Server instances (Definir essa linha de base de patch como a linha de base de patch padrão para instâncias do Windows Server).

> **Note**
>
> Essa opção está disponível somente se você acessou pela primeira vez Patch Manager antes do lançamento das políticas de patch em 22 de dezembro de 2022.
> Para obter informações sobre como definir uma linha de base de patch existente como padrão, consulte Definir uma linha de base de patches existente como padrão.

9. Na seção Approval rules for operating systems (Regras de aprovação para sistemas operacionais), use os campos para criar uma ou mais regras de aprovação automática.

   - Produto: a versão dos sistemas operacionais à qual a regra de aprovação se aplica, como `WindowsServer2012`. A seleção padrão é `All`.
   - Classificação: o tipo de patch ao qual a regra de aprovação se aplica, como `CriticalUpdates`, `Drivers` e `Tools`. A seleção padrão é `All`.

     > **Tip**
     >
     > É possível incluir instalações do Windows Service Pack nas regras de aprovação, incluindo `ServicePacks` ou escolhendo `All` na lista Classificação. Para obter um exemplo, consulte Tutorial: Criar uma lista de referência de patches para instalar o Windows Service Packs usando o console.

   - Severity (Gravidade): o valor da gravidade dos patches aos quais a regra se aplica, como `Critical`. A seleção padrão é `All`.
   - Auto-approval (Aprovação automática): o método para selecionar patches para aprovação automática.
     - Approve patches after a specified number of days (Aprovar patches após um número específico de dias): o número de dias que o Patch Manager deve aguardar para aprovar automaticamente um patch após o lançamento ou atualização de um patch. Insira qualquer número inteiro de zero (0) a 360. Para a maioria dos casos, recomendamos que não aguarde mais de 100 dias.
     - Approve patches released up to a specific date (Aprovar patches lançados até uma data específica): a data de lançamento do patch para a qual o Patch Manager aplica automaticamente todos os patches lançados ou com a última atualização nessa data ou antes dela. Por exemplo, se você especificar 7 de julho de 2023, nenhum patch lançado ou com última atualização em ou após 8 de julho de 2023 será instalado automaticamente.

- (Opcional) Compliance reporting (Relatórios de conformidade): o nível de gravidade que você deseja atribuir aos patches aprovados pela linha de base, como `High`.

> **Note**
>
> Se você especificar um nível de relatório de conformidade e o estado do patch de qualquer patch aprovado for relatado como `Missing`, a gravidade geral da conformidade relatada pela lista de referência de patches será o nível de gravidade especificado.

10. Na seção Approval rules for applications (Regras de aprovação para aplicações), use os campos para criar uma ou mais regras de aprovação automática.

> **Note**
>
> Em vez de especificar regras de aprovação, você pode especificar listas de patches aprovados e rejeitados como exceções de patch. Consulte as etapas 10 e 11.

- Product family (Família de produtos): a família de produtos Microsoft geral para a qual você deseja especificar uma regra, como `Office` ou `Exchange Server`.
- Produto: a versão do aplicativo à qual a regra de aprovação se aplica, como `Office 2016` ou `Active Directory Rights Management Services Client 2.0 2016`. A seleção padrão é `All`.
- Classification (Classificação): o tipo de patch ao qual a regra de aprovação se aplica, como `CriticalUpdates`. A seleção padrão é `All`.
- Severity (Gravidade): o valor da gravidade dos patches aos quais a regra se aplica, como `Critical`. A seleção padrão é `All`.
- Auto-approval (Aprovação automática): o método para selecionar patches para aprovação automática.
  - Approve patches after a specified number of days (Aprovar patches após um número específico de dias): o número de dias que o Patch Manager deve aguardar para aprovar automaticamente um patch após o lançamento ou atualização de um patch. Insira qualquer número inteiro de zero (0) a 360. Para a maioria dos casos, recomendamos que não aguarde mais de 100 dias.

- Approve patches released up to a specific date (Aprovar patches lançados até uma data específica): a data de lançamento do patch para a qual o Patch Manager aplica automaticamente todos os patches lançados ou com a última atualização nessa data ou antes dela. Por exemplo, se você especificar 7 de julho de 2023, nenhum patch lançado ou com última atualização em ou após 8 de julho de 2023 será instalado automaticamente.

- (Opcional) Relatórios de conformidade: o nível de gravidade que você deseja atribuir aos patches aprovados pela lista de referência, como `Critical` ou `High`.

> **Note**
>
> Se você especificar um nível de relatório de conformidade e o estado do patch de qualquer patch aprovado for relatado como `Missing`, a gravidade geral da conformidade relatada pela lista de referência de patches será o nível de gravidade especificado.

11. (Opcional) Se você quiser explicitamente aprovar qualquer patch em vez de permitir que os patches sejam selecionados de acordo com regras de aprovação, faça o seguinte na seção Patch exceptions (Exceções de patch):

    - Em Approved patches (Patches aprovados), insira uma lista separada por vírgulas dos patches que você deseja aprovar.

      Para obter mais informações sobre os formatos aceitos de listas de patches aprovados e rejeitados, consulte Sobre formatos de nomes de pacotes para listas de patches aprovados e rejeitados.

    - (Opcional) Em Approved patches compliance level (Nível de conformidade dos patches aprovados), atribua um nível de conformidade aos patches na lista.

12. Se quiser explicitamente rejeitar qualquer patch que de outra forma atendem às suas regras de aprovação, faça o seguinte na seção Patch exceptions (Exceções de patch):

    - Em Rejected patches (Patches rejeitados), insira uma lista separada por vírgulas dos patches que você deseja rejeitar.

      Para obter mais informações sobre os formatos aceitos de listas de patches aprovados e rejeitados, consulte Sobre formatos de nomes de pacotes para listas de patches aprovados e rejeitados.

- Em Rejected patches action (Ação para patches rejeitados), selecione a ação que o Patch Manager deve realizar para patches incluídos na lista Rejected patches (Patches rejeitados).
  - Permitir como dependência: o Windows Server não é compatível com o conceito de dependências de pacotes. Se um pacote estiver na lista de Patches rejeitados e já estiver instalado no nó, seu status será relatado como `INSTALLED_OTHER`. Qualquer pacote que ainda não esteja instalado no nó é ignorado.
  - Bloquear: os pacotes na lista de patches rejeitados não são instalados pelo Patch Manager em nenhuma circunstância. Se um pacote tiver sido instalado antes de ser adicionado à lista Patches rejeitados ou instalado fora do Patch Manager, ele será considerado como fora conformidade com a lista de referência de patches e seu status será indicado como `INSTALLED_REJECTED`.

  Para obter mais informações sobre ações de pacotes rejeitados, consulte Opções de lista de patches rejeitados em listas de referência de patches personalizados.

13. (Opcional) Em Manage tags (Gerenciar tags), aplique um ou mais pares de nome/valor de chave de tag à linha de base de patch.

    Tags são metadados opcionais que você atribui a um recurso. As tags permitem categorizar um recurso de diferentes formas, como por finalidade, proprietário ou ambiente. Por exemplo, você pode querer marcar uma linha de base de patch para identificar o nível de gravidade dos patches especificados, a família de sistemas operacionais aos quais ela se aplica e o tipo de ambiente. Nesse caso, você pode especificar tags semelhantes aos seguintes pares de nome/valor:

    - `Key=PatchSeverity,Value=Critical`
    - `Key=OS,Value=RHEL`
    - `Key=Environment,Value=Production`

14. Escolha Create patch baseline.

Atualizar ou excluir uma linha de base de patches personalizada

Você pode atualizar ou excluir uma lista de referência de patches' personalizada que criou no Patch Manager, uma ferramenta do AWS Systems Manager. Ao atualizar uma linha de base do patch, você pode alterar seu nome ou descrição, suas regras de aprovação e suas exceções para os patches aprovados e rejeitados. Você também pode atualizar as tags que são aplicadas à linha de base de patch. Você não pode alterar o tipo de sistema operacional para o qual uma linha de base de patch

foi criada e não pode fazer alterações em uma linha de base de patch predefinida fornecida pela AWS.

Atualizar ou excluir uma linha de base de patches

Siga estas etapas para atualizar ou excluir uma linha de base de patch.

> **Important**
>
> Tenha cuidado ao excluir uma lista de referência de patches personalizada que possa ser usada por uma configuração de política de patch no Quick Setup.
> Se você estiver usando uma configuração de política de patch em Quick Setup, as atualizações feitas nas listas de referência de patches personalizadas serão sincronizadas com Quick Setup uma vez por hora.
> Se uma lista de referência de patches personalizada que foi referenciada em uma política de patch for excluída, um banner será exibido na página Configuration details (Detalhes da configuração) do Quick Setup da sua política de patch. O banner informa que a política de patch faz referência a uma lista de referência de patches que não existe mais e que as operações de aplicação de patches subsequentes falharão. Nesse caso, retorne à página Configurations (Configurações) do Quick Setup, selecione a configuração Patch Manager e escolha Actions (Ações), Edit configuration (Editar configuração). O nome da lista de referência de patches excluída será destacado, e você deverá selecionar uma nova lista de referência de patches para o sistema operacional afetado.

Para atualizar ou excluir uma linha de base de patches

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Patch Manager.
3. Escolha a linha de base de patch que você deseja atualizar ou excluir e execute uma das seguintes ações:

   - Para remover a lista de referência de patches da sua Conta da AWS, escolha Delete (Excluir). O sistema solicitará que você confirme suas ações.
   - Para fazer alterações no nome ou na descrição da linha de base de patch, nas regras de aprovação ou nas exceções de patch, escolha Edit (Editar). Na página Edit patch baseline (Editar linha de base de patch), altere as opções e os valores desejados e escolha Save changes (Salvar alterações).

- Para adicionar, alterar ou excluir tags aplicadas à linha de base de patch, escolha a guia Tags e depois escolha Edit tags (Editar tags). Na página Edit patch baseline tags (Editar tags de linha de base de patch), faça atualizações nas tags da linha de base de patch e escolha Save changes (Salvar alterações).

Para obter informações sobre as opções de configuração que podem ser feitas, consulte Trabalhando com linhas de base de patch personalizadas.

Definir uma linha de base de patches existente como padrão

> **Important**
>
> Qualquer seleção padrão da lista de referência de patches que você fizer aqui não se aplicará às operações de aplicação de patches baseadas em uma política de patch. As políticas de patch usam suas próprias especificações de lista de referência de patches. Para obter mais informações sobre políticas de patch, consulte Configurações de políticas de patches em Quick Setup.

Ao criar uma lista de referência de patches personalizada em Patch Manager, uma ferramenta do AWS Systems Manager, você pode defini-la como padrão para o tipo de sistema operacional associado assim que a criar. Para ter mais informações, consulte Trabalhando com linhas de base de patch personalizadas.

Você também pode definir uma linha de base de patch existente como padrão para um tipo de sistema operacional.

> **Note**
>
> As etapas a serem seguidas variam se você acessou o Patch Manager pela primeira vez antes ou depois do lançamento das políticas de patch de 22 de dezembro de 2022. Se você usou o Patch Manager antes dessa data, use o procedimento do console. Caso contrário, use o procedimento da AWS CLI. O menu Ações referenciado no procedimento do console não é exibido em regiões em que o Patch Manager não era usado antes do lançamento das políticas de patch.

Para definir uma linha de base de patch como padrão

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Patch Manager.
3. Selecione a guia Patch baselines (Listas de referência do patch).
4. Na lista de referências do patch, escolha o botão de uma lista de referência do patch que não esteja definida como padrão para um tipo de sistema operacional.

   A coluna Default baseline (Linha de base padrão) indica quais linhas de base estão atualmente definidas como padrões.
5. No menu Actions (Ações), escolha Set default patch baseline (Definir linha de base de patch padrão).

   > **Important**
   >
   > O menu Ações não está disponível se você não trabalhou com o Patch Manager na Conta da AWS e na região atuais antes de 22 de dezembro de 2022. Consulte a observação anterior neste tópico para obter mais informações.
6. Na caixa de diálogo de confirmação, escolha Set default (Definir padrão).

Para definir uma lista de referência de patches como padrão (AWS CLI)

1. Execute o comando describe-patch-baselines para ver uma lista das listas de referência de patches disponíveis e seus IDs e nomes do recurso da Amazon (ARNs).

   ```
   aws ssm describe-patch-baselines
   ```
2. Execute o comando register-default-patch-baseline para definir uma lista de referência como padrão para o sistema operacional ao qual ela está associada. Substitua *baseline-id-or-ARN* pelo ID da lista de referência de patches personalizada ou da lista de referência predefinida a ser usada.

   Linux & macOS

   ```
   aws ssm register-default-patch-baseline \
       --baseline-id baseline-id-or-ARN
   ```

Veja a seguir um exemplo de como definir uma lista de referência personalizada como padrão.

```
aws ssm register-default-patch-baseline \
    --baseline-id pb-abc123cf9bEXAMPLE
```

Veja a seguir um exemplo de como definir uma lista de referência predefinida gerenciada pela AWS como padrão.

```
aws ssm register-default-patch-baseline \
    --baseline-id arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-0574b43a65ea646e
```

Windows Server

```
aws ssm register-default-patch-baseline ^
    --baseline-id baseline-id-or-ARN
```

Veja a seguir um exemplo de como definir uma lista de referência personalizada como padrão.

```
aws ssm register-default-patch-baseline ^
    --baseline-id pb-abc123cf9bEXAMPLE
```

Veja a seguir um exemplo de como definir uma lista de referência predefinida gerenciada pela AWS como padrão.

```
aws ssm register-default-patch-baseline ^
    --baseline-id arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-071da192df1226b63
```

Visualizar patches disponíveis

O Patch Manager, uma ferramenta do AWS Systems Manager, permite visualizar todos os patches disponíveis para um sistema operacional especificado e, opcionalmente, uma versão específica do sistema operacional.

> **Tip**
>
> Para gerar uma lista de patches disponíveis e salvá-los em um arquivo, você pode usar o comando describe-available-patches e especificar a saída de sua preferência.

Para visualizar patches disponíveis

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Patch Manager.
3. Selecione a guia Patches.

   - ou -

   Se estiver acessando Patch Manager pela primeira vez no atual Região da AWS, escolha Start with an overview (Iniciar com uma visão geral) e depois escolha a guia Patches.

   > **Note**
   >
   > No Windows Server, a guia Patches exibe as atualizações que estão disponíveis no Windows Server Update Services (WSUS).

4. Para Operating system (Sistema operacional), escolha o sistema operacional para o qual você deseja visualizar os patches disponíveis, como`Windows` ou `Amazon Linux`.
5. (Opcional) Para Product (Produto), escolha uma versão do sistema operacional, como `WindowsServer2019` ou `AmazonLinux2018.03`.
6. (Opcional) Para adicionar ou remover colunas de informações para seus resultados, escolha o botão Configure (Configurar) () no canto superior direito da lista de Patches. (Por padrão, a guia Patches exibe colunas para apenas alguns dos metadados de patch disponíveis).

   Para obter informações sobre os tipos de metadados que você pode adicionar à visualização, consulte Patch na Referência de API do AWS Systems Manager.

Como criar e gerenciar grupos de patches

Se não estiver usando políticas de patch em suas operações, você pode organizar suas iniciativas de aplicação de patches, adicionando nós gerenciados aos grupos de patches com o uso de etiquetas.

> **Note**
>
> Os grupos de patches não são usados em operações de aplicação de patches em políticas de patch. Para obter informações sobre como trabalhar com políticas de patch, consulte Configurações de políticas de patches em Quick Setup.
> Não há suporte no console à funcionalidade de grupos de patches para pares de conta-região que ainda não usavam grupos de patches antes do lançamento do suporte à política de patches em 22 de dezembro de 2022. A funcionalidade de grupo de patches ainda está disponível em pares de conta-região que começaram a usar grupos de patches antes dessa data.

Para usar etiquetas em operações de aplicação de patches, é necessário aplicar a chave de etiqueta `Patch Group` ou `PatchGroup` aos nós gerenciados. Também é necessário especificar o nome que você deseja dar ao grupo de patches como o valor da etiqueta. Você pode especificar qualquer valor de tag, mas a chave da tag precisa ser `Patch Group` ou `PatchGroup`.

`PatchGroup` (sem espaço) é obrigatório, se você tiver permissão para etiquetas nos metadados da instância do EC2.

Depois de agrupar os nós gerenciados usando etiquetas, adicione o valor do grupo de patches a uma lista de referência de patches. Ao registrar o grupo de patches em uma linha de base de patch, você garante que os patches corretos sejam instalados durante a aplicação de patches. Para obter mais informações sobre grupos de patches, consulte Grupos de patches.

Conclua as tarefas deste tópico para preparar seus nós gerenciados para aplicação de patches usando etiquetas com seus nós e a lista de referência de patches. A tarefa 1 é necessária somente para aplicar patches em instâncias do Amazon EC2. A tarefa 2 é necessária somente para aplicar patches em instâncias que não sejam do EC2 em um ambiente híbrido e multinuvem. A tarefa 3 é necessária para todos os nós gerenciados.

> **Tip**
>
> Também é possível adicionar tags aos nós gerenciados usando o comando da AWS CLI `add-tags-to-resource` ou a operação de API do Systems Manager ssm-agent-minimum-s3-permissions-required`AddTagsToResource`.

Tarefas

- Tarefa 1: adicionar instâncias do EC2 a um grupo de patches usando tags
- Tarefa 2: Adicionar nós gerenciados a um grupo de patches usando etiquetas
- Tarefa 3: Adicionar um grupo de patches a uma linha de base de patches

Tarefa 1: adicionar instâncias do EC2 a um grupo de patches usando tags

É possível adicionar etiquetas a instâncias do EC2 usando o console do Systems Manager ou o console do Amazon EC2. Essa tarefa é necessária somente para corrigir instâncias do Amazon EC2.

> **Important**
>
> Não é possível aplicar a tag `Patch Group` (com um espaço) a uma instância do Amazon EC2 se a opção Allow tags in instance metadata (Permitir tags em metadados de instância) estiver habilitada na instância. Permitir tags em metadados de instância impede que nomes de chaves de tag contenham espaços. Se você tiver permissão para tags nos metadados da instância do EC2, é necessário usar a chave de tag `PatchGroup` (sem um espaço).

Opção 1: para adicionar instâncias do EC2 a um grupo de patches (console do Systems Manager)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Na lista Instâncias gerenciadas, selecione o ID de uma instância gerenciada do EC2 que você deseja configurar para aplicação de patches. Os IDs de nós para instâncias do EC2 começam com `i-`.

> **Note**
>
> Ao usar o console e a AWS CLI do Amazon EC2, é possível aplicar tags `Key = Patch Group` ou `Key = PatchGroup` a instâncias que ainda não estejam configuradas para uso com o Systems Manager.
> Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

4. Selecione a guia Etiquetas e escolha Editar.
5. Na coluna esquerda, insira **`Patch Group`** ou **`PatchGroup`**. Se você tiver permissão para tags nos metadados da instância do EC2, é necessário usar `PatchGroup` (sem um espaço).
6. Na coluna direita, insira um valor de etiqueta para servir como nome do grupo de patches.
7. Escolha Salvar.
8. Repita esse procedimento para adicionar outras instâncias do EC2 ao mesmo grupo de patches.

Opção 2: para adicionar instâncias do EC2 a um grupo de patches (console do Amazon EC2)

1. Abra o console do Amazon EC2 e depois escolha Instances (Instâncias) no painel de navegação.
2. Na lista de instâncias, escolha uma instância que você deseja configurar para aplicação de patch.
3. No menu Ações, escolha Configurações da instância e, depois, Gerenciar etiquetas.
4. Selecione Adicionar nova tag.
5. Em Key (Chave), insira **`Patch Group`** ou **`PatchGroup`**. Se você tiver permissão para tags nos metadados da instância do EC2, é necessário usar `PatchGroup` (sem um espaço).
6. Em Valor, insira um valor para servir como nome do grupo de patches.
7. Escolha Salvar.
8. Repita esse procedimento para adicionar outras instâncias ao mesmo grupo de patches.

Tarefa 2: Adicionar nós gerenciados a um grupo de patches usando etiquetas

Siga as etapas deste tópico para adicionar etiquetas aos dispositivos principais do AWS IoT Greengrass e aos nós gerenciados ativados para ambiente híbrido que não são do EC2 (mi-*). Essa

tarefa é necessária somente para corrigir instâncias que não sejam do EC2 em um ambiente híbrido e multinuvem.

> **Note**
>
> Não é possível adicionar etiquetas para nós gerenciados que não são do EC2 usando o console do Amazon EC2.

Para adicionar nós gerenciados que não são do EC2 a um grupo de patches (console do Systems Manager)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Na lista Nós gerenciados, escolha o nome do nó gerenciado que você deseja configurar para a aplicação de patch.

   > **Note**
   >
   > Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

4. Selecione a guia Etiquetas e escolha Editar.
5. Na coluna esquerda, insira **`Patch Group`** ou **`PatchGroup`**. Se você tiver permissão para tags nos metadados da instância do EC2, é necessário usar `PatchGroup` (sem um espaço).
6. Na coluna direita, insira um valor de etiqueta para servir como nome do grupo de patches.
7. Escolha Salvar.
8. Repita esse procedimento para adicionar outros nós gerenciados ao mesmo grupo de patches.

Tarefa 3: Adicionar um grupo de patches a uma linha de base de patches

Para associar uma lista de referência de patches específica aos nós gerenciados, você deve adicionar o valor do grupo de patches a essa lista. Ao registrar o grupo de patches em uma linha de base de patch, você pode garantir que os patches corretos sejam instalados durante uma aplicação de patches. Essa tarefa é necessária para aplicar patches em instâncias do EC2, nós gerenciados que não são do EC2 ou ambos.

Para obter mais informações sobre grupos de patches, consulte Grupos de patches.

> **Note**
>
> As etapas a serem seguidas variam se você acessou o Patch Manager pela primeira vez antes ou depois do lançamento das políticas de patch de 22 de dezembro de 2022.

Para adicionar um grupo de patches a uma lista de referência de patches (console do Systems Manager)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Patch Manager.
3. Se você estiver acessando o Patch Manager pela primeira vez na Região da AWS atual, e a página inicial do Patch Manager for aberta, escolha Iniciar com uma visão geral.
4. Escolha a guia Listas de referência de patches e, em Listas de referência de patches, escolha o nome da lista de referência de patches que você deseja configurar para o grupo de patches.

   Se você só acessou o Patch Manager depois do lançamento das políticas de patch, é necessário escolher uma lista de referência personalizada que você criou.
5. Se a página de detalhes ID da lista de referência incluir um menu Ações, faça o seguinte:

   - Escolha Actions (Ações) e depois Modify patch groups (Modificar grupos de patches).
   - Insira o valor da etiqueta que você adicionou aos nós gerenciados e Tarefa 2: Adicionar nós gerenciados a um grupo de patches usando etiquetas escolha Adicionar.

   Se a página de detalhes ID da lista de referência não incluir um menu Ações, não será possível configurar os grupos de patches no console. Em vez disso, você pode fazer qualquer um dos seguintes:

   - (Recomendado) Configure uma política de patch na Quick Setup, uma ferramenta do AWS Systems Manager, para mapear uma lista de referência de patches em uma ou mais instâncias do EC2.

     Para obter mais informações, consulte Using Quick Setup patch policies e Automate organization-wide patching using a Quick Setup patch policy.

- Use o comando register-patch-baseline-for-patch-group na AWS Command Line Interface (AWS CLI) para configurar um grupo de patches.

Integrar o Patch Manager ao AWS Security Hub

O AWS Security Hub fornece uma visão abrangente do seu estado de segurança na AWS. O Security Hub coleta dados de segurança de Contas da AWS, Serviços da AWS e produtos compatíveis de parceiros. Com o Security Hub, você pode verificar o ambiente de acordo com os padrões e as melhores práticas do setor de segurança. O Security Hub ajuda você a analisar suas tendências de segurança e identificar os problemas de segurança de prioridade mais alta.

Usando a integração entre o Patch Manager, uma ferramenta do AWS Systems Manager, e o Security Hub, é possível enviar descobertas sobre nós que estão fora de conformidade do Patch Manager para o Security Hub. Uma descoberta é o registro observável de uma verificação de segurança ou detecção relacionada à segurança. O Security Hub pode então incluir essas descobertas relacionadas a patches na análise feita sobre sua postura de segurança.

As informações nos tópicos a seguir se aplicam independentemente do método ou tipo de configuração que você estiver usando para suas operações de aplicação de patch:

- Uma política de patch configurada no Quick Setup
- Uma opção do Host Management configurada no Quick Setup
- Uma janela de manutenção para executar um patch `Scan` ou tarefa `Install`
- Uma operação Patch now (Aplicar patch agora) sob demanda

Sumário

- Como o Patch Manager envia as descobertas para o Security Hub
  - Tipos de descobertas que o Patch Manager envia
  - Latência para enviar descobertas
  - Repetir quando o Security Hub não estiver disponível
  - Exibir descobertas do no Security Hub
- Descoberta típica do Patch Manager
- Ativar e configurar a integração
- Como parar de enviar descobertas

Como o Patch Manager envia as descobertas para o Security Hub

No Security Hub, os problemas de segurança são rastreados como descobertas. Algumas descobertas provêm de problemas que são detectados por outros Serviços da AWS ou por parceiros terceirizados. O Security Hub também tem um conjunto de regras que ele usa para detectar problemas de segurança e gerar descobertas.

O Patch Manager é uma das ferramentas do Systems Manager que envia descobertas para o Security Hub. Depois de executar uma operação de aplicação de patches executando um documento SSM (`AWS-RunPatchBaseline`, `AWS-RunPatchBaselineAssociation`, ou`AWS-RunPatchBaselineWithHooks`), as informações sobre a aplicação de patches serão enviadas às ferramentas Inventory ou Compliance do AWS Systems Manager ou a ambas. Depois que o Inventário, a Conformidade ou ambos receberem os dados, o Patch Manager receberá uma notificação. Em seguida, o Patch Manager avalia os dados quanto a precisão, formatação e conformidade. Se todas as condições forem cumpridas,Patch Managerencaminha os dados para o Security Hub.

O Security Hub fornece ferramentas para gerenciar descobertas em todas essas fontes. Você pode exibir e filtrar listas de descobertas e exibir detalhes de uma descoberta. Para obter mais informações, consulte Visualizar descobertas no Guia do usuário do AWS Security Hub. Você também pode rastrear o status de uma investigação em uma descoberta. Para obter mais informações, consulte Tomar medidas sobre descobertas no Manual do usuário do AWS Security Hub.

Todas as descobertas no Security Hub usam um formato JSON padrão chamado ASFF (Formato de Descoberta de Segurança da AWS). O ASFF inclui detalhes sobre a origem do problema, os recursos afetados e o status atual da descoberta. Para obter mais informações, consulte AWS Security Finding Format (ASFF) no Manual do usuário do AWS Security Hub.

Tipos de descobertas que o Patch Manager envia

O Patch Manager envia descobertas para o Security Hub usando o AWS Security Finding Format (ASFF). No ASFF, o campo `Types` fornece o tipo de descoberta. As descobertas do Patch Manager podem ter os seguintes valores para `Types`:

- Verificações de software e configuração/Gerenciamento de patches

O Patch Manager envia uma descoberta por nó gerenciado não compatível. A descoberta é relatada com o tipo de recurso `AwsEc2Instance` para que as descobertas possam ser correlacionadas com

outras integrações do Security Hub que relatam tipos de recursos do `AwsEc2Instance`. O Patch Manager somente encaminha uma descoberta para o Security Hub se a operação descobrir que a instância não é compatível. A descoberta inclui os resultados do Resumo do Patch.

> **Note**
>
> Depois de reportar um nó fora de conformidade ao Security Hub. O Patch Manager não envia uma atualização para o Security Hub depois que o nó é colocado em conformidade. É possível resolver manualmente as descobertas no Security Hub após a aplicação dos patches necessários ao nó gerenciado.

Para obter mais informações sobre definições de conformidade, consulte Valores de estados de conformidade de patches. Para obter mais informações sobre o `PatchSummary`, consulte Resumo de Patchesna Referência de API do AWS Security Hub.

Latência para enviar descobertas

Quando o Patch Manager cria uma nova descoberta, ela normalmente é enviada para o Security Hub dentro de alguns segundos a duas horas. A velocidade depende do tráfego na Região da AWS sendo processado naquele momento.

Repetir quando o Security Hub não estiver disponível

Se houver uma interrupção do serviço, uma função AWS Lambda será executada para colocar as mensagens de volta na fila principal depois que o serviço estiver sendo executado novamente. Depois que as mensagens estiverem na fila principal, a tentativa será automática.

Se o Security Hub não estiver disponível, o Patch Manager tentará enviar as descobertas novamente até que sejam recebidas.

Exibir descobertas do no Security Hub

Este procedimento descreve como visualizar as descobertas no Security Hub sobre nós gerenciados da frota que estão fora de conformidade com os patches.

Para analisar as descobertas do Security Hub quanto à conformidade de patches

1. Faça login no AWS Management Console e abra o console do AWS Security Hub em https://console.aws.amazon.com/securityhub/.
2. No painel de navegação, selecione Descobertas.

3. Escolha a caixa Adicionar filtros ().
4. No menu, em Filtros, escolha Nome do produto.
5. Na caixa de diálogo que é aberta, escolha é no primeiro campo e insira **`Systems Manager Patch Manager`** no segundo campo.
6. Escolha Aplicar.
7. Adicione quaisquer filtros que você quiser para ajudar a restringir os resultados.
8. Na lista de resultados, escolha o título da descoberta sobre a qual você deseja ver mais informações.

   Um painel é exibido no lado direito da tela com mais detalhes sobre o recurso, o problema descoberto e uma correção recomendada.

   > **Important**
   >
   > No momento, o Security Hub relata como `EC2 Instance` o tipo de recurso de todos os nós gerenciados. Isso inclui servidores on-premises e máquinas virtuais (VMs) registrados para uso com o Systems Manager.

Classificações de gravidade

A lista de descobertas do **`Systems Manager Patch Manager`** inclui um relatório da gravidade da descoberta. Os níveis de gravidade incluem este, do mais baixo ao mais alto:

- INFORMATIVO: nenhum problema foi encontrado.
- BAIXO: o problema não requer correção.
- MÉDIA: o problema deve ser solucionado, mas não é urgente.
- ALTA: o problema deve ser tratado como prioridade.
- CRÍTICA: o problema deve ser corrigido imediatamente para evitar que seja escalonado.

A gravidade é determinada pelo pacote fora de conformidade mais grave da instância. Como é possível ter várias listas de referência de patches com vários níveis de gravidade, a gravidade mais alta é relatada de todos os pacotes fora de conformidade. Por exemplo, suponha que você tenha dois pacotes fora de conformidade em que a gravidade do pacote A seja “Crítica” e a gravidade do pacote B seja “Baixa”. “Crítica” será relatada como a gravidade.

O campo da gravidade está diretamente correlacionado com o campo `Compliance` do Patch Manager. É um campo que você define para atribuir a patches individuais que correspondem à regra. Como esse campo `Compliance` é atribuído a patches individuais, ele não é refletido no nível de resumo do patch.

Conteúdo relacionado

- Findings no Guia do usuário do AWS Security Hub
- Conformidade de patches de várias contas com o Patch Manager e o Security Hub no Blog de gerenciamento e governança da AWS

Descoberta típica do Patch Manager

O Patch Manager envia descobertas para o Security Hub por meio do AWS Security Finding Format (ASFF).

Aqui está um exemplo de uma descoberta típica do Patch Manager.


```
{
  "SchemaVersion": "2018-10-08",
  "Id": "arn:aws:patchmanager:us-east-2:111122223333:instance/i-02573cafcfEXAMPLE/
document/AWS-RunPatchBaseline/run-command/d710f5bd-04e3-47b4-82f6-df4e0EXAMPLE",
  "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/ssm-patch-manager",
  "GeneratorId": "d710f5bd-04e3-47b4-82f6-df4e0EXAMPLE",
  "AwsAccountId": "111122223333",
  "Types": [
    "Software & Configuration Checks/Patch Management/Compliance"
  ],
  "CreatedAt": "2021-11-11T22:05:25Z",
  "UpdatedAt": "2021-11-11T22:05:25Z",
  "Severity": {
    "Label": "INFORMATIONAL",
    "Normalized": 0
  },
  "Title": "Systems Manager Patch Summary - Managed Instance Non-Compliant",
  "Description": "This AWS control checks whether each instance that is managed by AWS
 Systems Manager is in compliance with the rules of the patch baseline that applies to
 that instance when a compliance Scan runs.",
  "Remediation": {
    "Recommendation": {
      "Text": "For information about bringing instances into patch compliance, see
 'Remediating out-of-compliance instances (Patch Manager)'.",
```

```
      "Url": "https://docs.aws.amazon.com/systems-manager/latest/userguide/patch-
compliance-remediation.html"
    }
  },
  "SourceUrl": "https://us-east-2.console.aws.amazon.com/systems-manager/fleet-manager/
i-02573cafcfEXAMPLE/patch?region=us-east-2",
  "ProductFields": {
    "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-2::product/aws/ssm-
patch-manager/arn:aws:patchmanager:us-east-2:111122223333:instance/i-02573cafcfEXAMPLE/
document/AWS-RunPatchBaseline/run-command/d710f5bd-04e3-47b4-82f6-df4e0EXAMPLE",
    "aws/securityhub/ProductName": "Systems Manager Patch Manager",
    "aws/securityhub/CompanyName": "AWS"
  },
  "Resources": [
    {
      "Type": "AwsEc2Instance",
      "Id": "i-02573cafcfEXAMPLE",
      "Partition": "aws",
      "Region": "us-east-2"
    }
  ],
  "WorkflowState": "NEW",
  "Workflow": {
    "Status": "NEW"
  },
  "RecordState": "ACTIVE",
  "PatchSummary": {
    "Id": "pb-0c10e65780EXAMPLE",
    "InstalledCount": 45,
    "MissingCount": 2,
    "FailedCount": 0,
    "InstalledOtherCount": 396,
    "InstalledRejectedCount": 0,
    "InstalledPendingReboot": 0,
    "OperationStartTime": "2021-11-11T22:05:06Z",
    "OperationEndTime": "2021-11-11T22:05:25Z",
    "RebootOption": "NoReboot",
    "Operation": "SCAN"
  }
}
```

Ativar e configurar a integração

Para usar o Patch Manager com o Security Hub, é necessário ativar o Security Hub. Para obter informações sobre como habilitar o Hub de segurança, consulte Configurar o Security Hub no Guia do usuário do AWS Security Hub.

O procedimento a seguir descreve como integrar o Patch Manager e o Security Hub quando o Security Hub já está ativo, mas a integração com o Patch Manager está desativada. Você só precisa concluir este procedimento se a integração foi desativada manualmente.

Para adicionar o Patch Manager à integração com o Security Hub

1. No painel de navegação, escolha Patch Manager.
2. Escolha a guia Configurações.

   - ou -

   Se estiver acessando o Patch Manager pela primeira vez na Região da AWS atual, escolha Start with an overview (Começar com uma visão geral) e, em seguida, escolha a guia Settings (Configurações).
3. Na seção Exportar para Security Hub, à direita de As descobertas de conformidade de patches não estão sendo exportadas para o Security Hub, escolha Habilitar.

Como parar de enviar descobertas

Para parar de enviar descobertas para o Security Hub, você pode usar o console ou a API do Security Hub.

Para obter mais informações, consulte os seguintes tópicos no Guia do usuário do AWS Security Hub:

- Desabilitar e habilitar o fluxo de descobertas em uma integração (console)
- Desabilitar e habilitar o fluxo de descobertas em uma integração (Security Hub, API, AWS CLI)

Trabalhar com recursos do Patch Manager usando a AWS CLI

Esta seção inclui exemplos de comandos da AWS Command Line Interface (AWS CLI) que você pode usar para realizar tarefas de configuração do Patch Manager, uma ferramenta do AWS Systems Manager.

Para obter uma ilustração do uso da AWS CLI para aplicar um patch a um ambiente de servidor usando a linha de base de patch personalizada, consulte Tutorial: Aplicar patches a um ambiente de servidor usando a AWS CLI.

Para obter mais informações sobre como usar a AWS CLI para tarefas do AWS Systems Manager, consulte a Seção do AWS Systems Manager da Referência de comando da AWS CLI.

Tópicos

- Comandos da AWS CLI para linhas de base de patches
- Comandos da AWS CLI para grupos de patches
- Comandos da AWS CLI para visualizar resumos e detalhes de patches
- Comandos da AWS CLI para verificação e correção de nós gerenciados

Comandos da AWS CLI para linhas de base de patches

Exemplo de comandos para linhas de base de patch

- Criar uma linha de base de patch
- Criar uma linha de base de patch com repositórios personalizados para diferentes versões do SO
- Atualizar uma linha de base de patch
- Renomear uma linha de base de patch
- Excluir uma linha de base de patch
- Listar todas as linhas de base de patch
- Listar todas as listas de referência de patches fornecidas pela AWS
- Listar minhas linhas de base de patches
- Exibir uma linha de base de patch
- Obter a linha de base padrão de patch
- Definir uma linha de base de patch personalizada como padrão
- Redefinir uma lista de referência de patches da AWS como padrão
- Marcar uma linha de base de patch
- Listar as tags para uma linha de base de patch
- Remover uma tag de uma linha de base de patch

Criar uma linha de base de patch

O comando a seguir cria uma lista de referência de patches que aprova todas as atualizações de segurança críticas e importantes para o Windows Server 2012 R2 cinco dias após o lançamento. Os patches também foram especificados para as listas de patches Approved (Aprovado) e Rejected (Rejeitado). Além disso, a linha de base de patch foi marcada para indicar que é para um ambiente de produção.

Linux & macOS

```
aws ssm create-patch-baseline \
    --name "Windows-Server-2012R2" \
    --tags "Key=Environment,Value=Production" \
    --description "Windows Server 2012 R2, Important and Critical security updates"
 \
    --approved-patches "KB2032276,MS10-048" \
    --rejected-patches "KB2124261" \
    --rejected-patches-action "ALLOW_AS_DEPENDENCY" \
    --approval-rules
 "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Important,Critical
{Key=CLASSIFICATION,Values=SecurityUpdates},
{Key=PRODUCT,Values=WindowsServer2012R2}]},ApproveAfterDays=5}]"
```

Windows Server

```
aws ssm create-patch-baseline ^
    --name "Windows-Server-2012R2" ^
    --tags "Key=Environment,Value=Production" ^
    --description "Windows Server 2012 R2, Important and Critical security updates"
 ^
    --approved-patches "KB2032276,MS10-048" ^
    --rejected-patches "KB2124261" ^
    --rejected-patches-action "ALLOW_AS_DEPENDENCY" ^
    --approval-rules
 "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Important,Critical
{Key=CLASSIFICATION,Values=SecurityUpdates},
{Key=PRODUCT,Values=WindowsServer2012R2}]},ApproveAfterDays=5}]"
```

O sistema retorna informações como estas.

```
{
```

```
    "BaselineId":"pb-0c10e65780EXAMPLE"
}
```

Criar uma linha de base de patch com repositórios personalizados para diferentes versões do SO

Aplica-se somente aos nós gerenciados do Linux. O comando a seguir mostra como especificar o repositório de patches a ser usado para uma determinada versão do sistema operacional Amazon Linux. Este exemplo usa um repositório de origem ativado por padrão no Amazon Linux 2017.09, mas pode ser adaptado para outro repositório de origem que você tenha configurado para um nó gerenciado.

> **Note**
>
> Para demonstrar melhor este comando mais complexo, estamos usando a opção `--cli-input-json` com opções adicionais armazenadas em um arquivo JSON externo.

1. Crie um arquivo JSON com um nome como `my-patch-repository.json` e adicione o seguinte conteúdo a ele:

```
{
    "Description": "My patch repository for Amazon Linux 2017.09",
    "Name": "Amazon-Linux-2017.09",
    "OperatingSystem": "AMAZON_LINUX",
    "ApprovalRules": {
        "PatchRules": [
            {
                "ApproveAfterDays": 7,
                "EnableNonSecurity": true,
                "PatchFilterGroup": {
                    "PatchFilters": [
                        {
                            "Key": "SEVERITY",
                            "Values": [
                                "Important",
                                "Critical"
                            ]
                        },
                        {
                            "Key": "CLASSIFICATION",
                            "Values": [
```

```
                                "Security",
                                "Bugfix"
                            ]
                        },
                        {
                            "Key": "PRODUCT",
                            "Values": [
                                "AmazonLinux2017.09"
                            ]
                        }
                    ]
                }
            }
        ]
    },
    "Sources": [
        {
            "Name": "My-AL2017.09",
            "Products": [
                "AmazonLinux2017.09"
            ],
            "Configuration": "[amzn-main] \nname=amzn-main-Base
\nmirrorlist=http://repo./$awsregion./$awsdomain//$releasever/main/
mirror.list //nmirrorlist_expire=300//nmetadata_expire=300 \npriority=10
 \nfailovermethod=priority \nfastestmirror_enabled=0 \ngpgcheck=1
 \ngpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-amazon-ga \nenabled=1 \nretries=3
 \ntimeout=5\nreport_instanceid=yes"
        }
    ]
}
```

2. No diretório em que você salvou o arquivo, execute o seguinte comando:

```
aws ssm create-patch-baseline --cli-input-json file://my-patch-repository.json
```

O sistema retorna informações como estas.

```
{
    "BaselineId": "pb-0c10e65780EXAMPLE"
}
```

Atualizar uma linha de base de patch

O comando a seguir adiciona dois patches como rejeitados e um patch como aprovado para uma linha de base de patch existente.

Para obter mais informações sobre os formatos aceitos de listas de patches aprovados e rejeitados, consulte Sobre formatos de nomes de pacotes para listas de patches aprovados e rejeitados.

Linux & macOS

```
aws ssm update-patch-baseline \
    --baseline-id pb-0c10e65780EXAMPLE \
    --rejected-patches "KB2032276" "MS10-048" \
    --approved-patches "KB2124261"
```

Windows Server

```
aws ssm update-patch-baseline ^
    --baseline-id pb-0c10e65780EXAMPLE ^
    --rejected-patches "KB2032276" "MS10-048" ^
    --approved-patches "KB2124261"
```

O sistema retorna informações como estas.

```
{
   "BaselineId":"pb-0c10e65780EXAMPLE",
   "Name":"Windows-Server-2012R2",
   "RejectedPatches":[
      "KB2032276",
      "MS10-048"
   ],
   "GlobalFilters":{
      "PatchFilters":[

      ]
   },
   "ApprovalRules":{
      "PatchRules":[
         {
            "PatchFilterGroup":{
               "PatchFilters":[
                  {
```

```
                    "Values":[
                       "Important",
                       "Critical"
                    ],
                    "Key":"MSRC_SEVERITY"
                 },
                 {
                    "Values":[
                       "SecurityUpdates"
                    ],
                    "Key":"CLASSIFICATION"
                 },
                 {
                    "Values":[
                       "WindowsServer2012R2"
                    ],
                    "Key":"PRODUCT"
                 }
              ]
           },
           "ApproveAfterDays":5
        }
     ]
  },
  "ModifiedDate":1481001494.035,
  "CreatedDate":1480997823.81,
  "ApprovedPatches":[
     "KB2124261"
  ],
  "Description":"Windows Server 2012 R2, Important and Critical security updates"
}
```

Renomear uma linha de base de patch

Linux & macOS

```
aws ssm update-patch-baseline \
    --baseline-id pb-0c10e65780EXAMPLE \
    --name "Windows-Server-2012-R2-Important-and-Critical-Security-Updates"
```

Windows Server

```
aws ssm update-patch-baseline ^
```

```
    --baseline-id pb-0c10e65780EXAMPLE ^
    --name "Windows-Server-2012-R2-Important-and-Critical-Security-Updates"
```

O sistema retorna informações como estas.

```
{
   "BaselineId":"pb-0c10e65780EXAMPLE",
   "Name":"Windows-Server-2012-R2-Important-and-Critical-Security-Updates",
   "RejectedPatches":[
      "KB2032276",
      "MS10-048"
   ],
   "GlobalFilters":{
      "PatchFilters":[

      ]
   },
   "ApprovalRules":{
      "PatchRules":[
         {
            "PatchFilterGroup":{
               "PatchFilters":[
                  {
                     "Values":[
                        "Important",
                        "Critical"
                     ],
                     "Key":"MSRC_SEVERITY"
                  },
                  {
                     "Values":[
                        "SecurityUpdates"
                     ],
                     "Key":"CLASSIFICATION"
                  },
                  {
                     "Values":[
                        "WindowsServer2012R2"
                     ],
                     "Key":"PRODUCT"
                  }
               ]
```

```
            },
            "ApproveAfterDays":5
         }
      ]
   },
   "ModifiedDate":1481001795.287,
   "CreatedDate":1480997823.81,
   "ApprovedPatches":[
      "KB2124261"
   ],
   "Description":"Windows Server 2012 R2, Important and Critical security updates"
}
```

Excluir uma linha de base de patch

```
aws ssm delete-patch-baseline --baseline-id "pb-0c10e65780EXAMPLE"
```

O sistema retorna informações como estas.

```
{
   "BaselineId":"pb-0c10e65780EXAMPLE"
}
```

Listar todas as linhas de base de patch

```
aws ssm describe-patch-baselines
```

O sistema retorna informações como estas.

```
{
   "BaselineIdentities":[
      {
         "BaselineName":"AWS-DefaultPatchBaseline",
         "DefaultBaseline":true,
         "BaselineDescription":"Default Patch Baseline Provided by AWS.",
         "BaselineId":"arn:aws:ssm:us-east-2:111122223333:patchbaseline/
pb-0c10e65780EXAMPLE"
      },
      {
         "BaselineName":"Windows-Server-2012R2",
         "DefaultBaseline":false,
```

```
         "BaselineDescription":"Windows Server 2012 R2, Important and Critical security
 updates",
         "BaselineId":"pb-0c10e65780EXAMPLE"
      }
   ]
}
```

Este é um exemplo de outro comando que lista todas as listas de referência de patch em uma Região da AWS.

Linux & macOS

```
aws ssm describe-patch-baselines \
    --region us-east-2 \
    --filters "Key=OWNER,Values=[All]"
```

Windows Server

```
aws ssm describe-patch-baselines ^
    --region us-east-2 ^
    --filters "Key=OWNER,Values=[All]"
```

O sistema retorna informações como estas.

```
{
   "BaselineIdentities":[
      {
         "BaselineName":"AWS-DefaultPatchBaseline",
         "DefaultBaseline":true,
         "BaselineDescription":"Default Patch Baseline Provided by AWS.",
         "BaselineId":"arn:aws:ssm:us-east-2:111122223333:patchbaseline/
pb-0c10e65780EXAMPLE"
      },
      {
         "BaselineName":"Windows-Server-2012R2",
         "DefaultBaseline":false,
         "BaselineDescription":"Windows Server 2012 R2, Important and Critical security
 updates",
         "BaselineId":"pb-0c10e65780EXAMPLE"
      }
   ]
```

```
}
```

Listar todas as listas de referência de patches fornecidas pela AWS

Linux & macOS

```
aws ssm describe-patch-baselines \
    --region us-east-2 \
    --filters "Key=OWNER,Values=[AWS]"
```

Windows Server

```
aws ssm describe-patch-baselines ^
    --region us-east-2 ^
    --filters "Key=OWNER,Values=[AWS]"
```

O sistema retorna informações como estas.

```
{
   "BaselineIdentities":[
      {
         "BaselineName":"AWS-DefaultPatchBaseline",
         "DefaultBaseline":true,
         "BaselineDescription":"Default Patch Baseline Provided by AWS.",
         "BaselineId":"arn:aws:ssm:us-east-2:111122223333:patchbaseline/
pb-0c10e65780EXAMPLE"
      }
   ]
}
```

Listar minhas linhas de base de patches

Linux & macOS

```
aws ssm describe-patch-baselines \
    --region us-east-2 \
    --filters "Key=OWNER,Values=[Self]"
```

Windows Server

```
aws ssm describe-patch-baselines ^
```

```
    --region us-east-2 ^
    --filters "Key=OWNER,Values=[Self]"
```

O sistema retorna informações como estas.

```
{
   "BaselineIdentities":[
      {
         "BaselineName":"Windows-Server-2012R2",
         "DefaultBaseline":false,
         "BaselineDescription":"Windows Server 2012 R2, Important and Critical security
 updates",
         "BaselineId":"pb-0c10e65780EXAMPLE"
      }
   ]
}
```

Exibir uma linha de base de patch

```
aws ssm get-patch-baseline --baseline-id pb-0c10e65780EXAMPLE
```

> **Note**
>
> Para listas de referência de patches personalizadas, você pode especificar o ID da lista de referência de patches ou o Amazon Resource Name (ARN) completo. Para a lista de referência de patches fornecida pela AWS, você deve especificar o ARN completo. Por exemplo, `.arn:aws:ssm:us-east-2:075727635805:patchbaseline/pb-0c10e65780EXAMPLE`

O sistema retorna informações como estas.

```
{
   "BaselineId":"pb-0c10e65780EXAMPLE",
   "Name":"Windows-Server-2012R2",
   "PatchGroups":[
      "Web Servers"
   ],
   "RejectedPatches":[
```

```
   ],
   "GlobalFilters":{
      "PatchFilters":[

      ]
   },
   "ApprovalRules":{
      "PatchRules":[
         {
            "PatchFilterGroup":{
               "PatchFilters":[
                  {
                     "Values":[
                        "Important",
                        "Critical"
                     ],
                     "Key":"MSRC_SEVERITY"
                  },
                  {
                     "Values":[
                        "SecurityUpdates"
                     ],
                     "Key":"CLASSIFICATION"
                  },
                  {
                     "Values":[
                        "WindowsServer2012R2"
                     ],
                     "Key":"PRODUCT"
                  }
               ]
            },
            "ApproveAfterDays":5
         }
      ]
   },
   "ModifiedDate":1480997823.81,
   "CreatedDate":1480997823.81,
   "ApprovedPatches":[

   ],
   "Description":"Windows Server 2012 R2, Important and Critical security updates"
}
```

Obter a linha de base padrão de patch

```
aws ssm get-default-patch-baseline --region us-east-2
```

O sistema retorna informações como estas.

```
{
   "BaselineId":"arn:aws:ssm:us-east-2:111122223333:patchbaseline/pb-0c10e65780EXAMPLE"
}
```

Definir uma linha de base de patch personalizada como padrão

Linux & macOS

```
aws ssm register-default-patch-baseline \
    --region us-east-2 \
    --baseline-id "pb-0c10e65780EXAMPLE"
```

Windows Server

```
aws ssm register-default-patch-baseline ^
    --region us-east-2 ^
    --baseline-id "pb-0c10e65780EXAMPLE"
```

O sistema retorna informações como estas.

```
{
   "BaselineId":"pb-0c10e65780EXAMPLE"
}
```

Redefinir uma lista de referência de patches da AWS como padrão

Linux & macOS

```
aws ssm register-default-patch-baseline \
    --region us-east-2 \
    --baseline-id "arn:aws:ssm:us-east-2:123456789012:patchbaseline/
pb-0c10e65780EXAMPLE"
```

Windows Server

```
aws ssm register-default-patch-baseline ^
    --region us-east-2 ^
    --baseline-id "arn:aws:ssm:us-east-2:123456789012:patchbaseline/
pb-0c10e65780EXAMPLE"
```

O sistema retorna informações como estas.

```
{
   "BaselineId":"pb-0c10e65780EXAMPLE"
}
```

Marcar uma linha de base de patch

Linux & macOS

```
aws ssm add-tags-to-resource \
    --resource-type "PatchBaseline" \
    --resource-id "pb-0c10e65780EXAMPLE" \
    --tags "Key=Project,Value=Testing"
```

Windows Server

```
aws ssm add-tags-to-resource ^
    --resource-type "PatchBaseline" ^
    --resource-id "pb-0c10e65780EXAMPLE" ^
    --tags "Key=Project,Value=Testing"
```

Listar as tags para uma linha de base de patch

Linux & macOS

```
aws ssm list-tags-for-resource \
    --resource-type "PatchBaseline" \
    --resource-id "pb-0c10e65780EXAMPLE"
```

Windows Server

```
aws ssm list-tags-for-resource ^
```

```
    --resource-type "PatchBaseline" ^
    --resource-id "pb-0c10e65780EXAMPLE"
```

Remover uma tag de uma linha de base de patch

Linux & macOS

```
aws ssm remove-tags-from-resource \
    --resource-type "PatchBaseline" \
    --resource-id "pb-0c10e65780EXAMPLE" \
    --tag-keys "Project"
```

Windows Server

```
aws ssm remove-tags-from-resource ^
    --resource-type "PatchBaseline" ^
    --resource-id "pb-0c10e65780EXAMPLE" ^
    --tag-keys "Project"
```

Comandos da AWS CLI para grupos de patches

Comandos de exemplo para grupos de patches

- Criar um grupo de patches
- Registrar um grupo de patches "web servers" em uma linha de base de patches
- Registrar um grupo de patches "Backend" na lista de referência de patches fornecida pela AWS
- Exibir registros de grupos de patches
- Cancelar o registro de um grupo de patches de uma linha de base de patch

Criar um grupo de patches

> **Note**
>
> Os grupos de patches não são usados em operações de aplicação de patches em políticas de patch. Para obter informações sobre como trabalhar com políticas de patch, consulte Configurações de políticas de patches em Quick Setup.

Para ajudar você a organizar suas iniciativas de aplicação de patches, recomendamos que você adicione nós gerenciados aos grupos de patches usando etiquetas. Os grupos de patches exigem o uso da chave de tag `Patch Group` ou `PatchGroup`. Se você tiver permissão para tags nos metadados da instância do EC2, é necessário usar `PatchGroup` (sem um espaço). Você pode especificar qualquer valor de tag, mas a chave da tag precisa ser `Patch Group` ou `PatchGroup`. Para obter mais informações sobre grupos de patches, consulte Grupos de patches.

Depois de agrupar os nós gerenciados usando etiquetas, adicione o valor do grupo de patches a uma lista de referência de patches. Ao registrar o grupo de patches em uma linha de base de patch, você garante que os patches corretos sejam instalados durante a aplicação de patches.

Tarefa 1: adicionar instâncias do EC2 a um grupo de patches usando tags

> **Note**
>
> Ao usar o console e a AWS CLI do Amazon Elastic Compute Cloud (Amazon EC2), é possível aplicar as tags `Key = Patch Group` ou `Key = PatchGroup` a instâncias que ainda não estejam configuradas para uso com o Systems Manager. Se uma instância do EC2 que você espere ver no Patch Manager não estiver listada após a aplicação da tag `Patch Group` ou `Key = PatchGroup`, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

Execute o comando a seguir para adicionar a tag `PatchGroup` a uma instância do EC2.

```
aws ec2 create-tags --resources "i-1234567890abcdef0" --tags
 "Key=PatchGroup,Value=GroupValue"
```

Tarefa 2: Adicionar nós gerenciados a um grupo de patches usando etiquetas

Execute o comando a seguir para adicionar a etiqueta `PatchGroup` a um nó gerenciado.

Linux & macOS

```
aws ssm add-tags-to-resource \
    --resource-type "ManagedInstance" \
    --resource-id "mi-0123456789abcdefg" \
    --tags "Key=PatchGroup,Value=GroupValue"
```

Windows Server

```
aws ssm add-tags-to-resource ^
    --resource-type "ManagedInstance" ^
    --resource-id "mi-0123456789abcdefg" ^
    --tags "Key=PatchGroup,Value=GroupValue"
```

Tarefa 3: Adicionar um grupo de patches a uma linha de base de patches

Execute o comando a seguir para associar um valor de tag `PatchGroup` à linha de base de patch especificada.

Linux & macOS

```
aws ssm register-patch-baseline-for-patch-group \
    --baseline-id "pb-0c10e65780EXAMPLE" \
    --patch-group "Development"
```

Windows Server

```
aws ssm register-patch-baseline-for-patch-group ^
    --baseline-id "pb-0c10e65780EXAMPLE" ^
    --patch-group "Development"
```

O sistema retorna informações como estas.

```
{
  "PatchGroup": "Development",
  "BaselineId": "pb-0c10e65780EXAMPLE"
}
```

Registrar um grupo de patches "web servers" em uma linha de base de patches

Linux & macOS

```
aws ssm register-patch-baseline-for-patch-group \
    --baseline-id "pb-0c10e65780EXAMPLE" \
    --patch-group "Web Servers"
```

Windows Server

```
aws ssm register-patch-baseline-for-patch-group ^
    --baseline-id "pb-0c10e65780EXAMPLE" ^
    --patch-group "Web Servers"
```

O sistema retorna informações como estas.

```
{
   "PatchGroup":"Web Servers",
   "BaselineId":"pb-0c10e65780EXAMPLE"
}
```

Registrar um grupo de patches "Backend" na lista de referência de patches fornecida pela AWS

Linux & macOS

```
aws ssm register-patch-baseline-for-patch-group \
    --region us-east-2 \
    --baseline-id "arn:aws:ssm:us-east-2:111122223333:patchbaseline/
pb-0c10e65780EXAMPLE" \
    --patch-group "Backend"
```

Windows Server

```
aws ssm register-patch-baseline-for-patch-group ^
    --region us-east-2 ^
    --baseline-id "arn:aws:ssm:us-east-2:111122223333:patchbaseline/
pb-0c10e65780EXAMPLE" ^
    --patch-group "Backend"
```

O sistema retorna informações como estas.

```
{
   "PatchGroup":"Backend",
   "BaselineId":"arn:aws:ssm:us-east-2:111122223333:patchbaseline/pb-0c10e65780EXAMPLE"
}
```

Exibir registros de grupos de patches

```
aws ssm describe-patch-groups --region us-east-2
```

O sistema retorna informações como estas.

```
{
   "PatchGroupPatchBaselineMappings":[
      {
         "PatchGroup":"Backend",
         "BaselineIdentity":{
            "BaselineName":"AWS-DefaultPatchBaseline",
            "DefaultBaseline":false,
            "BaselineDescription":"Default Patch Baseline Provided by AWS.",
            "BaselineId":"arn:aws:ssm:us-east-2:111122223333:patchbaseline/
pb-0c10e65780EXAMPLE"
         }
      },
      {
         "PatchGroup":"Web Servers",
         "BaselineIdentity":{
            "BaselineName":"Windows-Server-2012R2",
            "DefaultBaseline":true,
            "BaselineDescription":"Windows Server 2012 R2, Important and Critical
 updates",
            "BaselineId":"pb-0c10e65780EXAMPLE"
         }
      }
   ]
}
```

Cancelar o registro de um grupo de patches de uma linha de base de patch

Linux & macOS

```
aws ssm deregister-patch-baseline-for-patch-group \
    --region us-east-2 \
    --patch-group "Production" \
    --baseline-id "arn:aws:ssm:us-east-2:111122223333:patchbaseline/
pb-0c10e65780EXAMPLE"
```

Windows Server

```
aws ssm deregister-patch-baseline-for-patch-group ^
    --region us-east-2 ^
    --patch-group "Production" ^
    --baseline-id "arn:aws:ssm:us-east-2:111122223333:patchbaseline/
pb-0c10e65780EXAMPLE"
```

O sistema retorna informações como estas.

```
{
   "PatchGroup":"Production",
   "BaselineId":"arn:aws:ssm:us-east-2:111122223333:patchbaseline/pb-0c10e65780EXAMPLE"
}
```

Comandos da AWS CLI para visualizar resumos e detalhes de patches

Comandos de exemplo para exibição de resumos e detalhes de patches

- Obter todos os patches definidos por uma linha de base de patch
- Obtenha todos os patches para o AmazonLinux2018.03 que tenha uma classificação de SECURITY e uma gravidade Critical
- Obtenha todos os patches para o Windows Server 2012 que tenham a gravidade MSRC como Critical.
- Obter todos os patches disponíveis
- Obter estados de resumo de patches por nó gerenciado
- Obter detalhes da conformidade de patches para um nó gerenciado
- Visualizar os resultados de conformidade dos patches (AWS CLI)

Obter todos os patches definidos por uma linha de base de patch

> **Note**
>
> Esse comando é compatível com listas de referência de patches do Windows Server.

Linux & macOS

```
aws ssm describe-effective-patches-for-patch-baseline \
    --region us-east-2 \
    --baseline-id "pb-0c10e65780EXAMPLE"
```

Windows Server

```
aws ssm describe-effective-patches-for-patch-baseline ^
    --region us-east-2 ^
    --baseline-id "pb-0c10e65780EXAMPLE"
```

O sistema retorna informações como estas.

```
{
   "NextToken":"--token string truncated--",
   "EffectivePatches":[
      {
         "PatchStatus":{
            "ApprovalDate":1384711200.0,
            "DeploymentStatus":"APPROVED"
         },
         "Patch":{
            "ContentUrl":"https://support.microsoft.com/en-us/kb/2876331",
            "ProductFamily":"Windows",
            "Product":"WindowsServer2012R2",
            "Vendor":"Microsoft",
            "Description":"A security issue has been identified in a Microsoft
 software
               product that could affect your system. You can help protect your system
               by installing this update from Microsoft. For a complete listing of the
               issues that are included in this update, see the associated Microsoft
               Knowledge Base article. After you install this update, you may have to
               restart your system.",
            "Classification":"SecurityUpdates",
            "Title":"Security Update for Windows Server 2012 R2 Preview (KB2876331)",
            "ReleaseDate":1384279200.0,
            "MsrcClassification":"Critical",
            "Language":"All",
            "KbNumber":"KB2876331",
            "MsrcNumber":"MS13-089",
            "Id":"e74ccc76-85f0-4881-a738-59e9fc9a336d"
```

```
        }
      },
      {
         "PatchStatus":{
            "ApprovalDate":1428858000.0,
            "DeploymentStatus":"APPROVED"
         },
         "Patch":{
            "ContentUrl":"https://support.microsoft.com/en-us/kb/2919355",
            "ProductFamily":"Windows",
            "Product":"WindowsServer2012R2",
            "Vendor":"Microsoft",
            "Description":"Windows Server 2012 R2 Update is a cumulative
               set of security updates, critical updates and updates. You
               must install Windows Server 2012 R2 Update to ensure that
               your computer can continue to receive future Windows Updates,
               including security updates. For a complete listing of the
               issues that are included in this update, see the associated
               Microsoft Knowledge Base article for more information. After
               you install this item, you may have to restart your computer.",
            "Classification":"SecurityUpdates",
            "Title":"Windows Server 2012 R2 Update (KB2919355)",
            "ReleaseDate":1428426000.0,
            "MsrcClassification":"Critical",
            "Language":"All",
            "KbNumber":"KB2919355",
            "MsrcNumber":"MS14-018",
            "Id":"8452bac0-bf53-4fbd-915d-499de08c338b"
         }
      }
     ---output truncated---
```

Obtenha todos os patches para o AmazonLinux2018.03 que tenha uma classificação de **`SECURITY`** e uma gravidade **`Critical`**

Linux & macOS

```
aws ssm describe-available-patches \
    --region us-east-2 \
    --filters Key=PRODUCT,Values=AmazonLinux2018.03 Key=SEVERITY,Values=Critical
```

Windows Server

```
aws ssm describe-available-patches ^
    --region us-east-2 ^
    --filters Key=PRODUCT,Values=AmazonLinux2018.03 Key=SEVERITY,Values=Critical
```

O sistema retorna informações como estas.

```
{
    "Patches": [
        {
            "AdvisoryIds": ["ALAS-2011-1"],
            "BugzillaIds": [ "1234567" ],
            "Classification": "SECURITY",
            "CVEIds": [ "CVE-2011-3192"],
            "Name": "zziplib",
            "Epoch": "0",
            "Version": "2.71",
            "Release": "1.3.amzn1",
            "Arch": "i686",
            "Product": "AmazonLinux2018.03",
            "ReleaseDate": 1590519815,
            "Severity": "CRITICAL"
        }
    ]
}
---output truncated---
```

Obtenha todos os patches para o Windows Server 2012 que tenham a gravidade MSRC como **`Critical`**.

Linux & macOS

```
aws ssm describe-available-patches \
    --region us-east-2 \
    --filters Key=PRODUCT,Values=WindowsServer2012 Key=MSRC_SEVERITY,Values=Critical
```

Windows Server

```
aws ssm describe-available-patches ^
    --region us-east-2 ^
```

```
    --filters Key=PRODUCT,Values=WindowsServer2012 Key=MSRC_SEVERITY,Values=Critical
```

O sistema retorna informações como estas.


```
{
   "Patches":[
      {
         "ContentUrl":"https://support.microsoft.com/en-us/kb/2727528",
         "ProductFamily":"Windows",
         "Product":"WindowsServer2012",
         "Vendor":"Microsoft",
         "Description":"A security issue has been identified that could
           allow an unauthenticated remote attacker to compromise your
           system and gain control over it. You can help protect your
           system by installing this update from Microsoft. After you
           install this update, you may have to restart your system.",
         "Classification":"SecurityUpdates",
         "Title":"Security Update for Windows Server 2012 (KB2727528)",
         "ReleaseDate":1352829600.0,
         "MsrcClassification":"Critical",
         "Language":"All",
         "KbNumber":"KB2727528",
         "MsrcNumber":"MS12-072",
         "Id":"1eb507be-2040-4eeb-803d-abc55700b715"
      },
      {
         "ContentUrl":"https://support.microsoft.com/en-us/kb/2729462",
         "ProductFamily":"Windows",
         "Product":"WindowsServer2012",
         "Vendor":"Microsoft",
         "Description":"A security issue has been identified that could
           allow an unauthenticated remote attacker to compromise your
           system and gain control over it. You can help protect your
           system by installing this update from Microsoft. After you
           install this update, you may have to restart your system.",
         "Classification":"SecurityUpdates",
         "Title":"Security Update for Microsoft .NET Framework 3.5 on
           Windows 8 and Windows Server 2012 for x64-based Systems (KB2729462)",
         "ReleaseDate":1352829600.0,
         "MsrcClassification":"Critical",
         "Language":"All",
         "KbNumber":"KB2729462",
```

```
         "MsrcNumber":"MS12-074",
         "Id":"af873760-c97c-4088-ab7e-5219e120eab4"
      }

---output truncated---
```

Obter todos os patches disponíveis

```
aws ssm describe-available-patches --region us-east-2
```

O sistema retorna informações como estas.

```
{
   "NextToken":"--token string truncated--",
   "Patches":[
      {
         "ContentUrl":"https://support.microsoft.com/en-us/kb/2032276",
         "ProductFamily":"Windows",
         "Product":"WindowsServer2008R2",
         "Vendor":"Microsoft",
         "Description":"A security issue has been identified that could allow an
           unauthenticated remote attacker to compromise your system and gain
           control over it. You can help protect your system by installing this
           update from Microsoft. After you install this update, you may have to
           restart your system.",
         "Classification":"SecurityUpdates",
         "Title":"Security Update for Windows Server 2008 R2 x64 Edition (KB2032276)",
         "ReleaseDate":1279040400.0,
         "MsrcClassification":"Important",
         "Language":"All",
         "KbNumber":"KB2032276",
         "MsrcNumber":"MS10-043",
         "Id":"8692029b-a3a2-4a87-a73b-8ea881b4b4d6"
      },
      {
         "ContentUrl":"https://support.microsoft.com/en-us/kb/2124261",
         "ProductFamily":"Windows",
         "Product":"Windows7",
         "Vendor":"Microsoft",
         "Description":"A security issue has been identified that could allow
           an unauthenticated remote attacker to compromise your system and gain
           control over it. You can help protect your system by installing this
           update from Microsoft. After you install this update, you may have
```

```
          to restart your system.",
         "Classification":"SecurityUpdates",
         "Title":"Security Update for Windows 7 (KB2124261)",
         "ReleaseDate":1284483600.0,
         "MsrcClassification":"Important",
         "Language":"All",
         "KbNumber":"KB2124261",
         "MsrcNumber":"MS10-065",
         "Id":"12ef1bed-0dd2-4633-b3ac-60888aa8ba33"
      }
      ---output truncated---
```

Obter estados de resumo de patches por nó gerenciado

O resumo por nó gerenciado fornece uma série de patches nos seguintes estados por nó: "NotApplicable", "Missing", "Failed", "InstalledOther" e "Installed".

Linux & macOS

```
aws ssm describe-instance-patch-states \
    --instance-ids i-08ee91c0b17045407 i-09a618aec652973a9
```

Windows Server

```
aws ssm describe-instance-patch-states ^
    --instance-ids i-08ee91c0b17045407 i-09a618aec652973a9
```

O sistema retorna informações como estas.

```
{
   "InstancePatchStates":[
      {
            "InstanceId": "i-08ee91c0b17045407",
            "PatchGroup": "",
            "BaselineId": "pb-0c10e65780EXAMPLE",
            "SnapshotId": "6d03d6c5-f79d-41d0-8d0e-00a9aEXAMPLE",
            "InstalledCount": 50,
            "InstalledOtherCount": 353,
            "InstalledPendingRebootCount": 0,
            "InstalledRejectedCount": 0,
            "MissingCount": 0,
```

```
            "FailedCount": 0,
            "UnreportedNotApplicableCount": -1,
            "NotApplicableCount": 671,
            "OperationStartTime": "2020-01-24T12:37:56-08:00",
            "OperationEndTime": "2020-01-24T12:37:59-08:00",
            "Operation": "Scan",
            "RebootOption": "NoReboot"
        },
        {
            "InstanceId": "i-09a618aec652973a9",
            "PatchGroup": "",
            "BaselineId": "pb-0c10e65780EXAMPLE",
            "SnapshotId": "c7e0441b-1eae-411b-8aa7-973e6EXAMPLE",
            "InstalledCount": 36,
            "InstalledOtherCount": 396,
            "InstalledPendingRebootCount": 0,
            "InstalledRejectedCount": 0,
            "MissingCount": 3,
            "FailedCount": 0,
            "UnreportedNotApplicableCount": -1,
            "NotApplicableCount": 420,
            "OperationStartTime": "2020-01-24T12:37:34-08:00",
            "OperationEndTime": "2020-01-24T12:37:37-08:00",
            "Operation": "Scan",
            "RebootOption": "NoReboot"
        }
     ---output truncated---
```

Obter detalhes da conformidade de patches para um nó gerenciado

```
aws ssm describe-instance-patches --instance-id i-08ee91c0b17045407
```

O sistema retorna informações como estas.

```
{
   "NextToken":"--token string truncated--",
   "Patches":[
      {
            "Title": "bind-libs.x86_64:32:9.8.2-0.68.rc1.60.amzn1",
            "KBId": "bind-libs.x86_64",
            "Classification": "Security",
            "Severity": "Important",
            "State": "Installed",
```

```
            "InstalledTime": "2019-08-26T11:05:24-07:00"
        },
        {
            "Title": "bind-utils.x86_64:32:9.8.2-0.68.rc1.60.amzn1",
            "KBId": "bind-utils.x86_64",
            "Classification": "Security",
            "Severity": "Important",
            "State": "Installed",
            "InstalledTime": "2019-08-26T11:05:32-07:00"
        },
        {
            "Title": "dhclient.x86_64:12:4.1.1-53.P1.28.amzn1",
            "KBId": "dhclient.x86_64",
            "Classification": "Security",
            "Severity": "Important",
            "State": "Installed",
            "InstalledTime": "2019-08-26T11:05:31-07:00"
        },
    ---output truncated---
```

Visualizar os resultados de conformidade dos patches (AWS CLI)

Para visualizar os resultados de conformidade de patches para um único nó gerenciado

Execute o seguinte comando na AWS Command Line Interface (AWS CLI) para exibir os resultados de conformidade do patch para um único nó gerenciado.

```
aws ssm describe-instance-patch-states --instance-id instance-id
```

Substitua *instance-id* pelo ID do nó gerenciado para o qual você deseja visualizar os resultados, no formato `i-02573cafcfEXAMPLE` ou `mi-0282f7c436EXAMPLE`.

O sistema retorna informações como as seguintes.

```
{
    "InstancePatchStates": [
        {
            "InstanceId": "i-02573cafcfEXAMPLE",
            "PatchGroup": "mypatchgroup",
            "BaselineId": "pb-0c10e65780EXAMPLE",
            "SnapshotId": "a3f5ff34-9bc4-4d2c-a665-4d1c1EXAMPLE",
            "CriticalNonCompliantCount": 2,
            "SecurityNonCompliantCount": 2,
```

```
            "OtherNonCompliantCount": 1,
            "InstalledCount": 123,
            "InstalledOtherCount": 334,
            "InstalledPendingRebootCount": 0,
            "InstalledRejectedCount": 0,
            "MissingCount": 1,
            "FailedCount": 2,
            "UnreportedNotApplicableCount": 11,
            "NotApplicableCount": 2063,
            "OperationStartTime": "2021-05-03T11:00:56-07:00",
            "OperationEndTime": "2021-05-03T11:01:09-07:00",
            "Operation": "Scan",
            "LastNoRebootInstallOperationTime": "2020-06-14T12:17:41-07:00",
            "RebootOption": "RebootIfNeeded"
        }
    ]
}
```

Para visualizar um resumo da contagem de patches para todas as instâncias do EC2 em uma região

O`describe-instance-patch-states`O oferece suporte à recuperação de resultados para apenas uma instância gerenciada por vez. No entanto, usando um script personalizado com o`describe-instance-patch-states`, você pode gerar um relatório mais granular.

Por exemplo, se a ferramenta de filtro jq estiver instalada na máquina local, você poderá executar o seguinte comando para identificar qual de suas instâncias do EC2 em uma determinada Região da AWS tem um status `InstalledPendingReboot`:

```
aws ssm describe-instance-patch-states \
    --instance-ids $(aws ec2 describe-instances --region region | jq
 '.Reservations[].Instances[] | .InstanceId' | tr '\n|"' ' ') \
    --output text --query 'InstancePatchStates[*].{Instance:InstanceId,
 InstalledPendingRebootCount:InstalledPendingRebootCount}'
```

A *`região`* representa o identificador da região para uma região da Região da AWS compatível com o AWS Systems Manager, como `us-east-2` para a região Leste dos EUA (Ohio). Para ver uma lista dos valores de *`região`* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

Por exemplo:

```
aws ssm describe-instance-patch-states \
```

```
    --instance-ids $(aws ec2 describe-instances --region us-east-2 | jq
'.Reservations[].Instances[] | .InstanceId' | tr '\n|"' ' ') \
    --output text --query 'InstancePatchStates[*].{Instance:InstanceId,
InstalledPendingRebootCount:InstalledPendingRebootCount}'
```

O sistema retorna informações como estas.

```
1       i-02573cafcfEXAMPLE
0       i-0471e04240EXAMPLE
3       i-07782c72faEXAMPLE
6       i-083b678d37EXAMPLE
0       i-03a530a2d4EXAMPLE
1       i-01f68df0d0EXAMPLE
0       i-0a39c0f214EXAMPLE
7       i-0903a5101eEXAMPLE
7       i-03823c2fedEXAMPLE
```

Além de`InstalledPendingRebootCount`, a lista de tipos de contagem que você pode pesquisar inclui o seguinte:

- `CriticalNonCompliantCount`
- `SecurityNonCompliantCount`
- `OtherNonCompliantCount`
- `UnreportedNotApplicableCount`
- `InstalledPendingRebootCount`
- `FailedCount`
- `NotApplicableCount`
- `InstalledRejectedCount`
- `InstalledOtherCount`
- `MissingCount`
- `InstalledCount`

Comandos da AWS CLI para verificação e correção de nós gerenciados

Depois de executar os seguintes comandos para verificar a conformidade do patch ou instalar patches, você pode usar comandos no Comandos da AWS CLI para visualizar resumos e detalhes de patches para exibir informações sobre o status e a conformidade do patch.

Exemplos de comandos

- Verificar a conformidade dos patches (AWS CLI) em nós gerenciados
- Instalar patches em nós gerenciados (AWS CLI)

Verificar a conformidade dos patches (AWS CLI) em nós gerenciados

Para verificar a conformidade dos patches em nós gerenciados

Execute o seguinte comando:

Linux & macOS

```
aws ssm send-command \
    --document-name 'AWS-RunPatchBaseline' \
    --targets Key=InstanceIds,Values='i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE' \
    --parameters 'Operation=Scan' \
    --timeout-seconds 600
```

Windows Server

```
aws ssm send-command ^
    --document-name "AWS-RunPatchBaseline" ^
    --targets Key=InstanceIds,Values="i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE" ^
    --parameters "Operation=Scan" ^
    --timeout-seconds 600
```

O sistema retorna informações como estas.

```
{
    "Command": {
        "CommandId": "a04ed06c-8545-40f4-87c2-a0babEXAMPLE",
        "DocumentName": "AWS-RunPatchBaseline",
        "DocumentVersion": "$DEFAULT",
        "Comment": "",
        "ExpiresAfter": 1621974475.267,
        "Parameters": {
            "Operation": [
                "Scan"
            ]
        },
```

```
        "InstanceIds": [],
        "Targets": [
            {
                "Key": "InstanceIds",
                "Values": [
                    "i-02573cafcfEXAMPLE,
                     i-0471e04240EXAMPLE"
                ]
            }
        ],
        "RequestedDateTime": 1621952275.267,
        "Status": "Pending",
        "StatusDetails": "Pending",
        "TimeoutSeconds": 600,

    ---output truncated---

    }
}
```

Para verificar os nós gerenciados quanto à conformidade do patch por etiqueta do grupo de patches

Execute o seguinte comando:

Linux & macOS

```
aws ssm send-command \
    --document-name 'AWS-RunPatchBaseline' \
    --targets Key='tag:PatchGroup',Values='Web servers' \
    --parameters 'Operation=Scan' \
    --timeout-seconds 600
```

Windows Server

```
aws ssm send-command ^
    --document-name "AWS-RunPatchBaseline" ^
    --targets Key="tag:PatchGroup",Values="Web servers" ^
    --parameters "Operation=Scan" ^
    --timeout-seconds 600
```

O sistema retorna informações como estas.

```
{
    "Command": {
        "CommandId": "87a448ee-8adc-44e0-b4d1-6b429EXAMPLE",
        "DocumentName": "AWS-RunPatchBaseline",
        "DocumentVersion": "$DEFAULT",
        "Comment": "",
        "ExpiresAfter": 1621974983.128,
        "Parameters": {
            "Operation": [
                "Scan"
            ]
        },
        "InstanceIds": [],
        "Targets": [
            {
                "Key": "tag:PatchGroup",
                "Values": [
                    "Web servers"
                ]
            }
        ],
        "RequestedDateTime": 1621952783.128,
        "Status": "Pending",
        "StatusDetails": "Pending",
        "TimeoutSeconds": 600,

    ---output truncated---

    }
}
```

Instalar patches em nós gerenciados (AWS CLI)

Para instalar patches em nós gerenciados específicos

Execute o seguinte comando:

> **Note**
>
> Os nós gerenciados de destino são reinicializados conforme necessário para concluir a instalação do patch. Para obter mais informações, consulte Documento de comando do SSM para a aplicação de patches: `AWS-RunPatchBaseline`.

Linux & macOS

```
aws ssm send-command \
    --document-name 'AWS-RunPatchBaseline' \
    --targets Key=InstanceIds,Values='i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE' \
    --parameters 'Operation=Install' \
    --timeout-seconds 600
```

Windows Server

```
aws ssm send-command ^
    --document-name "AWS-RunPatchBaseline" ^
    --targets Key=InstanceIds,Values="i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE" ^
    --parameters "Operation=Install" ^
    --timeout-seconds 600
```

O sistema retorna informações como estas.

```
{
    "Command": {
        "CommandId": "5f403234-38c4-439f-a570-93623EXAMPLE",
        "DocumentName": "AWS-RunPatchBaseline",
        "DocumentVersion": "$DEFAULT",
        "Comment": "",
        "ExpiresAfter": 1621975301.791,
        "Parameters": {
            "Operation": [
                "Install"
            ]
        },
        "InstanceIds": [],
        "Targets": [
            {
                "Key": "InstanceIds",
                "Values": [
                    "i-02573cafcfEXAMPLE,
                     i-0471e04240EXAMPLE"
                ]
            }
        ],
        "RequestedDateTime": 1621953101.791,
        "Status": "Pending",
```

```
        "StatusDetails": "Pending",
        "TimeoutSeconds": 600,

    ---output truncated---

    }
}
```

Para instalar patches em nós gerenciados em um grupo de patches específico

Execute o seguinte comando:

Linux & macOS

```
aws ssm send-command \
    --document-name 'AWS-RunPatchBaseline' \
    --targets Key='tag:PatchGroup',Values='Web servers' \
    -parameters 'Operation=Install' \
    --timeout-seconds 600
```

Windows Server

```
aws ssm send-command ^
    --document-name "AWS-RunPatchBaseline" ^
    --targets Key="tag:PatchGroup",Values="Web servers" ^
    --parameters "Operation=Install" ^
    --timeout-seconds 600
```

O sistema retorna informações como estas.

```
{
    "Command": {
        "CommandId": "fa44b086-7d36-4ad5-ac8d-627ecEXAMPLE",
        "DocumentName": "AWS-RunPatchBaseline",
        "DocumentVersion": "$DEFAULT",
        "Comment": "",
        "ExpiresAfter": 1621975407.865,
        "Parameters": {
            "Operation": [
                "Install"
            ]
        },
```

```
        "InstanceIds": [],
        "Targets": [
            {
                "Key": "tag:PatchGroup",
                "Values": [
                    "Web servers"
                ]
            }
        ],
        "RequestedDateTime": 1621953207.865,
        "Status": "Pending",
        "StatusDetails": "Pending",
        "TimeoutSeconds": 600,

    ---output truncated---

    }
}
```

Tutoriais do AWS Systems Manager Patch Manager

Os tutoriais desta seção demonstram como usar o Patch Manager, uma ferramenta do AWS Systems Manager, para vários cenários de aplicação de patches.

Tópicos

- Tutorial: Criar uma lista de referência de patches para instalar o Windows Service Packs usando o console
- Tutorial: Atualizar dependências de aplicações, corrigir um nó gerenciado e executar uma verificação de integridade específica da aplicação usando o console
- Tutorial: Aplicar patches a um ambiente de servidor usando a AWS CLI

Tutorial: Criar uma lista de referência de patches para instalar o Windows Service Packs usando o console

Ao criar uma lista de referência de patches personalizada, é possível especificar que todos, alguns ou apenas um tipo de patch compatível está instalado.

Nas listas de referência de patches do Windows, é possível selecionar `ServicePacks` como a única opção Classificação para limitar atualizações de patch somente para Service Packs. Os Service Packs podem ser instalados automaticamente pelo Patch Manager, uma ferramenta do AWS

Systems Manager, desde que a atualização esteja disponível no Windows Update ou no Windows Server Update Services (WSUS).

É possível configurar uma lista de referência de patches para controlar se os Service Packs de todas as versões do Windows estão instalados ou apenas aqueles de versões específicas, como o Windows 7 ou Windows Server 2016.

Use o procedimento a seguir para criar uma lista de referência de patches personalizada a ser usada exclusivamente para instalar todos os Service Packs em seus nós gerenciados do Windows.

Para criar uma lista de referência de patches para instalar o Windows Service Packs (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Patch Manager.
3. Escolha a guia Listas de referência de patches e, em seguida, escolha Criar lista de referência de patches.
4. Em Nome, insira um nome para a nova lista de referência de patches, como `MyWindowsServicePackPatchBaseline`.
5. (Opcional) Em Description (Descrição), insira uma descrição para essa linha de base de patch.
6. Em Operating system (Sistema operacional), escolha `Windows`.
7. Se você deseja começar a usar essa linha de base de patch como o padrão para o Windows assim que a criar, selecione Set this patch baseline as the default patch baseline for Windows Server instances (Definir essa linha de base de patch como a linha de base de patch padrão para instâncias do Windows Server).

   > **Note**
   >
   > Essa opção está disponível somente se você acessou pela primeira vez Patch Manager antes do lançamento das políticas de patch em 22 de dezembro de 2022.
   > Para obter informações sobre como definir uma linha de base de patch existente como padrão, consulte Definir uma linha de base de patches existente como padrão.

8. Na seção Approval rules for operating systems (Regras de aprovação para sistemas operacionais), use os campos para criar uma ou mais regras de aprovação automática.

   - Produtos: as versões do sistema operacional às quais a regra de aprovação se aplica, como `WindowsServer2012`. É possível escolher uma, mais de uma ou todas as versões compatíveis do Windows. A seleção padrão é `All`.

- Classificação: escolha `ServicePacks`.
- Gravidade: o valor da gravidade dos patches aos quais a regra se aplica. Para garantir que todos os Service Packs estejam incluídos pela regra, escolha `All`.
- Auto-approval (Aprovação automática): o método para selecionar patches para aprovação automática.
  - Approve patches after a specified number of days (Aprovar patches após um número específico de dias): o número de dias que o Patch Manager deve aguardar para aprovar automaticamente um patch após o lançamento ou atualização de um patch. Insira qualquer número inteiro de zero (0) a 360. Para a maioria dos casos, recomendamos que não aguarde mais de 100 dias.
  - Approve patches released up to a specific date (Aprovar patches lançados até uma data específica): a data de lançamento do patch para a qual o Patch Manager aplica automaticamente todos os patches lançados ou com a última atualização nessa data ou antes dela. Por exemplo, se você especificar 7 de julho de 2023, nenhum patch lançado ou com última atualização em ou após 8 de julho de 2023 será instalado automaticamente.
- (Opcional) Relatórios de conformidade: o nível de gravidade que você deseja atribuir aos Service Packs aprovados pela lista de referência, como `High`.

> **Note**
>
> Se você especificar um nível de relatório de conformidade e o estado do patch de qualquer pacote de serviço aprovado for relatado como `Missing`, a gravidade geral da conformidade relatada pela lista de referência de patches será o nível de gravidade especificado.

9. (Opcional) Em Manage tags (Gerenciar tags), aplique um ou mais pares de nome/valor de chave de tag à linha de base de patch.

   Tags são metadados opcionais que você atribui a um recurso. As tags permitem categorizar um recurso de diferentes formas, como por finalidade, proprietário ou ambiente. Para esta lista de referência de patches dedicada à atualização de Service Packs, é possível especificar pares de chave/valor como o seguinte:

   - `Key=OS,Value=Windows`
   - `Key=Classification,Value=ServicePacks`
10. Escolha Create patch baseline.

Tutorial: Atualizar dependências de aplicações, corrigir um nó gerenciado e executar uma verificação de integridade específica da aplicação usando o console

Em muitos casos, um nó gerenciado deve ser reinicializado depois de ter sido corrigido com a atualização de software mais recente. No entanto, a reinicialização de um nó gerenciado em produção sem proteções implementadas pode causar vários problemas, como invocação de alarmes, registro incorreto de dados de métrica e interrupção de sincronizações de dados.

Esse tutorial demonstra como evitar problemas como esses usando o documento do AWS Systems Manager (documento do SSM) `AWS-RunPatchBaselineWithHooks` para obter uma operação de aplicação de patches complexa e de várias etapas que realize o seguinte:

1. Impedir novas conexões com a aplicação
2. Instale as atualizações para o sistema operacional.
3. Atualizar as dependências do pacote da aplicação
4. Reinicie o daemon ().
5. Executar uma verificação de integridade específica da aplicação

Para este exemplo, configuramos nossa infraestrutura desta forma:

- As máquinas virtuais de destino são registradas como nós gerenciados com o Systems Manager.
- `Iptables` é usado como um firewall local.
- A aplicação hospedada em seus nós gerenciados está sendo executada na porta 443.
- A aplicação hospedada em seus nós gerenciados é uma aplicação `nodeJS`.
- A aplicação hospedada em seus nós gerenciados pelo gerenciador de processos pm2.
- A aplicação já possui um endpoint de verificação de integridade especificado.
- O endpoint da verificação de integridade da aplicação não requer autenticação do usuário final. O endpoint permite uma verificação de integridade que atenda aos requisitos da organização para estabelecer a disponibilidade. (Em seus ambientes, pode ser suficiente simplesmente verificar se a aplicação `nodeJS` está em execução e é capaz de receber solicitações. Em outros casos, também é possível verificar se uma conexão com a camada de armazenamento em cache ou com a camada de banco de dados já foi estabelecida.)

Os exemplos deste tutorial são apenas para fins de demonstração e não devem ser implementados inalterados em ambientes de produção. Além disso, tenha em mente que o recurso de ganchos de

ciclo de vida do Patch Manager, uma ferramenta do Systems Manager, com o documento `AWS-RunPatchBaselineWithHooks` pode oferecer suporte a vários outros cenários. Aqui estão alguns exemplos:

- Interrompa um agente de relatório de métricas antes de aplicar patches e reiniciá-lo após a reinicialização do nó gerenciado.
- Desconecte o nó gerenciado de um cluster CRM ou PCS antes de aplicar patches e reconecte após a reinicialização do nó.
- Atualize software de terceiros (por exemplo, aplicações Java, Tomcat, Adobe e outras) em máquinas do Windows Server depois que as atualizações do sistema operacional (SO) forem aplicadas, mas antes da reinicialização do nó gerenciado.

Para atualizar dependências de aplicações, corrigir um nó gerenciado e executar uma verificação de integridade específica da aplicação

1. Crie um documento do SSM para o script de pré-instalação com o conteúdo a seguir e nomeie-o como `NodeJSAppPrePatch`. Substituir *`your_application`* pelo nome da sua aplicação.

   Este script bloqueia imediatamente novas solicitações de entrada e fornece cinco segundos para que as já ativas sejam concluídas antes de iniciar a operação de patch. Para a opção `sleep`, especifique um número de segundos maior do que normalmente leva para que as solicitações recebidas sejam concluídas.

   ```
   # exit on error
   set -e
   # set up rule to block incoming traffic
   iptables -I INPUT -j DROP -p tcp --syn --destination-port 443 || exit 1
   # wait for current connections to end. Set timeout appropriate to your
    application's latency
   sleep 5
   # Stop your application
   pm2 stop your_application
   ```

   Para obter informações sobre como criar um documento do SSM, consulte Criar conteúdo de documento do SSM.

2. Crie outro documento SSM com o conteúdo a seguir para o script após a instalação, para atualizar as dependências da aplicação e dê a ele o nome `NodeJSAppPostPatch`. Substituir */your/application/path* pelo caminho para a sua aplicação.

```
cd /your/application/path
npm update
# you can use npm-check-updates if you want to upgrade major versions
```

3. Crie outro documento SSM com o seguinte conteúdo para o scipt do `onExit` para fazer backup da aplicação e executar uma verificação de integridade. Nomeie este documento do `SSMNodeJSApp0nExitPatch`. Substituir *`your_application`* pelo nome da sua aplicação.

```
# exit on error
set -e
# restart nodeJs application
pm2 start your_application
# sleep while your application starts and to allow for a crash
sleep 10
# check with pm2 to see if your application is running
pm2 pid your_application
# re-enable incoming connections
iptables -D INPUT -j DROP -p tcp --syn --destination-port
# perform health check
/usr/bin/curl -m 10 -vk -A "" http://localhost:443/health-check || exit 1
```

4. Criar uma associação no State Manager, uma ferramenta do AWS Systems Manager para emitir a operação executando as seguintes etapas:

   1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
   2. No painel de navegação, escolha State Manager e selecione Create association.
   3. Para Name (Nome), forneça um nome para ajudar a identificar a finalidade da associação.
   4. Na lista Document (Documento), escolha `AWS-RunPatchBaselineWithHooks`.
   5. Em Action (Ação), selecione Install (Instalar).
   6. (Opcional) Em Snapshot Id (ID do snapshot), forneça um GUID que você gera para ajudar a acelerar a operação e garantir a consistência. O valor GUID pode ser tão simples quanto`00000000-0000-0000-0000-111122223333`.
   7. Para Pre Install Hook Doc Name (Nome do Doc do Hook antes da instalação), insira `NodeJSAppPrePatch`.
   8. Para Post Install Hook Doc Name (Nome do Doc do Hook após instalação), insira `NodeJSAppPostPatch`.

9. Para On ExitHook Doc Name (No nome do documento ExitHook), insira `NodeJSApp0nExitPatch`.

5. Para Targets (Destinos), identifique os nós gerenciados especificando etiquetas, escolhendo os nós manualmente, escolhendo um grupo de recursos ou escolhendo todos os nós gerenciados.
6. Para Specify schedule (Especificar programação), especifique a frequência com que a associação deve ser executada. Por exemplo, a aplicação de patches em nós gerenciados uma vez por semana é uma cadência comum.
7. Na seção Rate control (Controle de taxa), escolha opções para controlar como a associação é executada em vários nós gerenciados. Verifique se apenas uma parte dos nós gerenciados é atualizada de cada vez. Caso contrário, toda ou a maior parte da sua frota poderá ficar offline de uma só vez. Para obter mais informações sobre como usar controles de taxa, consulte Sobre destinos e controles de taxa em associações do State Manager.
8. (Opcional) Em Opções de saída, para salvar a saída de comando em um arquivo, selecione a caixa Habilitar a gravação da saída no S3. Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

> **Note**
>
> As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil da instância atribuído ao nó gerenciado, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço IAM associada ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

9. Escolha Create Association (Criar associação).

Tutorial: Aplicar patches a um ambiente de servidor usando a AWS CLI

O procedimento a seguir descreve como aplicar patch a um ambiente de servidor usando uma linha de base de patch personalizada, grupos de patches e uma janela de manutenção.

Antes de começar

- Instale ou atualize o SSM Agent em seus nós gerenciados. Para aplicar patches a nós gerenciados do Linux, os nós devem estar em execução no SSM Agent versão 2.0.834.0 ou posterior. Para obter mais informações, consulte Atualização do SSM Agent por meio de Run Command.
- Configure as funções e permissões para a Maintenance Windows, uma ferramenta do AWS Systems Manager. Para obter mais informações, consulte Configurar o Maintenance Windows.
- Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

  Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

Para configurar o Patch Manager e aplicar patches aos nós gerenciados (linha de comando)

1. Execute o comando a seguir para criar uma lista de referência de patches para o Windows chamada `Production-Baseline`. Essa lista de referência de patches aprova patches para um ambiente de produção sete dias após o lançamento ou última atualização. Ou seja, marcamos a lista de referência de patches para indicar que é para um ambiente de produção.

   > **Note**
   >
   > O parâmetro `OperatingSystem` e `PatchFilters` varia dependendo do sistema operacional dos nós gerenciados aos quais a lista de referência de patches se aplica. Para obter mais informações, consulte OperatingSystem e PatchFilter.

   Linux & macOS

   ```
   aws ssm create-patch-baseline \
       --name "Production-Baseline" \
       --operating-system "WINDOWS" \
       --tags "Key=Environment,Value=Production" \
       --approval-rules
    "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Critical,Impor
   {Key=CLASSIFICATION,Values=[SecurityUpdates,Updates,ServicePacks,UpdateRollups,CriticalU
    \
       --description "Baseline containing all updates approved for production
    systems"
   ```

Windows Server

```
aws ssm create-patch-baseline ^
    --name "Production-Baseline" ^
    --operating-system "WINDOWS" ^
    --tags "Key=Environment,Value=Production" ^
    --approval-rules
 "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Critical,Impor
{Key=CLASSIFICATION,Values=[SecurityUpdates,Updates,ServicePacks,UpdateRollups,CriticalU
 ^
    --description "Baseline containing all updates approved for production
 systems"
```

O sistema retorna informações como estas.

```
{
   "BaselineId":"pb-0c10e65780EXAMPLE"
}
```

2. Execute os comandos a seguir para registrar a linha de base de patch "Production-Baseline" para dois grupos de patches. Os grupos são chamados de "Database Servers" (Servidores de banco de dados) e "Front-End Servers" (Servidores front-end).

   Linux & macOS

   ```
   aws ssm register-patch-baseline-for-patch-group \
       --baseline-id pb-0c10e65780EXAMPLE \
       --patch-group "Database Servers"
   ```

   Windows Server

   ```
   aws ssm register-patch-baseline-for-patch-group ^
       --baseline-id pb-0c10e65780EXAMPLE ^
       --patch-group "Database Servers"
   ```

   O sistema retorna informações como estas.

   ```
   {
   ```

```
   "PatchGroup":"Database Servers",
   "BaselineId":"pb-0c10e65780EXAMPLE"
}
```

Linux & macOS

```
aws ssm register-patch-baseline-for-patch-group \
    --baseline-id pb-0c10e65780EXAMPLE \
    --patch-group "Front-End Servers"
```

Windows Server

```
aws ssm register-patch-baseline-for-patch-group ^
    --baseline-id pb-0c10e65780EXAMPLE ^
    --patch-group "Front-End Servers"
```

O sistema retorna informações como estas.

```
{
   "PatchGroup":"Front-End Servers",
   "BaselineId":"pb-0c10e65780EXAMPLE"
}
```

3. Execute os comandos a seguir para criar duas janelas de manutenção para os servidores de produção. A primeira janela é executada todas as terças-feiras às 22 horas. A segunda janela é executada todos os sábados às 22 horas. Além disso, a janela de manutenção é marcada para indicar que é para um ambiente de produção.

   Linux & macOS

```
aws ssm create-maintenance-window \
    --name "Production-Tuesdays" \
    --tags "Key=Environment,Value=Production" \
    --schedule "cron(0 0 22 ? * TUE *)" \
    --duration 1 \
    --cutoff 0 \
    --no-allow-unassociated-targets
```

Windows Server

```
aws ssm create-maintenance-window ^
    --name "Production-Tuesdays" ^
    --tags "Key=Environment,Value=Production" ^
    --schedule "cron(0 0 22 ? * TUE *)" ^
    --duration 1 ^
    --cutoff 0 ^
    --no-allow-unassociated-targets
```

O sistema retorna informações como estas.

```
{
   "WindowId":"mw-0c50858d01EXAMPLE"
}
```

Linux & macOS

```
aws ssm create-maintenance-window \
    --name "Production-Saturdays" \
    --tags "Key=Environment,Value=Production" \
    --schedule "cron(0 0 22 ? * SAT *)" \
    --duration 2 \
    --cutoff 0 \
    --no-allow-unassociated-targets
```

Windows Server

```
aws ssm create-maintenance-window ^
    --name "Production-Saturdays" ^
    --tags "Key=Environment,Value=Production" ^
    --schedule "cron(0 0 22 ? * SAT *)" ^
    --duration 2 ^
    --cutoff 0 ^
    --no-allow-unassociated-targets
```

O sistema retorna informações como estas.

```
{
   "WindowId":"mw-9a8b7c6d5eEXAMPLE"
}
```

4. Execute os comandos a seguir para registrar os grupos de patches de servidores `Database` e `Front-End` com suas respectivas janelas de manutenção.

   Linux & macOS

   ```
   aws ssm register-target-with-maintenance-window \
       --window-id mw-0c50858d01EXAMPLE \
       --targets "Key=tag:PatchGroup,Values=Database Servers" \
       --owner-information "Database Servers" \
       --resource-type "INSTANCE"
   ```

   Windows Server

   ```
   aws ssm register-target-with-maintenance-window ^
       --window-id mw-0c50858d01EXAMPLE ^
       --targets "Key=tag:PatchGroup,Values=Database Servers" ^
       --owner-information "Database Servers" ^
       --resource-type "INSTANCE"
   ```

   O sistema retorna informações como estas.

   ```
   {
      "WindowTargetId":"e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
   }
   ```

   Linux & macOS

   ```
   aws ssm register-target-with-maintenance-window \
   --window-id mw-9a8b7c6d5eEXAMPLE \
   --targets "Key=tag:PatchGroup,Values=Front-End Servers" \
   --owner-information "Front-End Servers" \
   --resource-type "INSTANCE"
   ```

Windows Server

```
aws ssm register-target-with-maintenance-window ^
    --window-id mw-9a8b7c6d5eEXAMPLE ^
    --targets "Key=tag:PatchGroup,Values=Front-End Servers" ^
    --owner-information "Front-End Servers" ^
    --resource-type "INSTANCE"
```

O sistema retorna informações como estas.

```
{
   "WindowTargetId":"faa01c41-1d57-496c-ba77-ff9caEXAMPLE"
}
```

5. Execute os comandos a seguir para registrar uma tarefa de patch que instala atualizações ausentes nos servidores `Database` e `Front-End` durante suas respectivas janelas de manutenção.

   Linux & macOS

   ```
   aws ssm register-task-with-maintenance-window \
       --window-id mw-0c50858d01EXAMPLE \
       --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
    \
       --task-arn "AWS-RunPatchBaseline" \
       --service-role-arn "arn:aws:iam::123456789012:role/MW-Role" \
       --task-type "RUN_COMMAND" \
       --max-concurrency 2 \
       --max-errors 1 \
       --priority 1 \
       --task-invocation-parameters "RunCommand={Parameters={Operation=Install}}"
   ```

   Windows Server

   ```
   aws ssm register-task-with-maintenance-window ^
       --window-id mw-0c50858d01EXAMPLE ^
       --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
    ^
       --task-arn "AWS-RunPatchBaseline" ^
       --service-role-arn "arn:aws:iam::123456789012:role/MW-Role" ^
   ```

```
    --task-type "RUN_COMMAND" ^
    --max-concurrency 2 ^
    --max-errors 1 ^
    --priority 1 ^
    --task-invocation-parameters "RunCommand={Parameters={Operation=Install}}"
```

O sistema retorna informações como estas.

```
{
   "WindowTaskId":"4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE"
}
```

Linux & macOS

```
aws ssm register-task-with-maintenance-window \
    --window-id mw-9a8b7c6d5eEXAMPLE \
    --targets "Key=WindowTargetIds,Values=faa01c41-1d57-496c-ba77-ff9caEXAMPLE"
 \
    --task-arn "AWS-RunPatchBaseline" \
    --service-role-arn "arn:aws:iam::123456789012:role/MW-Role" \
    --task-type "RUN_COMMAND" \
    --max-concurrency 2 \
    --max-errors 1 \
    --priority 1 \
    --task-invocation-parameters "RunCommand={Parameters={Operation=Install}}"
```

Windows Server

```
aws ssm register-task-with-maintenance-window ^
    --window-id mw-9a8b7c6d5eEXAMPLE ^
    --targets "Key=WindowTargetIds,Values=faa01c41-1d57-496c-ba77-ff9caEXAMPLE"
 ^
    --task-arn "AWS-RunPatchBaseline" ^
    --service-role-arn "arn:aws:iam::123456789012:role/MW-Role" ^
    --task-type "RUN_COMMAND" ^
    --max-concurrency 2 ^
    --max-errors 1 ^
    --priority 1 ^
    --task-invocation-parameters "RunCommand={Parameters={Operation=Install}}"
```

O sistema retorna informações como estas.

```
{
   "WindowTaskId":"8a5c4629-31b0-4edd-8aea-33698EXAMPLE"
}
```

6. Execute o seguinte comando para obter o resumo de conformidade de patch de alto nível para um grupo de patches. O resumo de conformidade de patches de alto nível inclui o número de nós gerenciados com patches nos respectivos estados do patch.

> **Note**
>
> Durante a primeira janela de manutenção é normal que zeros apareçam como o número de nós gerenciados no resumo, até que a tarefa de aplicação do patch seja executada.

Linux & macOS

```
aws ssm describe-patch-group-state \
    --patch-group "Database Servers"
```

Windows Server

```
aws ssm describe-patch-group-state ^
    --patch-group "Database Servers"
```

O sistema retorna informações como estas.

```
{
   "Instances": number,
   "InstancesWithFailedPatches": number,
   "InstancesWithInstalledOtherPatches": number,
   "InstancesWithInstalledPatches": number,
   "InstancesWithInstalledPendingRebootPatches": number,
   "InstancesWithInstalledRejectedPatches": number,
   "InstancesWithMissingPatches": number,
   "InstancesWithNotApplicablePatches": number,
   "InstancesWithUnreportedNotApplicablePatches": number
```

```
}
```

7. Execute o seguinte comando para obter os estados de resumo de patches por nó gerenciado para um grupo de patches. O resumo por nó gerenciado inclui vários patches nos respectivos estados de patch por instância para um grupo de patches.

   Linux & macOS

   ```
   aws ssm describe-instance-patch-states-for-patch-group \
       --patch-group "Database Servers"
   ```

   Windows Server

   ```
   aws ssm describe-instance-patch-states-for-patch-group ^
       --patch-group "Database Servers"
   ```

   O sistema retorna informações como estas.

   ```
   {
      "InstancePatchStates": [
         {
            "BaselineId": "string",
            "FailedCount": number,
            "InstalledCount": number,
            "InstalledOtherCount": number,
            "InstalledPendingRebootCount": number,
            "InstalledRejectedCount": number,
            "InstallOverrideList": "string",
            "InstanceId": "string",
            "LastNoRebootInstallOperationTime": number,
            "MissingCount": number,
            "NotApplicableCount": number,
            "Operation": "string",
            "OperationEndTime": number,
            "OperationStartTime": number,
            "OwnerInformation": "string",
            "PatchGroup": "string",
            "RebootOption": "string",
            "SnapshotId": "string",
            "UnreportedNotApplicableCount": number
         }
   ```

```
    ]
}
```

Para obter exemplos de outros comandos da AWS CLI que você pode usar nas tarefas de configuração do Patch Manager, consulte Trabalhar com recursos do Patch Manager usando a AWS CLI.

Solução de problemas do Patch Manager

Use as informações a seguir para obter ajuda para solucionar problemas com o Patch Manager, uma ferramenta do AWS Systems Manager.

Tópicos

- Problema: erro “Invoke-PatchBaselineOperation: Access Denied” ou erro “Unable to download file from S3” para baseline_overrides.json
- Problema: a aplicação de patches falha sem uma causa aparente ou mensagem de erro
- Problema: resultados inesperados de conformidade de patches
- Erros ao executar AWS-RunPatchBaseline no Linux
- Erros ao executar AWS-RunPatchBaseline no Windows Server
- Como usar runbooks do AWS Support Automation
- Entrar em contato com o AWS Support

Problema: erro “Invoke-PatchBaselineOperation: Access Denied” ou erro “Unable to download file from S3” para **`baseline_overrides.json`**

Problema: quando as operações de aplicação de patches especificadas pela política de patch são executadas, você recebe um erro semelhante ao exemplo a seguir.

Example error on Windows Server

```
----------ERROR-------
Invoke-PatchBaselineOperation : Access Denied
At C:\ProgramData\Amazon\SSM\InstanceData\i-02573cafcfEXAMPLE\document\orchestr
ation\792dd5bd-2ad3-4f1e-931d-abEXAMPLE\PatchWindows\_script.ps1:219 char:13
+ $response = Invoke-PatchBaselineOperation -Operation Install -Snapsho ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
+ CategoryInfo : OperationStopped: (Amazon.Patch.Ba...UpdateOpera
```

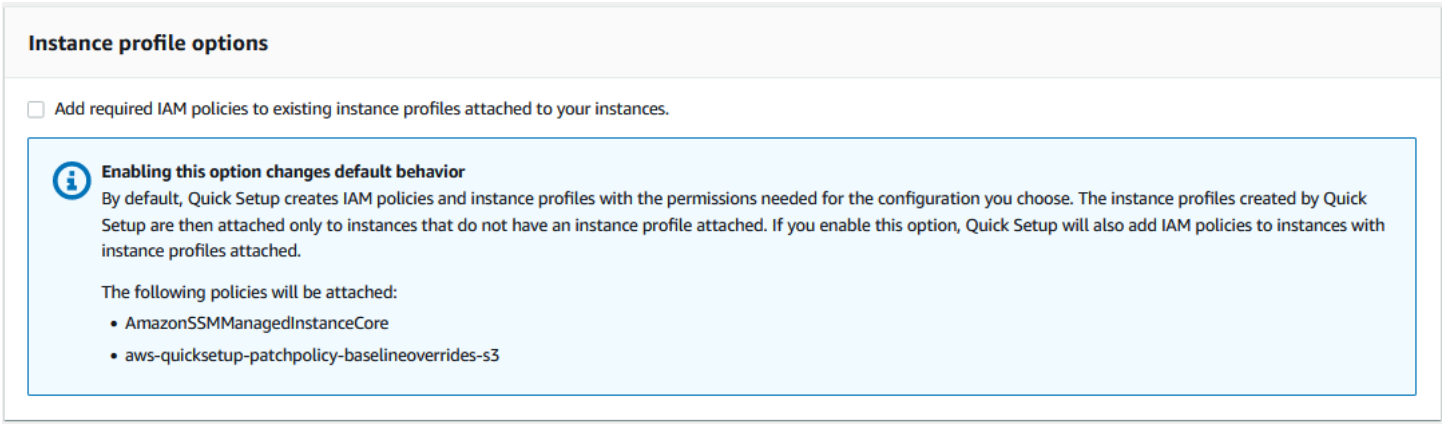
```
tion:InstallWindowsUpdateOperation) [Invoke-PatchBaselineOperation], Amazo
nS3Exception
+ FullyQualifiedErrorId : PatchBaselineOperations,Amazon.Patch.Baseline.Op
erations.PowerShellCmdlets.InvokePatchBaselineOperation
failed to run commands: exit status 0xffffffff
```

Example error on Linux

```
[INFO]: Downloading Baseline Override from s3://aws-quicksetup-
patchpolicy-123456789012-abcde/baseline_overrides.json
[ERROR]: Unable to download file from S3: s3://aws-quicksetup-
patchpolicy-123456789012-abcde/baseline_overrides.json.
[ERROR]: Error loading entrance module.
```

Causa: você criou uma política de patch na Quick Setup, e alguns de seus nós gerenciados já tinham um perfil de instância anexado (para instâncias do EC2) ou um perfil de serviço anexado (para máquinas que não são do EC2).

No entanto, conforme mostrado na imagem a seguir, você não marcou a caixa de seleção Adicionar políticas do IAM necessárias aos perfis de instância existentes anexados às suas instâncias.



Quando você cria uma política de patch, também é criado um bucket do Amazon S3 para armazenar o arquivo de configuração `baseline_overrides.json` da política. Se você não marcar a caixa de seleção Adicionar políticas do IAM necessárias aos perfis de instância existentes anexados às suas instâncias ao criar a política, as políticas do IAM e as etiquetas de recursos necessárias para acessar `baseline_overrides.json` no bucket do S3 não serão adicionadas automaticamente aos perfis de instâncias e perfis de serviço já existentes no IAM.

Solução 1: exclua a configuração da política de patch atual e crie uma substituta, certificando-se de marcar a caixa de seleção Adicionar políticas do IAM necessárias aos perfis de instância existentes

anexados às suas instâncias. Essa seleção aplica as políticas do IAM criadas por essa configuração da Quick Setup aos nós que já têm um perfil de instância ou um perfil de serviço anexado. (Por padrão, a Quick Setup adiciona as políticas necessárias às instâncias e aos nós que ainda não tenham perfis de instância ou perfis de serviço.) Para obter mais informações, consulte Automate organization-wide patching using a Quick Setup patch policy.

Solução 2: adicione manualmente as permissões e etiquetas necessárias a cada perfil de instância do IAM e perfil de serviço do IAM que você usa com a Quick Setup. Para instruções, consulte Permissões para o bucket do S3 da política de patch.

Problema: a aplicação de patches falha sem uma causa aparente ou mensagem de erro

Problema: uma operação de aplicação de patches falha sem retornar uma mensagem de erro.

Possível causa: se houver mais de uma invocação de `AWS-RunPatchBaseline` ao mesmo tempo, elas podem entrar em conflito umas com as outras, fazendo com que as tarefas de aplicação de patches falhem. Isso pode não estar indicado nos logs de aplicação de patches.

Para verificar se as operações simultâneas de aplicação de patches foram interrompidas, analise o histórico de comandos no Run Command, uma ferramenta do AWS Systems Manager. Para um nó gerenciado com uma falha de aplicação de patches, verifique se várias operações tentaram aplicar patches à máquina com uma diferença de até dois minutos uma da outra. Às vezes, esse cenário pode causar falha.

Você também pode usar a AWS Command Line Interface (AWS CLI) para verificar tentativas simultâneas de aplicação de pacthes usando o comando a seguir. Substitua o valor de *node-id* pelo ID de seu nó gerenciado.

```
aws ssm list-commands \
    --filter "key=DocumentName,value=AWS-RunPatchBaseline" \
    --query 'Commands[*].
{CommandId:CommandId,RequestedDateTime:RequestedDateTime,Status:Status}' \
    --instance-id node-id \
    --output table
```

Solução: se você determinar que a aplicação de patches falhou devido a operações de aplicação de patches concorrentes no mesmo nó gerenciado, ajuste as configurações de aplicação de patches para evitar que isso ocorra novamente. Por exemplo, se duas janelas de manutenção especificarem horários de aplicação de patches coincidentes, remova ou revise um deles. Se uma

janela de manutenção especificar uma operação de aplicação de patches, mas uma política de patch especificar outra para o mesmo horário, considere remover a tarefa da janela de manutenção.

Se você determinar que operações de aplicação de patches conflitantes não foram a causa da falha nesse cenário, recomendamos entrar em contato com o AWS Support.

Problema: resultados inesperados de conformidade de patches

Problema: ao analisar os detalhes de conformidade dos patches gerados após uma operação `Scan`, os resultados incluem informações que não refletem as regras definidas na lista de referência de patches. Por exemplo, uma exceção adicionada à lista Rejected patches (Patches rejeitados) em uma lista de referência de patches é listada como `Missing`. Ou os patches classificados como `Important` estão listados como ausentes, mesmo que sua lista de referência de patches especifique somente patches `Critical`.

Causa: Patch Manager atualmente oferece suporte a vários métodos de execução de operações `Scan`:

- Uma política de patch configurada no Quick Setup
- Uma opção do Host Management configurada no Quick Setup
- Uma janela de manutenção para executar um patch `Scan` ou tarefa `Install`
- Uma operação Patch now (Aplicar patch agora) sob demanda

Quando uma operação `Scan` é executada, ela substitui os detalhes de conformidade da verificação mais recente. Se você tiver mais de um método configurado para executar uma operação `Scan`, e eles usarem listas de referência de patches diferentes com regras diferentes, eles resultarão em resultados de conformidade de patches diferentes.

Solução: para evitar resultados inesperados de conformidade de patches, recomendamos usar somente um método por vez para executar a operação `Scan` do Patch Manager. Para obter mais informações, consulte Prevenção de substituições não intencionais de dados de conformidade de patches.

Erros ao executar **`AWS-RunPatchBaseline`** no Linux

Tópicos

- Problema: erro 'Nenhum arquivo ou diretório'
- Problema: erro 'outro processo adquiriu bloqueio yum'

- Problema: erro 'Permissão negada/falhou ao executar comandos'
- Problema: Erro 'Não é possível baixar a carga de pagamento'
- Problema: erro 'gerenciador de pacotes não suportado e combinação de versão python'
- Problema:Patch Managernão está aplicando regras especificadas para excluir determinados pacotes
- Problema: falha na aplicação de patches e o Patch Manager relata que a extensão Indicação de Nome do Servidor para TLS não está disponível
- Problema: o Patch Manager relata "Não há mais espelhos para tentar"
- Problema: a aplicação de patches falhou com “Error code returned from curl is 23”
- Problema: falha na aplicação de patches com a mensagem “Error unpacking rpm package...”
- Problema: a aplicação de patches falha com a mensagem “Encounter service side error when uploading the inventory”
- Problema: falha na aplicação de patches com a mensagem “Errors were encountered while downloading packages”
- Problema: falha na aplicação de patches com a mensagem “Não foi possível verificar as seguintes assinaturas, pois a chave pública não está disponível”
- Problema: falha na aplicação de patches com a mensagem “NoMoreMirrorsRepoError”
- Problema: falha na aplicação de patches com a mensagem “Unable to download payload”
- Problema: falha na aplicação de patches com a mensagem “install errors: dpkg: error: dpkg frontend is locked by another process”
- Problema: a aplicação de patches no Ubuntu Server falha com um erro “dpkg was interrupted”
- Problema: o utilitário gerenciador de pacotes não consegue resolver uma dependência de pacote
- Problema: falhas na dependência do bloqueio de pacotes zypper em nós gerenciados pelo SLES

Problema: erro 'Nenhum arquivo ou diretório'

Problema: Quando você executa`AWS-RunPatchBaseline`, o patch falha com um dos seguintes erros.

```
IOError: [Errno 2] No such file or directory: 'patch-baseline-operations-X.XX.tar.gz'
```

```
Unable to extract tar file: /var/log/amazon/ssm/patch-baseline-operations/patch-
baseline-operations-1.75.tar.gz.failed to run commands: exit status 155
```

```
Unable to load and extract the content of payload, abort.failed to run commands: exit
 status 152
```

Causa 1: dois comandos para executar `AWS-RunPatchBaseline` estavam em execução ao mesmo tempo no mesmo nó. Isso cria uma condição de corrida que resulta no`file patch-baseline-operations*`não sendo criado ou acessado corretamente.

Causa 2: espaço de armazenamento insuficiente permanece no diretório `/var`.

Solução 1: certifique-se de que nenhuma janela de manutenção tenha duas ou mais tarefas do Run Command que executam o `AWS-RunPatchBaseline` com o mesmo nível de prioridade e que executem nos mesmos IDs de destino. Se for esse o caso, reordene a prioridade. O Run Command é uma ferramenta do AWS Systems Manager.

Solução 2: certifique-se de que apenas uma janela de manutenção de cada vez esteja executando tarefas do Run Command que usam `AWS-RunPatchBaseline` nos mesmos alvos e na mesma programação. Nesse caso, altere o horário.

Solução 3: certifique-se de que apenas uma associação do State Manager esteja em execução no `AWS-RunPatchBaseline` na mesma agenda e voltada para os mesmos nós gerenciados. O State Manager é uma ferramenta do AWS Systems Manager.

Solução 4: libere espaço de armazenamento suficiente sob o diretório `/var` para os pacotes de atualização.

Problema: erro 'outro processo adquiriu bloqueio yum'

Problema: Quando você executa`AWS-RunPatchBaseline`A aplicação de patch falha com o erro a seguir.

```
12/20/2019 21:41:48 root [INFO]: another process has acquired yum lock, waiting 2 s and
 retry.
```

Causa: o documento `AWS-RunPatchBaseline` começou a ser executado em um nó gerenciado onde já está sendo executado em outra operação e adquiriu o processo `yum` do gerenciador de pacotes.

Solução: certifique-se de que nenhuma associação, tarefa da janela de manutenção ou outras configuração do State Manager executada no `AWS-RunPatchBaseline` em uma programação esteja voltada para o mesmo nó gerenciado de destino aproximadamente ao mesmo tempo.

Problema: erro 'Permissão negada/falhou ao executar comandos'

Problema: Quando você executa`AWS-RunPatchBaseline`A aplicação de patch falha com o erro a seguir.

```
sh:
/var/lib/amazon/ssm/instanceid/document/orchestration/commandid/PatchLinux/_script.sh:
 Permission denied
failed to run commands: exit status 126
```

Causa: o `/var/lib/amazon/` pode ser montado com permissões do `noexec`. Este é um problema porque o SSM Agent faz o download de scripts de carga para `/var/lib/amazon/ssm` e os executa a partir desse local.

Solução: confirme se você configurou partições exclusivas para `/var/log/amazon` e `/var/lib/amazon`, e se elas estão montadas com permissões `exec`.

Problema: Erro 'Não é possível baixar a carga de pagamento'

Problema: Quando você executa`AWS-RunPatchBaseline`A aplicação de patch falha com o erro a seguir.

```
Unable to download payload: https://s3.amzn-s3-demo-bucket.region.amazonaws.com/
aws-ssm-region/patchbaselineoperations/linux/payloads/patch-baseline-operations-
X.XX.tar.gz.failed to run commands: exit status 156
```

Causa: o nó gerenciado não tem as permissões necessárias para acessar o bucket especificado do Amazon Simple Storage Service (Amazon S3).

Solução: atualize sua configuração de rede para que os endpoints do S3 estejam acessíveis. Para obter mais detalhes, consulte informações sobre o acesso necessário aos buckets do S3 para o Patch Manager em Comunicações do SSM Agent com os buckets do S3 gerenciados pela AWS.

Problema: erro 'gerenciador de pacotes não suportado e combinação de versão python'

Problema: Quando você executa`AWS-RunPatchBaseline`A aplicação de patch falha com o erro a seguir.

```
An unsupported package manager and python version combination was found. Apt requires
 Python3 to be installed.
```

```
failed to run commands: exit status 1
```

Causa: uma versão com suporte do python3 não está instalada na instância do Debian Server, Raspberry Pi OS ou Ubuntu Server.

Solução: instale uma versão com suporte do python3 (3.0-3.10) no servidor, que é necessária para nós gerenciados do Debian Server, Raspberry Pi OS e Ubuntu Server.

Problema:Patch Managernão está aplicando regras especificadas para excluir determinados pacotes

Problema: Você tentou excluir determinados pacotes especificando-os na seção`/etc/yum.conf`, no formato`exclude=`*`package-name`*, mas eles não são excluídos durante oPatch Manager `Install`operação.

Causa: o Patch Manager não incorpora exclusões especificadas no arquivo `/etc/yum.conf`.

Solução: Para excluir pacotes específicos, crie uma linha de base de patch personalizada e crie uma regra para excluir os pacotes que você não deseja instalar.

Problema: falha na aplicação de patches e o Patch Manager relata que a extensão Indicação de Nome do Servidor para TLS não está disponível

Problema: A operação de patch emite a seguinte mensagem.

```
/var/log/amazon/ssm/patch-baseline-operations/urllib3/util/ssl_.py:369:
SNIMissingWarning: An HTTPS request has been made, but the SNI (Server Name Indication)
 extension
to TLS is not available on this platform. This might cause the server to present an
 incorrect TLS
certificate, which can cause validation failures. You can upgrade to a newer version of
 Python
to solve this.
For more information, see https://urllib3.readthedocs.io/en/latest/advanced-
usage.html#ssl-warnings
```

Causa: esta mensagem não indica um erro. Em vez disso, é um aviso de que a versão mais antiga do Python distribuída com o sistema operacional não suporta a Indicação de Nome do Servidor TLS. O script de carga de patch do Systems Manager emite este aviso ao se conectar a APIs da AWS que oferecem suporte a SNI.

Solução: para solucionar problemas de falhas de patch quando essa mensagem é relatada, revise o conteúdo do`stdout`e`stderr`arquivos. Se você não configurou a lista de referência de patches

para armazenar esses arquivos em um bucket do S3 ou no Amazon CloudWatch Logs, é possível localizar os arquivos no local a seguir em seu nó gerenciado do Linux.

`/var/lib/amazon/ssm/`*`instance-id`*`/document/orchestration/`*`Run-Command-execution-id`*`/awsrunShellScript/PatchLinux`

Problema: o Patch Manager relata "Não há mais espelhos para tentar"

Problema: A operação de patch emite a seguinte mensagem.

```
[Errno 256] No more mirrors to try.
```

Causa: os repositórios configurados em seu nó gerenciado não estão funcionando corretamente. As possíveis causas incluem:

- OyumA cache está corrompida.
- Um URL de repositório não pode ser alcançado devido a problemas relacionados à rede.

Solução: o Patch Manager usa o gerenciador de pacotes padrão do nó gerenciado para executar a operação de aplicação de patch. Verifique se os repositórios estão configurados e funcionando corretamente.

Problema: a aplicação de patches falhou com “Error code returned from curl is 23”

Problema: uma operação de aplicação de patches que usa `AWS-RunPatchBaseline` falha com um erro semelhante a este:

```
05/01/2025 17:04:30 root [ERROR]: Error code returned from curl is 23
```

Causa: a ferramenta curl usada em seus sistemas não tem as permissões necessárias para gravar no sistema de arquivos. Isso pode ocorrer quando a ferramenta curl padrão do gerenciador de pacotes foi substituída por outra versão, como uma instalada com snap.

Solução: se a versão curl fornecida pelo gerenciador de pacotes foi desinstalada quando outra versão foi instalada, reinstale-a.

Se você precisar manter várias versões do curl instaladas, certifique-se de que a versão associada ao gerenciador de pacotes esteja no primeiro diretório listado na variável `PATH`. Você pode verificar isso executando o comando `echo $PATH` para ver a ordem atual dos diretórios de seu sistema que estão sendo verificados quanto a arquivos executáveis.

Problema: falha na aplicação de patches com a mensagem “Error unpacking rpm package...”

Problema: uma operação de aplicação de patches falha com um erro semelhante a este:

```
Error : Error unpacking rpm package python-urllib3-1.25.9-1.amzn2.0.2.noarch
python-urllib3-1.25.9-1.amzn2.0.1.noarch was supposed to be removed but is not!
failed to run commands: exit status 1
```

Causa 1: quando um pacote específico está presente em vários instaladores de pacotes, como `pip` e `yum` ou `dnf`, podem ocorrer conflitos ao usar o gerenciador de pacotes padrão.

Um exemplo comum ocorre com o pacote `urllib3`, encontrado em `pip`, `yum` e `dnf`.

Causa 2: o pacote `python-urllib3` está corrompido. Isso poderá acontecer se os arquivos do pacote tiverem sido instalados ou atualizados pelo `pip` após o pacote `rpm` ter sido instalado anteriormente por `yum` ou `dnf`.

Solução: remova o pacote `python-urllib3` do pip executando o comando `sudo pip uninstall urllib3`, mantendo o pacote somente no gerenciador de pacotes padrão (`yum` ou `dnf`).

Problema: a aplicação de patches falha com a mensagem “Encounter service side error when uploading the inventory”

Problema: ao executar o documento `AWS-RunPatchBaseline`, você recebe a seguinte mensagem de erro:

```
Encounter service side error when uploading the inventory
```

Causa: dois comandos para executar `AWS-RunPatchBaseline` estavam em execução ao mesmo tempo no mesmo nó gerenciado. Isso cria uma condição de corrida ao inicializar o cliente boto3 durante as operações de aplicação de patches.

Solução: certifique-se de que nenhuma associação, tarefa da janela de manutenção ou outras configuração do State Manager executada no `AWS-RunPatchBaseline` em uma programação esteja voltada para o mesmo nó gerenciado de destino aproximadamente ao mesmo tempo.

Problema: falha na aplicação de patches com a mensagem “Errors were encountered while downloading packages”

Problema: durante a aplicação de patches, você recebe um erro semelhante a este:

```
YumDownloadError: [u'Errors were encountered while downloading
packages.', u'libxml2-2.9.1-6.el7_9.6.x86_64: [Errno 5] [Errno 12]
Cannot allocate memory', u'libxslt-1.1.28-6.el7.x86_64: [Errno 5]
[Errno 12] Cannot allocate memory', u'libcroco-0.6.12-6.el7_9.x86_64:
[Errno 5] [Errno 12] Cannot allocate memory', u'openldap-2.4.44-25.el7_9.x86_64:
[Errno 5] [Errno 12] Cannot allocate memory',
```

Causa: esse erro pode ocorrer quando a memória disponível em um nó gerenciado é insuficiente.

Solução: configure a memória swap ou atualize a instância para um tipo diferente para aumentar o suporte à memória. Em seguida, inicie uma nova operação de aplicação de patches.

Problema: falha na aplicação de patches com a mensagem “Não foi possível verificar as seguintes assinaturas, pois a chave pública não está disponível”

Problema: aplicação de patches falha no Ubuntu Server com um erro semelhante a este:

```
02/17/2022 21:08:43 root [ERROR]: W:GPG error:
http://repo.mysql.com/apt/ubuntu  bionic InRelease: The following
signatures couldn't be verified because the public key is not available:
NO_PUBKEY 467B942D3A79BD29, E:The repository ' http://repo.mysql.com/apt/ubuntu bionic
```

Causa: a chave GNU Privacy Guard (GPG) expirou ou está ausente.

Solução: atualize a chave GPG ou adicione a chave novamente.

Por exemplo, usando o erro mostrado anteriormente, vemos que a chave 467B942D3A79BD29 está ausente e deve ser adicionada. Para fazer isso, execute um dos comandos a seguir:

```
sudo apt-key adv --keyserver hkps://keyserver.ubuntu.com --recv-keys 467B942D3A79BD29
```

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys 467B942D3A79BD29
```

Ou, para atualizar todas as chaves, execute este comando:

```
sudo apt-key adv --keyserver hkps://keyserver.ubuntu.com --refresh-keys
```

Se o erro persistir depois disso, é recomendável relatar o problema à organização que retém o repositório. Até que uma correção esteja disponível, você pode editar o arquivo `/etc/apt/sources.list` de modo a omitir o repositório durante o processo de aplicação de patches.

Para fazer isso, abra o arquivo `sources.list` para edição, localize a linha do repositório e insira um caractere # no início da linha para comentá-la. Depois, salve e feche o arquivo.

Problema: falha na aplicação de patches com a mensagem “NoMoreMirrorsRepoError”

Problema: você recebe um erro semelhante a este:

```
NoMoreMirrorsRepoError: failure: repodata/repomd.xml from pgdg94: [Errno 256] No more
 mirrors to try.
```

Causa: há um erro no repositório de origem.

Solução: é recomendável relatar o problema à organização que retém o repositório. Até que o erro seja corrigido, é possível desabilitar o repositório no nível do sistema operacional. Para isso, execute o seguinte comando, substituindo o valor de *`repo-name`* pelo nome do repositório:

```
yum-config-manager --disable repo-name
```

Veja um exemplo a seguir.

```
yum-config-manager --disable pgdg94
```

Depois de executar esse comando, execute outra operação de aplicação de patches.

Problema: falha na aplicação de patches com a mensagem “Unable to download payload”

Problema: você recebe um erro semelhante a este:

```
Unable to download payload:
https://s3.dualstack.eu-west-1.amazonaws.com/aws-ssm-eu-west-1/patchbaselineoperations/
linux/payloads/patch-baseline-operations-1.83.tar.gz.
failed  to run commands: exit status 156
```

Causa: a configuração do nó gerenciado contém erros ou está incompleta.

Solução: verifique se o nó gerenciado está configurado desta forma:

- Regra de saída TCP 443 no grupo de segurança.
- Regra de saída TCP 443 em NACL.
- Regra de entrada TCP 1024-65535 em NACL.

- NAT/IGW na tabela de rotas para fornecer conectividade a um endpoint do S3. Se a instância não tiver acesso à Internet, forneça conectividade com o endpoint do S3. Para isso, adicione um endpoint de gateway do S3 à VPC e integre-o à tabela de rotas do nó gerenciado.

Problema: falha na aplicação de patches com a mensagem “install errors: dpkg: error: dpkg frontend is locked by another process”

Problema: aplicação de patches falha com um erro semelhante a este:

```
install errors: dpkg: error: dpkg frontend is locked by another process
failed to run commands: exit status 2
Failed to install package; install status Failed
```

Causa: o gerenciador de pacotes já está executando outro processo em um nó gerenciado no nível do sistema operacional. Se esse outro processo levar muito tempo para ser concluído, a operação de aplicação de patches do Patch Manager poderá atingir o tempo limite e falhar.

Solução: após a conclusão do outro processo que está usando o gerenciador de pacotes, execute uma nova operação de aplicação de patches.

Problema: a aplicação de patches no Ubuntu Server falha com um erro “dpkg was interrupted”

Problema: no Ubuntu Server, a aplicação de patches falha com um erro semelhante a este:

```
E: dpkg was interrupted, you must manually run
'dpkg --configure -a' to correct the problem.
```

Causa: um ou mais pacotes foram configurados incorretamente.

SoluçãoSiga estas etapas:

1. Execute os seguintes comandos, um por vez, para verificar quais pacotes são afetados e quais são os problemas com cada pacote:

   ```
   sudo apt-get check
   ```

   ```
   sudo dpkg -C
   ```

   ```
   dpkg-query -W -f='${db:Status-Abbrev} ${binary:Package}\n' | grep -E ^.[^nci]
   ```

2. Execute este comando para corrigir os pacotes com problemas:

```
sudo dpkg --configure -a
```

3. Se o comando anterior não resolver totalmente o problema, execute este comando:

```
sudo apt --fix-broken install
```

Problema: o utilitário gerenciador de pacotes não consegue resolver uma dependência de pacote

Problema: o gerenciador de pacotes nativo no nó gerenciado não consegue resolver uma dependência de pacote, e a aplicação de patches falha. O exemplo de mensagem de erro a seguir indica esse tipo de falha em um sistema operacional que usa o `yum` como gerenciador de pacotes.

```
09/22/2020 08:56:09 root [ERROR]: yum update failed with result code: 1,
message: [u'rpm-python-4.11.3-25.amzn2.0.3.x86_64 requires rpm = 4.11.3-25.amzn2.0.3',
u'awscli-1.18.107-1.amzn2.0.1.noarch requires python2-botocore = 1.17.31']
```

Causa: em sistemas operacionais Linux, o Patch Manager usa o gerenciador de pacotes nativo da máquina para executar operações de aplicação de patches, como `yum`, `dnf`, `apt` e `zypper`. As aplicações detectam, instalam, atualizam ou removem automaticamente pacotes dependentes, conforme necessário. Porém, algumas condições podem fazer com que o gerenciador de pacotes não consiga concluir uma operação de dependência, como:

- Múltiplos repositórios conflitantes estão configurados no sistema operacional.
- O URL do repositório remoto está inacessível por causa de problemas relacionados à rede.
- Foi encontrado um pacote para a arquitetura errada no repositório.

Solução: a aplicação de patches pode falhar por causa de um problema de dependência de diversos motivos. Portanto, é recomendável entrar em contato com o AWS Support para ajudar na solução de problemas.

Problema: falhas na dependência do bloqueio de pacotes zypper em nós gerenciados pelo SLES

Problema: quando você executa `AWS-RunPatchBaseline` com a operação `Install` em instâncias do SUSE Linux Enterprise Server, a aplicação de patches falha com erros de verificação de dependência relacionados a bloqueios de pacotes. Você poderá receber mensagens de erro semelhantes à seguinte mensagem:

```
Problem: mock-pkg-has-dependencies-0.2.0-21.adistro.noarch requires mock-pkg-standalone
 = 0.2.0, but this requirement cannot be provided
  uninstallable providers: mock-pkg-standalone-0.2.0-21.adistro.noarch[local-repo]
 Solution 1: remove lock to allow installation of mock-pkg-
standalone-0.2.0-21.adistro.noarch[local-repo]
 Solution 2: do not install mock-pkg-has-dependencies-0.2.0-21.adistro.noarch
 Solution 3: break mock-pkg-has-dependencies-0.2.0-21.adistro.noarch by ignoring some
 of its dependencies

Choose from above solutions by number or cancel [1/2/3/c] (c): c
```

Neste exemplo, o pacote `mock-pkg-standalone` está bloqueado, o que você pode verificar executando `sudo zypper locks` e procurando esse nome de pacote na saída.

Ou você pode ver as entradas de logs indicando falhas na verificação de dependência:

```
Encountered a known exception in the CLI Invoker:
 CLIInvokerError(error_message='Dependency check failure during commit process',
 error_code='4')
```

> **Note**
>
> Esse problema ocorre somente durante as operações `Install`. As operações `Scan` não aplicam bloqueios de pacotes e não são afetadas pelos bloqueios existentes.

Causa: este erro ocorre quando os bloqueios de pacotes zypper impedem a instalação ou atualização de pacotes por causa de conflitos de dependência. Os bloqueios de pacotes podem estar presentes por diversos motivos:

- Bloqueios aplicados pelo cliente: você ou o administrador do sistema bloqueou pacotes manualmente usando comandos do zypper, como `zypper addlock`.
- Patches rejeitados pelo Gerenciador de Patches: o Patch Manager aplica automaticamente bloqueios de pacotes quando você especifica pacotes na lista Patches rejeitados da sua lista de referência de patches para impedir a instalação.
- Bloqueios residuais de operações interrompidas: em casos raros, se uma operação de patch foi interrompida (como por uma reinicialização do sistema) antes de o Patch Manager poder limpar os bloqueios temporários, os bloqueios residuais de pacotes poderão permanecer em seu nó gerenciado.

Solução: para resolver problemas de bloqueio de pacotes zypper, siga estas etapas com base na causa:

Etapa 1: identificar pacotes bloqueados

Conecte-se ao seu nó gerenciado pelo SLES e execute o seguinte comando para listar todos os pacotes atualmente bloqueados:

```
sudo zypper locks
```

Etapa 2: determinar a origem dos bloqueios

- Se os pacotes bloqueados forem os que você bloqueou intencionalmente para a estabilidade do sistema, analise se eles precisam permanecer bloqueados ou se podem ser desbloqueados temporariamente para a aplicação de patches.
- Se os pacotes bloqueados corresponderem às entradas da lista Patches rejeitados da sua lista de referência de patches, provavelmente são bloqueios residuais de uma operação de patch interrompida. Durante as operações normais, o Patch Manager aplica esses bloqueios temporariamente e os remove automaticamente quando a operação é concluída. Você pode remover os pacotes da lista de rejeitados ou modificar suas regras da lista de referência de patches.
- Se você não reconhece os pacotes bloqueados e eles não foram bloqueados intencionalmente, eles podem ser bloqueios residuais de uma operação anterior de patch interrompida.

Etapa 3: remover os bloqueios conforme apropriado

Para remover bloqueios de pacotes específicos, use o seguinte comando:

```
sudo zypper removelock package-name
```

Para remover todos os bloqueios de pacotes (use com cautela), execute:

```
sudo zypper cleanlocks
```

Etapa 4: atualizar a lista de referência de patches (se aplicável)

Se os bloqueios foram causados por patches rejeitados na lista de referência de patches:

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Patch Manager.
3. Escolha a guia Listas de referência de patches, e depois escolha sua lista de referência de patches personalizada.
4. Escolha Ações, Modificar lista de referência de patches.
5. Na seção Patches rejeitados, revise os pacotes listados e remova os que devem ter permissão para instalação.
6. Escolha Salvar alterações.

Etapa 5: tentar novamente a operação de patch

Depois de remover os bloqueios problemáticos e atualizar a lista de referência de patches, se necessário, execute o documento `AWS-RunPatchBaseline` novamente.

> **Note**
>
> Quando o Patch Manager aplica bloqueios para patches rejeitados durante as operações `Install`, ele foi projetado para limpar esses bloqueios automaticamente após a conclusão da operação de patch. Se você vir esses bloqueios durante a execução de `sudo zypper locks`, isso indica que uma operação de patch anterior foi interrompida antes que a limpeza pudesse ocorrer. No entanto, se uma operação de patch for interrompida, a limpeza manual poderá ser necessária, conforme descrito neste procedimento.

Prevenção: para evitar futuros conflitos de bloqueios do zypper:

- Analise cuidadosamente a lista de patches rejeitados da sua lista de referência de patches para garantir que ela inclua somente pacotes que você realmente deseja excluir.
- Evite bloquear manualmente pacotes que possam ser necessários como dependências para as atualizações de segurança.
- Se você precisar bloquear pacotes manualmente, documente os motivos e revise os bloqueios periodicamente.
- Certifique-se de que as operações de patch sejam concluídas com êxito e não sejam interrompidas por reinicializações do sistema ou outros fatores.
- Monitore as operações de patch até a conclusão e evite interrompê-las com reinicializações do sistema ou outras ações que possam impedir a limpeza adequada dos bloqueios temporários.

Erros ao executar **`AWS-RunPatchBaseline`** no Windows Server

Tópicos

- Problema: família de produtos/pares de produtos incompatíveis
- Problema:AWS-RunPatchBaselineretorna umHRESULT(Windows Server)
- Problema: o nó gerenciado não tem acesso ao Catálogo do Windows Update ou ao WSUS
- Problema: o módulo PatchBaselineOperations PowerShell não pode ser baixado
- Problema: patches ausentes

Problema: família de produtos/pares de produtos incompatíveis

Problema: ao criar uma lista de referência de patches no console do Systems Manager, você especifica uma família de produtos e um produto. Por exemplo, você pode escolher:

- Product family (Família de produtos): `Office`

  Product (Produto): `Office 2016`

Cause: se você tentar criar uma linha de base de patch com um par ou uma família de produtos sem correspondência, uma mensagem de erro será exibida. Isso pode acontecer pelos seguintes motivos:

- Você selecionou um par e uma família de produtos, mas depois removeu a seleção da família de produtos.
- Você escolheu um produto na sublista Obsolete or mismatched options (Opções obsoletas ou incompatíveis) em vez de sublista Available and matching options (Opções disponíveis e correspondentes).

  Os itens na sublista Obsolete or mismatched options (Opções obsoletas ou incompatíveis) do produto podem ter sido inseridos incorretamente por um SDK ou comando `create-patch-baseline` da AWS Command Line Interface (AWS CLI). Isso pode significar um erro ortográfico foi introduzido ou um produto foi atribuído à família de produtos errada. Um produto também será incluído na sublista Obsolete or mismatched options (Opções obsoletas ou incompatíveis) se tiver sido especificado para uma linha de base de patch anterior, mas não tiver patches disponibilizados pela Microsoft.

Solução: para evitar esse problema no console, sempre escolha opções das sublistas Currently available options (Opções atualmente disponíveis).

Você também pode visualizar os produtos com patches atualmente disponíveis usando o comando `describe-patch-properties` na AWS CLI ou o comando da API `DescribePatchProperties`.

Problema:**`AWS-RunPatchBaseline`**retorna um**`HRESULT`**(Windows Server)

Problema: você recebeu um erro semelhante ao seguinte:

```
----------ERROR-------
Invoke-PatchBaselineOperation : Exception Details: An error occurred when
attempting to search Windows Update.
Exception Level 1:
 Error Message: Exception from HRESULT: 0x80240437
 Stack Trace: at WUApiLib.IUpdateSearcher.Search(String criteria)..
(Windows updates)
11/22/2020 09:17:30 UTC | Info | Searching for Windows Updates.
11/22/2020 09:18:59 UTC | Error | Searching for updates resulted in error: Exception
 from HRESULT: 0x80240437
----------ERROR-------
failed to run commands: exit status 4294967295
```

Causa: esta saída indica que as APIs nativas do Windows Update não conseguiram executar as operações de aplicação de patches.

Solução: verifique o código `HResult` nos tópicos a seguir em microsoft.com para identificar as etapas de solução de problemas e resolver o erro:

- Códigos de erro do Windows Update por componente
- Erros comuns e mitigações do Windows Update

Problema: o nó gerenciado não tem acesso ao Catálogo do Windows Update ou ao WSUS

Problema: você recebeu um erro semelhante ao seguinte:

```
Downloading PatchBaselineOperations PowerShell module from https://s3.aws-api-
domain/path_to_module.zip to C:\Windows\TEMP\Amazon.PatchBaselineOperations-1.29.zip.

Extracting PatchBaselineOperations zip file contents to temporary folder.
```

```
Verifying SHA 256 of the PatchBaselineOperations PowerShell module files.

Successfully downloaded and installed the PatchBaselineOperations PowerShell module.

Patch Summary for

PatchGroup :

BaselineId :

Baseline : null

SnapshotId :

RebootOption : RebootIfNeeded

OwnerInformation :

OperationType : Scan

OperationStartTime : 1970-01-01T00:00:00.0000000Z

OperationEndTime : 1970-01-01T00:00:00.0000000Z

InstalledCount : -1

InstalledRejectedCount : -1

InstalledPendingRebootCount : -1

InstalledOtherCount : -1

FailedCount : -1

MissingCount : -1

NotApplicableCount : -1

UnreportedNotApplicableCount : -1

EC2AMAZ-VL3099P - PatchBaselineOperations Assessment Results - 2020-12-30T20:59:46.169

----------ERROR-------
```

```
Invoke-PatchBaselineOperation : Exception Details: An error occurred when attempting to
 search Windows Update.

Exception Level 1:

Error Message: Exception from HRESULT: 0x80072EE2

Stack Trace: at WUApiLib.IUpdateSearcher.Search(String criteria)

at
 Amazon.Patch.Baseline.Operations.PatchNow.Implementations.WindowsUpdateAgent.SearchForUpdates(

searchCriteria)

At C:\ProgramData\Amazon\SSM\InstanceData\i-02573cafcfEXAMPLE\document\orchestration
\3d2d4864-04b7-4316-84fe-eafff1ea58

e3\PatchWindows\_script.ps1:230 char:13

+ $response = Invoke-PatchBaselineOperation -Operation Install -Snapsho ...

+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

+ CategoryInfo : OperationStopped:
 (Amazon.Patch.Ba...UpdateOperation:InstallWindowsUpdateOperation) [Inv

oke-PatchBaselineOperation], Exception

+ FullyQualifiedErrorId : Exception Level 1:

Error Message: Exception Details: An error occurred when attempting to search Windows
 Update.

Exception Level 1:

Error Message: Exception from HRESULT: 0x80072EE2

Stack Trace: at WUApiLib.IUpdateSearcher.Search(String criteria)

at
 Amazon.Patch.Baseline.Operations.PatchNow.Implementations.WindowsUpdateAgent.SearchForUpdates(
 searc
```

```
---Error truncated----
```

Causa: este erro pode estar relacionado aos componentes do Windows Update ou a falta de conectividade com o Catálogo do Windows Update ou Windows Server Update Services (WSUS).

Solução: confirme se o nó gerenciado tem conectividade com o Catálogo do Microsoft Update por meio de um gateway da Internet, um gateway de NAT ou uma instância NAT. Se você estiver usando o WSUS, confirme se o nó gerenciado tem conectividade com o servidor WSUS em seu ambiente. Se a conectividade estiver disponível para o destino pretendido, verifique a documentação da Microsoft para outras causas potenciais do`HResult 0x80072EE2`. Isso pode indicar um problema no nível do sistema operacional.

Problema: o módulo PatchBaselineOperations PowerShell não pode ser baixado

Problema: você recebeu um erro semelhante ao seguinte:

```
Preparing to download PatchBaselineOperations PowerShell module from S3.

Downloading PatchBaselineOperations PowerShell module from https://s3.aws-api-
domain/path_to_module.zip to C:\Windows\TEMP\Amazon.PatchBaselineOperations-1.29.zip.
----------ERROR-------

C:\ProgramData\Amazon\SSM\InstanceData\i-02573cafcfEXAMPLE\document\orchestration
\aaaaaaaa-bbbb-cccc-dddd-4f6ed6bd5514\

PatchWindows\_script.ps1 : An error occurred when executing PatchBaselineOperations:
 Unable to connect to the remote server

+ CategoryInfo : NotSpecified: (:) [Write-Error], WriteErrorException

+ FullyQualifiedErrorId : Microsoft.PowerShell.Commands.WriteErrorException,_script.ps1

failed to run commands: exit status 4294967295
```

Solução: confira a conectividade e as permissões do nó gerenciado para o Amazon Simple Storage Service (Amazon S3). A função AWS Identity and Access Management (IAM) do nó gerenciado deve usar as permissões mínimas citadas em Comunicações do SSM Agent com os buckets do S3 gerenciados pela AWS. O nó deve se comunicar com o endpoint do Amazon S3 por meio do endpoint do gateway do Amazon S3, do gateway NAT ou do gateway da Internet. Para obter mais informações sobre os requisitos do endpoint da VPC para oAWS Systems Manager SSM Agent

(SSM Agent), consulte Melhorar a segurança das instâncias do EC2 usando endpoints da VPC para o Systems Manager.

Problema: patches ausentes

Problema:`AWS-RunPatchbaseline`concluída com êxito, mas há alguns patches ausentes.

Veja a seguir algumas causas comuns e suas soluções.

Causa 1: lista de referência não é eficaz.

Solução 1: Para verificar se essa é a causa, use o procedimento a seguir.

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Run Command.
3. Selecione o Histórico de comandos e selecione o comando cuja linha de base você deseja verificar.
4. Selecione o nó gerenciado que tiver patches ausentes.
5. Selecione Etapa 1: saída e encontre o valor de `BaselineId`.
6. Verifique a configuração da lista de referência de patches atribuída, ou seja, o sistema operacional, o nome do produto, a classificação e a gravidade da lista de referência do patch.
7. Acesse o Microsoft Update Catalog.
8. Pesquise os IDs de artigo da Base de Dados de Conhecimento Microsoft (KB) (por exemplo, KB3216916).
9. Verifique se o valor em Product (Produto) corresponde ao do nó gerenciado e selecione o Title (Título) correspondente. Uma nova janela Update Details (Atualizar detalhes) será aberta.
10. Na guia Visão geral, a classificação e a gravidade do MSRC devem corresponder à configuração da lista de referência de patches encontrada anteriormente.

Causa 2: o patch foi substituído.

Solução 2: Para conferir se isso de fato ocorreu, use o procedimento a seguir.

1. Acesse o Microsoft Update Catalog.
2. Pesquise os IDs de artigo da Base de Dados de Conhecimento Microsoft (KB) (por exemplo, KB3216916).

3. Verifique se o valor em Product (Produto) corresponde ao do nó gerenciado e selecione o Title (Título) correspondente. Uma nova janela Update Details (Atualizar detalhes) será aberta.
4. Acesse a guia Package Details (Detalhes do Pacote). Procure por uma entrada no cabeçalho Esta atualização foi substituída pelas seguintes atualizações:

Causa 3: o mesmo patch pode ter números de KB diferentes porque as atualizações online do WSUS e do Windows são tratadas como canais de lançamento independentes pela Microsoft.

Solução 3: Verifique a elegibilidade do patch. Se o pacote não estiver disponível no WSUS, instale a Compilação do SO 14393.3115. Se o pacote estiver disponível para todas as compilações do sistema operacional, instale as Compilações do SO 18362.1256 e 18363.1256.

Como usar runbooks do AWS Support Automation

O AWS Support fornece dois runbooks do Automation que você pode usar para solucionar determinados problemas relacionados à aplicação de patches.

- `AWSSupport-TroubleshootWindowsUpdate`: o runbook `AWSSupport-TroubleshootWindowsUpdate` é usado para identificar problemas que podem causar falhas nas atualizações do Windows Server para instâncias Windows Server do Amazon Elastic Compute Cloud (Amazon EC2).
- `AWSSupport-TroubleshootPatchManagerLinux`: o runbook `AWSSupport-TroubleshootPatchManagerLinux` soluciona problemas comuns que podem causar uma falha de aplicação de patch em nós gerenciados baseados em Linux ao usar o Patch Manager. O objetivo principal desse runbook é identificar a causa raiz da falha do comando de patch e sugerir um plano de correção.

> **Note**
>
> A execução de runbooks do Automation é cobrada. Para obter informações, consulte Preços do AWS Systems Manager para o Automation.

Entrar em contato com o AWS Support

Se não conseguir encontrar soluções para a solução de problemas nesta seção ou nos problemas do Systems Manager em AWS re:Post, e você tiver um plano do Suporte Developer, Business, or Enterprise, você poderá criar um caso de suporte técnico em AWS Support.

Antes de entrar em contato com o Suporte, colete os seguintes itens:

- Registros de agentes do SSM
- ID de comando, ID de janela de manutenção ou ID de execução de automação do Run Command
- Para os nós gerenciados do Windows Server, colete também o seguinte:
  - `%PROGRAMDATA%\Amazon\PatchBaselineOperations\Logs`, conforme descrito na guia Windows do Como os patches são instalados
  - Registros de atualização do Windows: ParaWindows Server2012 R2 e anteriores, use`%windir%/WindowsUpdate.log`. Para o Windows Server 2016 e mais recente, primeiro execute o comando do PowerShell`Get-WindowsUpdateLog`, antes de usar o `%windir%/WindowsUpdate.log`
- Para os nós gerenciados do Linux, colete também o seguinte:
  - O conteúdo do diretório `/var/lib/amazon/ssm/`*`instance-id`*`/document/orchestration/`*`Run-Command-execution-id`*`/awsrunShellScript/PatchLinux`

AWS Systems Manager Run Command

Usando o Run Command, uma ferramenta do AWS Systems Manager, é possível gerenciar remotamente e de forma segura a configuração de nós gerenciados. O nó gerenciado é qualquer instância do Amazon Elastic Compute Cloud (Amazon EC2) ou máquina que não é do EC2 em seu ambiente híbrido e multinuvem, que tenha sido configurada para o Systems Manager. O Run Command permite automatizar tarefas administrativas comuns e executar alterações de configuração únicas em escala. Você pode usar o Run Command a partir do AWS Management Console, da AWS Command Line Interface (AWS CLI), do AWS Tools for Windows PowerShell ou dos AWS SDKs. O Run Command é oferecido sem custo adicional. Para começar a usar o Run Command, abra o Systems Manager console (Console do gerenciador de sistemas). No painel de navegação, escolha Run Command.

Os administradores usam o Run Command para instalar ou fazer bootstrap de aplicações, criar um pipeline de implantação, capturar arquivos de log quando uma instância for removida de um grupo do Auto Scaling, integrar instâncias a um domínio do Windows e muito mais.

A API Run Command segue um modelo de consistência eventual devido à natureza distribuída do sistema que respalda a API. Isso significa que o resultado de um comando da API que for executado e que afete seus recursos poderá não estar imediatamente visível para todos os comandos

subsequentes que forem executados. Você deve ter isso em mente ao executar um comando de API que segue imediatamente um comando de API anterior.

Conceitos básicos

A tabela a seguir inclui informações para ajudá-lo a começar a trabalhar com o Run Command.

Tópico	Detalhes
Configurar nós gerenciados para o AWS Systems Manager	Verifique se você concluiu os requisitos de configuração para instâncias do Amazon Elastic Compute Cloud (Amazon EC2) e máquinas que não são do EC2 em um ambiente híbrido e multinuvem.
Gerenciar nós em ambientes híbridos e multinuvem com o Systems Manager	(Opcional) Registre servidores on-premises e VMs na AWS para que possa gerenciá-los usando o Run Command.
the section called “Gerenciar dispositivos de borda com o Systems Manager”	(Opcional) Configure dispositivos de borda para que você possa gerenciá-los usando Run Command.
Execução de comandos em nós gerenciados	Saiba como executar um comando destinado a um ou mais nós gerenciados usando o AWS Management Console.
Demonstrações do Run Command	Saiba como executar comandos usando as Tools for Windows PowerShell ou aAWS CLI.

Suporte ao EventBridge

Essa ferramenta do Systems Manager é compatível com um tipo evento e um tipo destino nas regras do Amazon EventBridge. Para obter informações, consulte Monitorar eventos do Systems Manager com o Amazon EventBridge e Referência: padrões e tipos de eventos do Amazon EventBridge para o Systems Manager.

Mais informações

- Executar o Run Command remotamente em uma instância do EC2 (tutorial de 10 minutos)

- Systems Manager service quotas no Referência geral da Amazon Web Services
- Referência de API do AWS Systems Manager

Tópicos

- Configurar o Run Command
- Execução de comandos em nós gerenciados
- Uso de códigos de saída em comandos
- Noções básicas sobre status de comando
- Demonstrações do Run Command
- Solução de problemas do Run Commando do Systems Manager

Configurar o Run Command

Antes de gerenciar nós usando o Run Command, uma ferramenta do AWS Systems Manager, configure uma política do AWS Identity and Access Management (IAM) para os usuários que executarão comandos. Se você usar alguma chave de condição global para a ação `SendCommand` em suas políticas do IAM, deverá incluir a chave de condição `aws:ViaAWSService` e definir o valor booleano como `true`. Veja um exemplo a seguir.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:document/YourDocument"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:SourceVpce": [
                        "vpce-1234567890abcdef0"
                    ]
```

```
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:document/YourDocument"
            ],
            "Condition": {
                "Bool": {
                    "aws:ViaAWSService": "true"
                }
            }
        }
    ]
}
```

Você também deve configurar os nós para o Systems Manager. Para obter mais informações, consulte Configurar nós gerenciados para o AWS Systems Manager.

Recomendamos concluir as tarefas de configuração opcionais a seguir para ajudar a minimizar o procedimento de segurança e o gerenciamento diário de seus nós gerenciados.

Monitorar execuções de comandos usando o Amazon EventBridge

É possível usar o EventBridge para registrar em log alterações no status da execução do comando. Você pode criar uma regra que será executada sempre que houver uma transição de estado ou quando houver uma transição para um ou mais estados que sejam de interesse. Você pode também especificar o Run Command como ação de destino quando ocorre um evento do EventBridge. Para obter mais informações, consulte Configurar o EventBridge para eventos do Systems Manager.

Monitorar execuções de comandos usando o Amazon CloudWatch Logs

É possível configurar o Run Command para enviar periodicamente todos os logs de erros e saída de comandos para um grupo de logs do Amazon CloudWatch. É possível monitorar esses logs de saída quase em tempo real, pesquisar valores, frases específicas ou padrões e criar alarmes com

base na pesquisa. Para obter mais informações, consulte Configurar o Amazon CloudWatch Logs para Run Command.

Restringir o acesso do Run Command a nós gerenciados específicos

Você pode restringir a capacidade de um usuário executar comandos em nós gerenciados usando o AWS Identity and Access Management (IAM). Especificamente, é possível criar uma política do IAM com uma condição em que o usuário poderá somente executar comandos em nós gerenciados que são marcados com etiquetas específicas. Para obter mais informações, consulte Restringir o acesso ao Run Command com base em etiquetas.

Restringir o acesso ao Run Command com base em etiquetas

Esta seção descreve como restringir a capacidade de um usuário executar comandos em nós gerenciados especificando uma condição de etiqueta em uma política do IAM. Os nós gerenciados incluem instâncias do Amazon EC2 e nós que não são do EC2 em um ambiente híbrido e multinuvem configurado para o Systems Manager. Embora as informações não sejam apresentadas explicitamente, você também pode restringir o acesso a dispositivos principais do AWS IoT Greengrass gerenciados. Para começar, você deve marcar com etiquetas os dispositivos do AWS IoT Greengrass. Para obter mais informações, consulte Marcar recursos do AWS IoT Greengrass Version 2 no Guia do desenvolvedor do AWS IoT Greengrass Version 2.

É possível restringir a execução de comandos a nós gerenciados específicos ao criar uma política do IAM que inclua uma condição em que o usuário poderá somente executar comandos em nós que estiverem marcados com etiquetas específicas. No exemplo a seguir, o usuário tem permissão para usar o Run Command (`Effect: Allow, Action: ssm:SendCommand`) usando qualquer documento do SSM (`Resource: arn:aws:ssm:*:*:document/*`) em qualquer nó (`Resource: arn:aws:ec2:*:*:instance/*`) com a condição de que o nó seja um Finance WebServer (`ssm:resourceTag/Finance: WebServer`). Se o usuário enviar um comando para um nó que não esteja marcado ou que possui qualquer outra etiqueta que não seja `Finance: WebServer`, os resultados da execução mostrarão `AccessDenied`.

JSON

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

```
         "Action":[
            "ssm:SendCommand"
         ],
         "Resource":[
            "arn:aws:ssm:*:*:document/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "ssm:SendCommand"
         ],
         "Resource":[
            "arn:aws:ec2:*:*:instance/*"
         ],
         "Condition":{
            "StringLike":{
               "ssm:resourceTag/Finance":[
                  "WebServers"
               ]
            }
         }
      }
   ]
}
```

Você pode criar políticas do IAM que permitem que um usuário execute comandos em nós gerenciados que são marcados com várias etiquetas. A política a seguir permite que o usuário execute comandos em nós gerenciados que têm duas etiquetas. Se um usuário enviar um comando para um nó que não esteja marcado com ambas as etiquetas, os resultados da execução mostrarão `AccessDenied`.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "ssm:SendCommand"
```

```
         ],
         "Resource":"*",
         "Condition":{
            "StringLike":{
               "ssm:resourceTag/tag_key1":[
                  "tag_value1"
               ],
               "ssm:resourceTag/tag_key2":[
                  "tag_value2"
               ]
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":[
            "ssm:SendCommand"
         ],
         "Resource":[
            "arn:aws:ssm:us-west-1::document/AWS-*",
            "arn:aws:ssm:us-east-2::document/AWS-*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "ssm:UpdateInstanceInformation",
            "ssm:ListCommands",
            "ssm:ListCommandInvocations",
            "ssm:GetDocument"
         ],
         "Resource":"*"
      }
   ]
}
```

Você também pode criar políticas do IAM que permitem que um usuário execute comandos em vários grupos de nós gerenciados marcados. A política de exemplo a seguir permite que o usuário execute comandos em um grupo de nós com etiquetas ou em ambos os grupos.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "ssm:SendCommand"
         ],
         "Resource":"*",
         "Condition":{
            "StringLike":{
               "ssm:resourceTag/tag_key1":[
                  "tag_value1"
               ]
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":[
            "ssm:SendCommand"
         ],
         "Resource":"*",
         "Condition":{
            "StringLike":{
               "ssm:resourceTag/tag_key2":[
                  "tag_value2"
               ]
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":[
            "ssm:SendCommand"
         ],
         "Resource":[
            "arn:aws:ssm:us-west-1::document/AWS-*",
            "arn:aws:ssm:us-east-2::document/AWS-*"
         ]
      },
```

```
      {
         "Effect":"Allow",
         "Action":[
            "ssm:UpdateInstanceInformation",
            "ssm:ListCommands",
            "ssm:ListCommandInvocations",
            "ssm:GetDocument"
         ],
         "Resource":"*"
      }
   ]
}
```

Para obter mais informações sobre como criar políticas do IAM, consulte Políticas gerenciadas e em linha no Guia do usuário do IAM. Para obter mais informações sobre como marcar nós gerenciados, consulte Editor de etiquetas no Guia do usuário do AWS Resource Groups.

Execução de comandos em nós gerenciados

Esta seção inclui informações sobre como enviar comandos do console AWS Systems Managerpara nós gerenciados. Ela também inclui informações sobre como cancelar um comando.

Observe que se seu nó estiver configurado com a opção de montagem `noexec` para o diretório, o Run Command não conseguirá executar comandos com sucesso.

> **Important**
>
> Ao enviar um comando usando o Run Command, não inclua informações confidenciais formatadas como texto sem formatação, como senhas, dados de configuração ou outros segredos. Todas as atividades de API do Systems Manager em sua conta são registradas em um bucket do S3 para logs do AWS CloudTrail. Isso significa que qualquer usuário com acesso ao bucket do S3 poderá visualizar os valores de texto simples desses segredos. Por esse motivo, recomendamos a criação e o uso de parâmetros `SecureString` para criptografar os dados sigilosos que você usa nas operações do Systems Manager. Para obter mais informações, consulte Restringir o acesso a parâmetros do Parameter Store usando políticas do IAM.

Retenção do histórico de execução

O histórico de cada comando está disponível por até 30 dias. Além disso, você pode armazenar uma cópia de todos os arquivos de log no Amazon Simple Storage Service ou ter uma trilha de auditoria de todas as chamadas de API em AWS CloudTrail.

Informações relacionadas

Para obter informações sobre como enviar comandos usando outras ferramentas, consulte os seguintes tópicos:

- Demonstração: Usar a AWS Tools for Windows PowerShell com o Run Command ou os exemplos na seção AWS Systems Manager da Referência do Cmdlet Ferramentas da AWS para PowerShell.
- Demonstração: Usar a AWS CLI com o Run Command ou os exemplos na Referência da CLI do SSM

Conteúdo

- Executar comandos no console
- Execução de comandos usando uma versão específica de documento
- Execução de comandos em escala
- Cancelar um comando

Executar comandos no console

Você pode usar o Run Command, uma ferramentado AWS Systems Manager no AWS Management Console, para configurar os nós gerenciados sem precisar fazer login neles. Este tópico inclui um exemplo que mostra como atualizar o SSM Agent em um nó gerenciado usando o Run Command.

Antes de começar

Antes de enviar um comando usando o Run Command, verifique se os nós gerenciados atendem a todo os requisitos de configuração do Systems Manager.

Para enviar um comando usando o Run Command

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Run Command.
3. Selecione Run command.

4. Na lista Command document (Documento de comando), escolha um documento do Systems Manager.
5. Na seção Command parameters, especifique valores para os parâmetros necessários.
6. Na seção Targets (Destinos), escolha os nós gerenciados nos quais você quer executar essa operação, especificando as etiquetas, selecionando as instâncias ou dispositivos de borda manualmente ou especificando um grupo de recursos.

   > **Tip**
   >
   > Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

7. Para Other parameters (Outros parâmetros):
   - Em Comment (Comentário), digite as informações sobre esse comando.
   - Em Timeout (seconds) (Tempo limite [segundos]), especifique o número de segundos para o sistema aguardar até a falha de execução do comando total.
8. Para Rate control (Controle de taxa):
   - Em Concurrency (Concorrência), especifique um número ou uma porcentagem de nós gerenciados nos quais executar o comando ao mesmo tempo.

     > **Note**
     >
     > Se você selecionou destinos especificando tags aplicadas a instâncias a nós gerenciados ou especificando grupos de recursos da AWS, e não tiver certeza de quantas instâncias são direcionadas, restrinja o número de instâncias que poderão executar o documento ao mesmo tempo, especificando uma porcentagem.

   - Em Error threshold (Limite de erro), especifique quando parar de executar o comando em outros nós depois de falhar em alguns ou em uma porcentagem de nós. Por exemplo, se você especificar três erros, o Systems Manager deixará de enviar o comando quando o 4° erro for recebido. Os nós gerenciados que continuam processando o comando também podem enviar erros.
9. (Opcional) Escolha um alarme do CloudWatch para aplicar ao seu comando para monitoramento. Para anexar um alarme do CloudWatch ao seu comando, a

entidade principal do IAM que executa o comando deve ter permissão para a ação `iam:createServiceLinkedRole`. Para obter mais informações sobre alarmes do CloudWatch, consulte Usar alarmes do Amazon CloudWatch. Observe que, se o alarme for ativado, quaisquer invocações de comando pendentes não serão executadas.

10. (Opcional) Em Output options (Opções de saída), para salvar a saída do comando em um arquivo, selecione a caixa Write command output to an S3 bucket (Gravar saída do comando em um bucket do S3). Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

    > **Note**
    >
    > As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil de instância (para instâncias do EC2) ou perfil de serviço do IAM (máquinas ativadas para ambientes híbridos) atribuído à instância, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço do IAM associado ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

11. Na seção SNS notifications (Notificações do SNS), se quiser enviar notificações sobre o status da execução do comando, marque a caixa de seleção Enable SNS notifications (Habilitar notificações do SNS).

    Para obter mais informações sobre a configuração de notificações do Amazon SNS para o Run Command, consulte Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS.

12. Escolha Executar.

Para obter informações sobre como cancelar um comando, consulte the section called “Cancelar um comando”.

Executar comandos novamente

O Systems Manager inclui duas opções para ajudar você a executar novamente um comando na página Run Command no console do Systems Manager.

- Rerun (Reexecutar): este botão permite que você execute o mesmo comando sem fazer alterações nele.
- Copy to new (Copiar para novo): este botão copia as configurações de um comando para um novo comando e dá a você a opção de editar essas configurações antes de executá-lo.

Como executar novamente um comando

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Run Command.
3. Escolha um comando a ser reexecutado. É possível executar novamente um comando imediatamente após sua execução na página de detalhes do comando. Ou, é possível escolher um comando que você executou anteriormente na guia Command history (Histórico de comandos).
4. Escolha Rerun (Executar novamente) para executar o mesmo comando sem alterações ou escolha Copy to new (Copiar para novo) para editar as configurações do comando antes de executá-lo.

Execução de comandos usando uma versão específica de documento

Você pode usar o parâmetro de versão de documento para especificar qual versão de um documento do AWS Systems Manager usar quando o comando for executado. Você pode especificar uma das seguintes opções para este parâmetro:

- $DEFAULT
- $LATEST
- Número da versão

Execute o procedimento a seguir para executar um comando usando o parâmetro de versão do documento.

Linux

Como executar comandos usando a AWS CLI em máquinas Linux locais

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Listar todos os documentos disponíveis

   Esse comando lista todos os documentos disponíveis para sua conta com base em permissões do AWS Identity and Access Management (IAM).

   ```
   aws ssm list-documents
   ```

3. Execute o comando a seguir para visualizar as diferentes versões de um documento. Substitua *nome do documento* pelas próprias informações.

   ```
   aws ssm list-document-versions \
       --name "document name"
   ```

4. Execute o comando a seguir para executar um comando que use uma versão do documento do SSM. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   ```
   aws ssm send-command \
       --document-name "AWS-RunShellScript" \
       --parameters commands="echo Hello" \
       --instance-ids instance-ID \
       --document-version '$LATEST'
   ```

Windows

Como executar comandos usando a AWS CLI em computadores Windows locais

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Listar todos os documentos disponíveis

   Esse comando lista todos os documentos disponíveis para sua conta com base em permissões do AWS Identity and Access Management (IAM).

   ```
   aws ssm list-documents
   ```

3. Execute o comando a seguir para visualizar as diferentes versões de um documento. Substitua *nome do documento* pelas próprias informações.

```
aws ssm list-document-versions ^
    --name "document name"
```

4. Execute o comando a seguir para executar um comando que use uma versão do documento do SSM. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

```
aws ssm send-command ^
    --document-name "AWS-RunShellScript" ^
    --parameters commands="echo Hello" ^
    --instance-ids instance-ID ^
    --document-version "$LATEST"
```

PowerShell

Para executar comandos usando as Tools for PowerShell

1. Instale e configure o Ferramentas da AWS para PowerShell (Ferramentas para Windows PowerShell), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar o Ferramentas da AWS para PowerShell.

2. Listar todos os documentos disponíveis

   Esse comando lista todos os documentos disponíveis para sua conta com base em permissões do AWS Identity and Access Management (IAM).

```
Get-SSMDocumentList
```

3. Execute o comando a seguir para visualizar as diferentes versões de um documento. Substitua *nome do documento* pelas próprias informações.

```
Get-SSMDocumentVersionList `
    -Name "document name"
```

4. Execute o comando a seguir para executar um comando que use uma versão do documento do SSM. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

```
Send-SSMCommand `
```

```
    -DocumentName "AWS-RunShellScript" `
    -Parameter @{commands = "echo helloWorld"} `
    -InstanceIds "instance-ID" `
    -DocumentVersion $LATEST
```

Execução de comandos em escala

É possível usar o Run Command, uma ferramenta do AWS Systems Manager, para executar comandos em uma frota de nós gerenciados usando `targets`. O parâmetro `targets` aceita uma combinação `Key,Value` baseada em etiquetas que você especificou para seus nós gerenciados. Quando você executa o comando, o sistema localiza e tenta executar o comando em todos os nós gerenciados que correspondem às etiquetas especificadas. Para obter mais informações sobre como etiquetar instâncias gerenciadas, consulte Tagging your AWS resources no Guia do usuário de recursos de marcação da AWS. Para obter informações sobre como marcar os dispositivos IoT gerenciados, consulte Marcar com etiqueta os recursos do AWS IoT Greengrass Version 2 no Guia do desenvolvedor do AWS IoT Greengrass Version 2.

Você também pode usar o parâmetro `targets` para direcionar uma lista de IDs de nós gerenciados específicos, conforme descrito na próxima seção.

Para controlar como os comandos são executados em centenas ou milhares de nós gerenciados, o Run Command também inclui parâmetros para restringir quantos nós podem processar simultaneamente uma solicitação e quantos erros podem ser gerados por um comando antes que ele seja cancelado.

Conteúdo

- Selecionar vários nós gerenciados como destino
- Usar controles de taxa

Selecionar vários nós gerenciados como destino

Você pode executar um comando e os nós gerenciados de destino especificando etiquetas, nomes de grupos de recursos da AWS ou IDs dos nós gerenciados.

Os exemplos a seguir mostram o formato do comando ao usar o Run Command do AWS Command Line Interface (AWS CLI). Substitua cada *`espaço reservado para recurso de exemplo`* por suas próprias informações. Os comandos de exemplo nesta seção são truncados usando `[...]`.

Exemplo 1: direcionar tags

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key=tag:tag-name,Values=tag-value \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --targets Key=tag:tag-name,Values=tag-value ^
    [...]
```

Exemplo 2: direcionar um grupo de recursos da AWS por nome

Você pode especificar um máximo de um nome de grupo de recursos por comando. Ao criar um grupo de recursos, recomendamos incluir `AWS::SSM:ManagedInstance` e `AWS::EC2::Instance` como tipos de recurso em seus critérios de agrupamento.

> **Note**
>
> Para enviar comandos que têm um grupo de recursos como destino, você deverá ter recebido permissões do AWS Identity and Access Management (IAM) para listar ou visualizar os recursos que pertencem a esse grupo. Para obter mais informações, consulte Configurar permissões no Guia do usuário do AWS Resource Groups.

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key=resource-groups:Name,Values=resource-group-name \
    [...]
```

Windows

```
aws ssm send-command ^
```

```
    --document-name document-name ^
    --targets Key=resource-groups:Name,Values=resource-group-name ^
    [...]
```

Exemplo 3: direcionar um grupo de recursos da AWS por tipo de recurso

Você pode especificar um máximo de cinco tipos de grupo de recursos por comando. Ao criar um grupo de recursos, recomendamos incluir `AWS::SSM:ManagedInstance` e `AWS::EC2::Instance` como tipos de recurso em seus critérios de agrupamento.

> **Note**
>
> Para enviar comandos que têm um grupo de recursos como destino, você deverá ter recebido permissões do IAM para listar ou visualizar os recursos que pertencem a esse grupo. Para obter mais informações, consulte Configurar permissões no Guia do usuário do AWS Resource Groups.

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key=resource-groups:ResourceTypeFilters,Values=resource-
type-1,resource-type-2 \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --targets Key=resource-groups:ResourceTypeFilters,Values=resource-
type-1,resource-type-2 ^
    [...]
```

Exemplo 4: IDs de instâncias como destino

Os exemplos a seguir mostram como direcionar os nós gerenciados usando a chave `instanceids` com o parâmetro `targets`. Você pode usar essa chave para utilizar os dispositivos principais do

AWS IoT Greengrass porque cada dispositivo recebe um mi-*ID_number*. É possível visualizar os IDs de dispositivos no Fleet Manager, uma ferramenta do AWS Systems Manager.

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key=instanceids,Values=instance-ID-1,instance-ID-2,instance-ID-3 \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --targets Key=instanceids,Values=instance-ID-1,instance-ID-2,instance-ID-3 ^
    [...]
```

Se você marcou nós gerenciados para diferentes ambientes usando uma `Key` chamada `Environment` e `Values` de `Development`, `Test`, `Pre-production` e `Production`, poderá enviar um comando para todas os nós gerenciados em um desses ambientes usando o parâmetro `targets` com a sintaxe a seguir.

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key=tag:Environment,Values=Development \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --targets Key=tag:Environment,Values=Development ^
    [...]
```

Você pode ter nós gerenciados adicionais como destino em outros ambientes, adicionando à lista `Values`. Separe itens usando vírgulas.

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key=tag:Environment,Values=Development,Test,Pre-production \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --targets Key=tag:Environment,Values=Development,Test,Pre-production ^
    [...]
```

Variação: refinar os destinos usando vários critérios `Key`

É possível refinar o número de destinos para o seu comando incluindo vários critérios para `Key`. Se você incluir mais de um critério para `Key`, o sistema usará os nós gerenciados que atenderem a todos os critérios. O seguinte comando direciona todos os nós gerenciados marcados para o Departamento de finanças e marcados para a função de servidor de banco de dados.

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key=tag:Department,Values=Finance Key=tag:ServerRole,Values=Database \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --targets Key=tag:Department,Values=Finance Key=tag:ServerRole,Values=Database ^
    [...]
```

Variação: usar vários critérios `Key` e `Value`

Expandindo o exemplo anterior, você pode direcionar vários departamentos e várias funções de servidor, incluindo itens adicionais nos critérios `Values`.

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key=tag:Department,Values=Finance,Marketing
 Key=tag:ServerRole,Values=WebServer,Database \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --targets Key=tag:Department,Values=Finance,Marketing
 Key=tag:ServerRole,Values=WebServer,Database ^
    [...]
```

Variação: definir nós gerenciados marcados como destino usando vários critérios `Values`

Se você marcou nós gerenciados para diferentes ambientes usando uma `Key` chamada `Department` e `Values` de `Sales` e `Finance`, poderá enviar um comando para todas os nós em um desses ambientes usando o parâmetro `targets` com a sintaxe a seguir.

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key=tag:Department,Values=Sales,Finance \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --targets Key=tag:Department,Values=Sales,Finance ^
    [...]
```

É possível especificar um máximo de cinco chaves e cinco valores para cada chave.

Se uma chave de etiqueta (o nome da etiqueta) ou um valor de etiqueta incluir espaços, você deverá incluir a chave ou o valor da etiqueta entre aspas, conforme mostrado nos exemplos a seguir.

Exemplo: espaços na tag `Value`

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key=tag:OS,Values="Windows Server 2016" \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --targets Key=tag:OS,Values="Windows Server 2016" ^
    [...]
```

Exemplo: espaços na chave `tag` e em `Value`

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key="tag:Operating System",Values="Windows Server 2016" \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --targets Key="tag:Operating System",Values="Windows Server 2016" ^
    [...]
```

Exemplo: espaços em um item em uma lista de `Values`

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key=tag:Department,Values="Sales","Finance","Systems Mgmt" \
```

```
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --targets Key=tag:Department,Values="Sales","Finance","Systems Mgmt" ^
    [...]
```

Usar controles de taxa

Você pode controlar a taxa na qual os comandos são enviados para nós gerenciados em um grupo usando controles de simultaneidade e controles de erro.

Tópicos

- Usar controles de simultaneidade
- Usar controles de erro

Usar controles de simultaneidade

Você pode controlar quantos nós gerenciados executam o comando ao mesmo tempo usando o parâmetro `max-concurrency` (as opções Concurrency [Simultaneidade] na página Run a command [Executar um comando]). Você pode especificar um número absoluto de nós gerenciados, por exemplo, **`10`** ou uma porcentagem do conjunto de destino, por exemplo, **`10%`**. O sistema de enfileiramento entrega o comando a um único nó e aguarda até que o sistema reconheça a invocação inicial antes de enviar o comando para mais dois nós. O sistema envia de forma exponencial comandos para mais nós até que o valor `max-concurrency` seja atingido. O padrão para o valor `max-concurrency` é 50. Os exemplos a seguir mostram como especificar valores para o parâmetro `max-concurrency`.

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --max-concurrency 10 \
    --targets Key=tag:Environment,Values=Development \
    [...]
```

```
aws ssm send-command \
    --document-name document-name \
    --max-concurrency 10% \
    --targets Key=tag:Department,Values=Finance,Marketing
 Key=tag:ServerRole,Values=WebServer,Database \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --max-concurrency 10 ^
    --targets Key=tag:Environment,Values=Development ^
    [...]
```

```
aws ssm send-command ^
    --document-name document-name ^
    --max-concurrency 10% ^
    --targets Key=tag:Department,Values=Finance,Marketing
 Key=tag:ServerRole,Values=WebServer,Database ^
    [...]
```

Usar controles de erro

Você também pode controlar a execução de um comando para centenas ou milhares de nós gerenciados definindo um limite de erro usando os parâmetros `max-errors` no campo Error threshold (Limitação de erros) na página Run a command (Executar um comando). O parâmetro especifica quantos erros são permitidos antes que o sistema pare de enviar o comando para nós gerenciados adicionais. Você pode especificar um número absoluto de erros (por exemplo, **`10`**) ou uma porcentagem do conjunto de destino (por exemplo, **`10%`**). Se você especificar **`3`**, por exemplo, o sistema deixará de enviar o comando quando o quarto erro for recebido. Se você especificar **`0`**, o sistema deixará de enviar o comando para nós adicionais depois que o primeiro resultado de erro for retornado. Se você enviar um comando para 50 nós gerenciados e definir `max-errors` como **`10%`**, o sistema deixará de enviar o comando para nós adicionais quando o sexto erro for recebido.

As invocações que já estão executando um comando quando `max-errors` é atingido podem ser concluídas, mas algumas dessas invocações também podem falhar. Para garantir que não haverá mais do que `max-errors` invocações com falha, defina `max-concurrency` como **`1`** para que as

invocações prossigam uma por vez. O padrão para o máximo de erros é 0. Os exemplos a seguir mostram como especificar valores para o parâmetro `max-errors`.

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --max-errors 10 \
    --targets Key=tag:Database,Values=Development \
    [...]
```

```
aws ssm send-command \
    --document-name document-name \
    --max-errors 10% \
    --targets Key=tag:Environment,Values=Development \
    [...]
```

```
aws ssm send-command \
    --document-name document-name \
    --max-concurrency 1 \
    --max-errors 1 \
    --targets Key=tag:Environment,Values=Production \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --max-errors 10 ^
    --targets Key=tag:Database,Values=Development ^
    [...]
```

```
aws ssm send-command ^
    --document-name document-name ^
    --max-errors 10% ^
    --targets Key=tag:Environment,Values=Development ^
    [...]
```

```
aws ssm send-command ^
    --document-name document-name ^
```

```
    --max-concurrency 1 ^
    --max-errors 1 ^
    --targets Key=tag:Environment,Values=Production ^
    [...]
```

Cancelar um comando

Você pode tentar cancelar um comando desde que o serviço mostre que ele está em um estado pendente ou em execução. No entanto, mesmo que um comando ainda esteja em um desses estados, não poderemos garantir que o comando será cancelado e o processo subjacente será interrompido.

Para cancelar um comando usando o console

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Run Command.
3. Selecione a invocação do comando que você deseja cancelar.
4. Escolha Cancel command (Cancelar comando).

Para cancelar um comando usando a AWS CLI

Execute o seguinte comando . Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

Linux & macOS

```
aws ssm cancel-command \
    --command-id "command-ID" \
    --instance-ids "instance-ID"
```

Windows

```
aws ssm cancel-command ^
    --command-id "command-ID" ^
    --instance-ids "instance-ID"
```

Para obter informações sobre o status de um comando cancelado, consulte Noções básicas sobre status de comando.

Uso de códigos de saída em comandos

Em alguns casos, pode ser necessário gerenciar a forma como lidar com seus comandos com o uso de códigos de saída.

Especifique códigos de saída nos comandos

Usando o Run Command, uma ferramenta do AWS Systems Manager, é possível especificar códigos de saída para determinar como os comandos são tratados. Por padrão, o código de saída do último comando executado em um script é relatado como o código de saída de todo o script. Por exemplo, você tem um script que contém três comandos. O primeiro falha, mas os seguintes são bem-sucedidos. Como o comando final foi bem-sucedido, o status da execução é relatado como `succeeded`.

Scripts de shell

Para falhar todo o script na primeira falha do comando, você pode incluir uma instrução condicional shell para sair do script se algum comando antes do final falhar. Use a abordagem a seguir.

```
<command 1>
    if [ $? != 0 ]
    then
        exit <N>
    fi
    <command 2>
    <command 3>
```

No exemplo a seguir, o script inteiro falhará se o primeiro comando falhar.

```
cd /test
    if [ $? != 0 ]
    then
        echo "Failed"
        exit 1
    fi
    date
```

Scripts PowerShell

O PowerShell requer que você chame `exit` explicitamente em seus scripts para o Run Command capturar com êxito o código de saída.

```
<command 1>
    if ($?) {<do something>}
    else {exit <N>}
    <command 2>
    <command 3>
    exit <N>
```

Exemplo:

```
cd C:\
    if ($?) {echo "Success"}
    else {exit 1}
    date
```

Tratamento de reinicializações ao executar comandos

Se você usar o Run Command, uma ferramenta do AWS Systems Manager, para executar scripts que reinicializam os nós gerenciados, recomendamos especificar um código de saída em seu script. Se você tentar reiniciar um nó em um script usando algum outro mecanismo, a execução do script status pode não ser atualizada corretamente, mesmo se a reinicialização for a última etapa em seu script. Para os nós gerenciados do Windows, especifique `exit 3010` em seu script. Para instâncias gerenciadas do Linux e do macOS especifique a `exit 194`. O código de saída instrui o Agente do AWS Systems Manager (SSM Agent) para reinicializar o nó gerenciado e reinicia o script depois que a reinicialização for concluída. Antes de iniciar a reinicialização, o SSM Agent informa o serviço do Systems Manager na nuvem que a comunicação será interrompida durante a reinicialização do servidor.

> **Note**
>
> O script de reinicialização não pode fazer parte de um plugin `aws:runDocument`. Se um documento contiver o script de reinicialização e outro documento tentar executar esse documento por meio do plugin `aws:runDocument`, o SSM Agent causará um erro.

Criar script idempotente

Ao desenvolver scripts que reinicializam nós gerenciados, torne os scripts idempotentes para que a execução do script continue de onde parou depois da reinicialização. Scripts impotentes gerenciam

o estado e validam se a ação foi realizada ou não. Isso evita que uma etapa seja executada várias vezes quando ela deve ser executada apenas uma vez.

Veja a seguir um exemplo de um script idempotente que reinicializa o nó diversas vezes.

```
$name = Get current computer name
If ($name –ne $desiredName)
    {
        Rename computer
        exit 3010
    }

$domain = Get current domain name
If ($domain –ne $desiredDomain)
    {
        Join domain
        exit 3010
    }

If (desired package not installed)
    {
        Install package
        exit 3010
    }
```

Exemplos

O script a seguir usa exemplos de códigos de saída para reiniciar nós gerenciados. O exemplo instala atualizações de pacote Linux no Amazon Linux e, em seguida, reinicia o nó. O exemplo do Windows Server instala a aplicação Telnet-Client em seu nó e, em seguida, o reinicia.

Amazon Linux

```
#!/bin/bash
yum -y update
needs-restarting -r
if [ $? -eq 1 ]
then
        exit 194
else
        exit 0
fi
```

Windows

```
$telnet = Get-WindowsFeature -Name Telnet-Client
if (-not $telnet.Installed)
    {
        # Install Telnet and then send a reboot request to SSM Agent.
        Install-WindowsFeature -Name "Telnet-Client"
        exit 3010
    }
```

Noções básicas sobre status de comando

O Run Command, uma ferramenta do AWS Systems Manager, relata informações detalhadas de status sobre os estados diferentes pelos quais um comando passa durante o processamento e para cada nó gerenciado que processou o comando. Você pode monitorar o status de comandos usando os seguintes métodos:

- Escolha o ícone Refresh (Atualizar) na guia Commands (Comandos) na interface do console Run Command.
- Chame list-commands ou list-command-invocations usando a AWS Command Line Interface (AWS CLI) Ou chame Get-SSMCommand ou Get-SSMCommandInvocation usando o AWS Tools for Windows PowerShell.
- Configure o Amazon EventBridge para responder a alterações de estado ou status.
- Configure o Amazon Simple Notification Service (Amazon SNS) para enviar notificações para todas as alterações de status ou para status específicos, como `Failed` ou `TimedOut`.

Status do Run Command

O Run Command relata detalhes de status para três áreas: plug-ins, invocações e um status de comando geral. Um plugin é um bloco de execução de código definido no documento do SSM do comando. Para obter mais informações sobre plug-ins, consulte Referência de plug-ins de documentos de comando.

Quando você envia um comando para vários nós gerenciados ao mesmo tempo, cada cópia do comando que é dirigida a cada nó é uma invocação de comando. Por exemplo, se você usar o documento `AWS-RunShellScript` e enviar um comando `ifconfig` para 20 instâncias Linux, esse comando terá 20 invocações. Cada invocação de comando comunica individualmente o status. Os plug-ins para uma determinada invocação de comando também comunicam individualmente o status.

Por fim, o Run Command inclui um status de comando agregado para todos os plug-ins e invocações. O status do comando agregado pode ser diferente do status relatado por plug-ins ou invocações, conforme observado nas tabelas a seguir.

> **Note**
>
> Se você executar comandos para vários nós gerenciados usando os parâmetros `max-concurrency` ou `max-errors`, o status do comando refletirá os limites impostos por esses parâmetros, conforme descrito nas tabelas a seguir. Para mais informações sobre esses parâmetros, consulte Execução de comandos em escala.

Status detalhado para plug-ins de comandos e invocações

Status	Detalhes
Pendente	O comando ainda não foi enviado para o nó gerenciado ou não foi recebido pelo SSM Agent. Se o comando não for recebido pelo agente antes do período de tempo que igual à soma dos parâmetros Timeout (seconds) (Tempo limite (segundos) e Execution timeout (Tempo limite de execução), o status será alterado para `Delivery Timed Out`.
InProgress	O Systems Manager está tentando enviar o comando para o nó gerenciado ou o comando foi recebido pelo SSM Agent e começou a ser executado na instância. Dependendo do resultado de todos os plugins de comando, o status mudará para `Success`, `Failed` `Delivery Timed Out` ou `Execution Timed Out`. Exceção: se o agente não estiver em execução ou disponível em um nó, o status do comando permanecerá em `In Progress` até que o agente esteja disponível novamente ou até que o limite de runtime seja atingido. O status mudará para um estado terminal.

Status	Detalhes
Atrasado	O sistema tentou enviar o comando para o nó gerenciado, mas não foi bem-sucedido. O sistema tenta novamente.

Status	Detalhes
Bem-sucedida	Este status é retornado sob várias condições . Esse status não significa que o comando foi processado nesse nó. Por exemplo, o comando pode ser recebido pelo SSM Agent no nó gerenciado e retornar um código de saída zero porque o PowerShell `ExecutionPolicy` impediu a execução do comando. Este é um estado terminal. As condições que resultam em um comando retornando um status `Success` são: • Ao segmentar uma única instância, o comando foi recebido pelo SSM Agent em seu nó gerenciado e retornou um código de saída igual a zero. • Ao segmentar várias instâncias, o número de invocações com falha não ultrapassou o limite de erro especificado no comando. • Ao segmentar várias instâncias, pelo menos uma invocação foi bem-sucedida, enquanto outras atingiram o tempo limite. O limite de erro especificado ainda se aplica. • Ao segmentar uma tag, nenhuma instância é encontrada associada à tag. • Ao segmentar várias instâncias, o número de invocações com falha não ultrapassou o limite de erro especificado no comando. • Ao segmentar uma tag, pelo menos uma invocação foi bem-sucedida, enquanto outras atingiram o tempo limite. O limite de erro especificado ainda se aplica. • Você tem aplicações ou políticas em vigor no nível do sistema operacional que impedem ou substituem a execução de um comando,

Status	Detalhes
	resultando no retorno de um código de saída zero. **Note** As mesmas condições se aplicam ao segmentar grupos de recursos. Para solucionar erros ou obter mais informações sobre a execução de comandos, envie um comando que manipule erros ou exceções retornando códigos de saída apropriados (códigos de saída diferentes de zero para falhas de comando).
DeliveryTimedOut	O comando não foi entregue ao nó gerenciado antes do tempo limite total expirado. Os tempos limite totais não contam para o limite de `max-errors` do comando pai, mas contribuem para determinar se o status do comando pai é `Success`, `Incomplete` ou `Delivery Timed Out`. Este é um estado terminal.
ExecutionTimedOut	A automação do comando foi iniciada no nó gerenciado, mas o comando não foi concluído antes do tempo limite de execução expirar. Os tempos limite de execução contam como uma falha, que enviará uma resposta diferente de zero e o Systems Manager encerrará a tentativa de executar a automação de comando e relatará um status de falha.

Status	Detalhes
Falha	O comando não foi bem-sucedido em seu nó gerenciado. Para um plugin, isso indica que o código do resultado não foi zero. Para uma invocação de comando, isso indica que o código de resultado para um ou mais plug-ins não foi zero. Falhas de invocação contam para o limite `max-errors` do comando pai. Este é um estado terminal.
Cancelado	O comando foi cancelado antes de ser concluído. Este é um estado terminal.
Não é possível entregar	O comando não pode ser entregue ao nó gerenciado. O nó pode não existir ou pode não estar respondendo. Invocações não entregues não contam para o limite de `max-errors` do comando pai, mas não contribuem para determinar se o status do comando pai é `Success` ou `Incomplete`. Por exemplo, se todas as invocações em um comando tiverem o status `Undeliverable`, o status do comando retornado será `Failed`. No entanto, se um comando tiver cinco invocações, quatro das quais retornarem o status `Undeliverable` e uma retornar o status `Success`, o status do comando pai será `Success`. Este é um estado terminal.
Rescindido	O comando pai excedeu o limite de `max-errors` e as invocações de comando subsequentes foram canceladas pelo sistema. Este é um estado terminal.

Status	Detalhes
InvalidPlatform	O comando foi enviado a um nó gerenciad o que não correspondeu às plataformas necessárias especificadas pelo documento escolhido. O `Invalid Platform` não é considerado para o limite máximo de erros do comando pai, mas contribui para determina r se o status do comando pai é Success (Êxito) ou Failed (Falha). Por exemplo, se todas as invocações em um comando tiverem o status `Invalid Platform`, o status do comando retornado será `Failed`. No entanto, se um comando tiver cinco invocações, quatro das quais retornarem o status `Invalid Platform` e uma retornar o status `Success`, o status do comando pai será `Success`. Este é um estado terminal.
AccessDenied	O usuário ou a função do AWS Identity and Access Management (IAM) que inicia o comando não tem acesso ao nó gerenciado de destino. O `Access Denied` não é considera do no limite `max-errors` do comando pai, mas contribuirá se o status do comando pai for `Success` ou `Failed`. Por exemplo, se todas as invocações em um comando tiverem o status `Access Denied`, o status do comando retornado será `Failed`. No entanto, se um comando tiver cinco invocações, quatro das quais retornarem o status `Access Denied` e uma retornar o status `Success`, o status do comando pai será `Success`. Este é um estado terminal.

Status detalhado de um comando

Status	Detalhes
Pendente	O comando ainda não foi recebido por um agente em um nó gerenciado.
InProgress	O comando foi enviado para pelo menos um nó gerenciado, mas não chegou a um estado final em todos os nós.
Atrasado	O sistema tentou enviar o comando para o nó, mas não foi bem-sucedido. O sistema tenta novamente.
Bem-sucedida	O comando foi recebido pelo SSM Agent em todos nós gerenciados especificados ou definidos como destino e retornou um código de saída igual a zero. Todas as invocações de comando atingiram um estado terminal e o valor de `max-errors` não foi alcançado. Esse status não significa que o comando foi processado com sucesso em todos os nós gerenciados especificados ou de destino. Este é um estado terminal. **Note** Para solucionar erros ou obter mais informações sobre a execução de comandos, envie um comando que manipule erros ou exceções retornando códigos de saída apropriados (códigos de saída diferentes de zero para falhas de comando).
DeliveryTimedOut	O comando não foi entregue ao nó gerenciado antes do tempo limite total expirado. O valor

Status	Detalhes
	de `max-errors` ou mais invocações de comandos mostram um status de `Delivery Timed Out`. Este é um estado terminal.
Falha	O comando não foi bem-sucedido em seu nó gerenciado. O valor de `max-errors` ou mais invocações de comandos mostram um status de `Failed`. Este é um estado terminal.
Incompleto	O comando foi tentado em todos os nós gerenciados, e uma ou mais das invocações não tem um valor de `Success`. No entanto, não houve um número suficiente de invocaçõe s com falha para que o status fosse `Failed`. Este é um estado terminal.
Cancelado	O comando foi cancelado antes de ser concluído. Este é um estado terminal.
RateExceeded	O número de nós gerenciados visados pelo comando excedeu o limite da conta para invocações pendentes. O sistema cancelou o comando antes de executá-lo em qualquer nó. Este é um estado terminal.

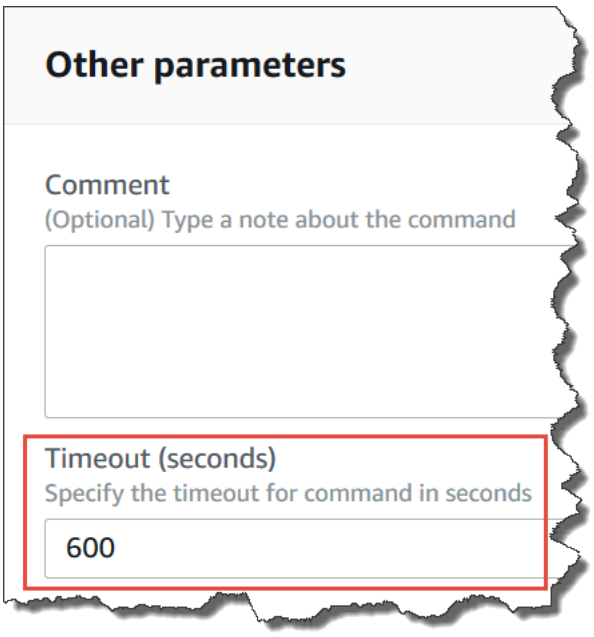
Status	Detalhes
AccessDenied	O usuário ou o perfil que inicia o comando não tem acesso ao grupo de recursos de destino. O `AccessDenied` não é contabilizado em relação ao limite `max-errors` do comando pai, mas contribuirá se o status do comando pai for `Success` ou `Failed`. (Por exemplo, se todas as invocações em um comando tiverem o status `AccessDenied` , então o status do comando retornado será `Failed`. No entanto, se um comando tiver cinco invocações, quatro das quais retornarem o status `AccessDenied` e uma retornou o status `Success`, então o status do comando pai será `Success`.) Este é um estado terminal.
Nenhuma instância na tag	O valor do par de chaves da etiqueta ou o grupo de recursos visado pelo comando não corresponde a nenhum nó gerenciado. Este é um estado terminal.

Noções básicas sobre valores de tempo limite de comandos

O Systems Manager impõe os seguintes valores de tempo limite ao executar comandos.

Tempo limite total

No console do Systems Manager, especifique o valor do tempo limite de entrega no campo Timeout (seconds) (Tempo limite [segundos]). Depois que um comando é enviado, o Run Command verifica se o comando expirou ou não. Se um comando atingir o limite de expiração do comando (tempo limite total), ele altera o status para`DeliveryTimedOut`para todas as invocações que têm o status`InProgress`,`Pending`ou`Delayed`.



Em um nível mais técnico, o Timeout (seconds) (Tempo limite [segundos]) total é uma combinação de dois valores de tempo limite, como mostrado aqui:

```
Total timeout = "Timeout(seconds)" from the console + "timeoutSeconds":
"{{ executionTimeout }}" from your SSM document
```

Por exemplo, o valor padrão de Timeout (seconds) (Tempo limite em segundos) no console do Systems Manager é de 600 segundos. Se você executar um comando usando o comando`AWS-RunShellScript`Documento do SSM, o valor padrão de“TimeoutSeconds”: “{{executionTimeout}}”é 3600 segundos, como mostrado na seguinte amostra de documento:

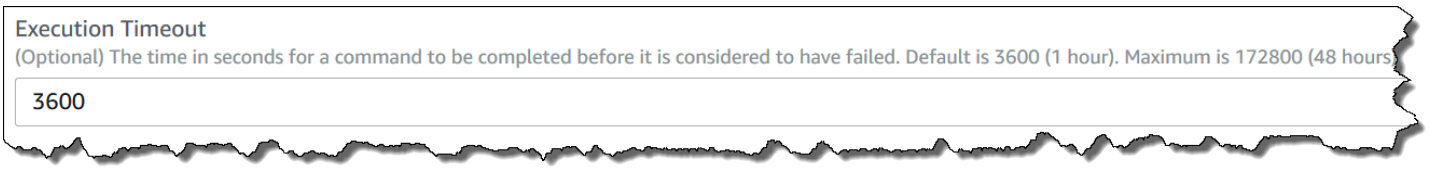
```
"executionTimeout": {
    "type": "String",
    "default": "3600",

"runtimeConfig": {
  "aws:runShellScript": {
    "properties": [
      {
        "timeoutSeconds": "{{ executionTimeout }}"
```

Isso significa que o comando é executado por 4.200 segundos (70 minutos) antes que o sistema defina o status do comando como `DeliveryTimedOut`.

Tempo limite de execução

No console do Systems Manager, especifique o valor do tempo limite de execução no campo Execution Timeout (Tempo limite de execução), se disponível. Nem todos os documentos do SSM exigem especificar um tempo limite de execução. O campo Execution Timeout (Tempo limite de execução) é exibido somente quando um parâmetro de entrada correspondente foi definido no documento SSM. Se especificado, o comando deve ser concluído dentro desse período.

> **Note**
>
> O Run Command depende da resposta do terminal SSM Agent para determinar se o comando foi entregue ou não ao agente. O SSM Agent deve enviar um sinal do `ExecutionTimed0ut` para uma invocação ou comando a ser marcado como `ExecutionTimed0ut`.



Tempo limite de execução padrão

Se um documento do SSM do não exigir especificar explicitamente um valor de tempo limite de execução, o Systems Manager vai impor o tempo limite de execução padrão codificado.

Como o Systems Manager relata os tempos limite

Se o Systems Manager receber uma resposta `execution timeout` do SSM Agent em um destino, o Systems Manager marcará a invocação do comando como `executionTimeout`.

SeRun Commandnão recebe uma resposta de terminal de documento deSSM Agent, a invocação do comando é marcada como`deliveryTimeout`.

Para determinar o status de tempo limite em um destino, o SSM Agent combina todos os parâmetros e o conteúdo do documento do SSM para o qual será calculado `executionTimeout`. Quando o SSM Agent determinar que um comando expirou, ele enviará `executionTimeout` para o serviço.

O padrão paraTempo limite (segundos)é de 3600 segundos. O padrão paraTempo limite de execuçãotambém é 3600 segundos. Portanto, o tempo limite padrão total para um comando é 7200 segundos.

> **Note**
>
> O SSM Agent processa `executionTimeout` de forma diferente dependendo do tipo de documento do SSM e da versão do documento.

Demonstrações do Run Command

As demonstrações nesta seção mostram como executar comandos com o Run Command, uma ferramenta do AWS Systems Manager, usando a AWS Command Line Interface (AWS CLI) ou o AWS Tools for Windows PowerShell.

Conteúdo

- Atualização de softwares usando Run Command
- Demonstração: Usar a AWS CLI com o Run Command
- Demonstração: Usar a AWS Tools for Windows PowerShell com o Run Command

Você também pode ver comandos de exemplo nas seguintes referências.

- Systems Manager (Gerenciador de sistemas)AWS CLIReferência do
- AWS Tools for Windows PowerShell - AWS Systems Manager

Atualização de softwares usando Run Command

O procedimento a seguir descreve como atualizar softwares em seus nós gerenciados.

Atualização do SSM Agent por meio de Run Command

O procedimento a seguir descreve como atualizar o SSM Agent em execução em seus nós gerenciados. Você pode atualizar para a versão mais recente do SSM Agent ou fazer downgrade para uma versão mais antiga. Quando você executa o comando, o sistema faz download da versão da AWS, instala essa versão e desinstala a versão que existia antes do comando ser executado. Se ocorrer um erro durante esse processo, o sistema retornará à versão no servidor antes de o comando ser executado, e o status do comando mostrará que o comando falhou.

> **Note**
>
> Se uma instância estiver sendo executada no macOS versão 11.0 (Big Sur) ou posterior, a instância deverá ter o SSM Agent versão 3.1.941.0 ou superior para executar o documento AWS-UpdateSSMAgent. Se a instância estiver executando uma versão do SSM Agent lançada antes da 3.1.941.0, você poderá atualizar o SSM Agent para executar o documento AWS-UpdateSSMAgentexecutando os comandos `brew upgrade amazon-ssm-agent` e `brew update`.

Para receber notificações sobre atualizações do SSM Agent, inscreva-se na página Notas de versão do SSM Agent no GitHub.

Para atualizar o SSM Agent usando o Run Command

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Run Command.
3. Selecione Run command.
4. Na lista Command document (Documento do comando), escolha **`AWS-UpdateSSMAgent`**.
5. Na seção Command parameters, especifique valores para os parâmetros a seguir, se desejar:
   a. (Opcional) Em Version (Versão), digite a versão do SSM Agent a ser instalada. Você pode instalar versões antigas do agente. Se você não especificar uma versão, o serviço instalará a versão mais recente.
   b. (Opcional) Para Allow Downgrade (Permitir downgrade), escolha true para instalar uma versão anterior do SSM Agent. Se você escolher essa opção, especifique o número da versão antiga. Escolha false para instalar apenas a versão mais recente do serviço.
6. Na seção Targets (Destinos), escolha os nós gerenciados nos quais você quer executar essa operação, especificando as etiquetas, selecionando as instâncias ou dispositivos de borda manualmente ou especificando um grupo de recursos.

> **Tip**
>
> Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

7. Para Other parameters (Outros parâmetros):

   - Em Comment (Comentário), digite as informações sobre esse comando.
   - Em Timeout (seconds) (Tempo limite [segundos]), especifique o número de segundos para o sistema aguardar até a falha de execução do comando total.

8. Para Rate control (Controle de taxa):

   - Em Concurrency (Concorrência), especifique um número ou uma porcentagem de nós gerenciados nos quais executar o comando ao mesmo tempo.

     > **Note**
     >
     > Se você selecionou destinos especificando tags aplicadas a instâncias a nós gerenciados ou especificando grupos de recursos da AWS, e não tiver certeza de quantas instâncias são direcionadas, restrinja o número de instâncias que poderão executar o documento ao mesmo tempo, especificando uma porcentagem.

   - Em Error threshold (Limite de erro), especifique quando parar de executar o comando em outros nós depois de falhar em alguns ou em uma porcentagem de nós. Por exemplo, se você especificar três erros, o Systems Manager deixará de enviar o comando quando o 4° erro for recebido. Os nós gerenciados que continuam processando o comando também podem enviar erros.

9. (Opcional) Em Output options (Opções de saída), para salvar a saída do comando em um arquivo, selecione a caixa Write command output to an S3 bucket (Gravar saída do comando em um bucket do S3). Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

   > **Note**
   >
   > As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil de instância (para instâncias do EC2) ou perfil de serviço do IAM (máquinas ativadas para ambientes híbridos) atribuído à instância, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço do IAM associado ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

10. Na seção SNS notifications (Notificações do SNS), se quiser enviar notificações sobre o status da execução do comando, marque a caixa de seleção Enable SNS notifications (Habilitar notificações do SNS).

    Para obter mais informações sobre a configuração de notificações do Amazon SNS para o Run Command, consulte Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS.

11. Escolha Executar.

Atualização da PowerShell por meio de Run Command

O procedimento a seguir descreve como atualizar o PowerShell para a versão 5.1 em seus nós gerenciados do Windows Server 2012 e 2012 R2. O script fornecido neste procedimento transfere a actualização do Windows Management Framework (WMF) versão 5.1 e inicia a instalação da actualização. Os nós são reinicializados durante esse processo porque isso é necessário ao instalar o WMF 5.1. O download e a instalação da atualização demoram aproximadamente cinco minutos para serem concluídos.

Para atualizar o PowerShell usandoRun Command

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Run Command.
3. Selecione Run command.
4. Na lista Command document (Documento do comando), escolha **AWS-RunPowerShellScript**.
5. NoComandosColar os comandos a seguir para o sistema operacional.

   Windows Server 2012 R2

```
Set-Location -Path "C:\Windows\Temp"

Invoke-WebRequest "https://go.microsoft.com/fwlink/?linkid=839516" -OutFile
 "Win8.1AndW2K12R2-KB3191564-x64.msu"

Start-Process -FilePath "$env:systemroot\system32\wusa.exe" -Verb RunAs -
ArgumentList ('Win8.1AndW2K12R2-KB3191564-x64.msu', '/quiet')
```

Windows Server 2012

```
Set-Location -Path "C:\Windows\Temp"

Invoke-WebRequest "https://go.microsoft.com/fwlink/?linkid=839513" -OutFile
 "W2K12-KB3191565-x64.msu"

Start-Process -FilePath "$env:systemroot\system32\wusa.exe" -Verb RunAs -
ArgumentList ('W2K12-KB3191565-x64.msu', '/quiet')
```

6. Na seção Targets (Destinos), escolha os nós gerenciados nos quais você quer executar essa operação, especificando as etiquetas, selecionando as instâncias ou dispositivos de borda manualmente ou especificando um grupo de recursos.

> **Tip**
>
> Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

7. Para Other parameters (Outros parâmetros):

   - Em Comment (Comentário), digite as informações sobre esse comando.
   - Em Timeout (seconds) (Tempo limite [segundos]), especifique o número de segundos para o sistema aguardar até a falha de execução do comando total.

8. Para Rate control (Controle de taxa):

   - Em Concurrency (Concorrência), especifique um número ou uma porcentagem de nós gerenciados nos quais executar o comando ao mesmo tempo.

   > **Note**
   >
   > Se você selecionou destinos especificando tags aplicadas a instâncias a nós gerenciados ou especificando grupos de recursos da AWS, e não tiver certeza de quantas instâncias são direcionadas, restrinja o número de instâncias que poderão executar o documento ao mesmo tempo, especificando uma porcentagem.

   - Em Error threshold (Limite de erro), especifique quando parar de executar o comando em outros nós depois de falhar em alguns ou em uma porcentagem de nós. Por exemplo, se você

especificar três erros, o Systems Manager deixará de enviar o comando quando o 4° erro for recebido. Os nós gerenciados que continuam processando o comando também podem enviar erros.

9. (Opcional) Em Output options (Opções de saída), para salvar a saída do comando em um arquivo, selecione a caixa Write command output to an S3 bucket (Gravar saída do comando em um bucket do S3). Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

   > **Note**
   >
   > As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil de instância (para instâncias do EC2) ou perfil de serviço do IAM (máquinas ativadas para ambientes híbridos) atribuído à instância, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço do IAM associado ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

10. Na seção SNS notifications (Notificações do SNS), se quiser enviar notificações sobre o status da execução do comando, marque a caixa de seleção Enable SNS notifications (Habilitar notificações do SNS).

    Para obter mais informações sobre a configuração de notificações do Amazon SNS para o Run Command, consulte Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS.

11. Escolha Executar.

Depois que a reinicialização do nó gerenciado e a instalação da atualização estiver concluída, conecte-se ao nó para confirmar se o PowerShell foi atualizado com êxito para a versão 5.1. Para conferir a versão do PowerShell em seu nó, abra o PowerShell e digite `$PSVersionTable`. O`PSVersion`na tabela de saída mostra 5.1 se a atualização foi bem-sucedida.

Se o `PSVersion` for diferente de 5.1, por exemplo 3.0 ou 4.0, consulte os logs de Configuração no Visualizador de eventos em Logs do Windows. Esses logs indicam por que a instalação da atualização falhou.

Demonstração: Usar a AWS CLI com o Run Command

A demonstração de exemplo a seguir mostra como usar a AWS Command Line Interface (AWS CLI) para ver informações sobre comandos e parâmetros de comando, como executar comandos e como visualizar o status desses comandos.

> **Important**
>
> Apenas administradores confiáveis devem ter permissão para usar os documentos pré-configurados do AWS Systems Manager mostrados neste tópico. Os comandos ou scripts especificados em documentos do Systems Manager são executados com permissões administrativas em seus nós. Se um usuário tiver permissão para executar qualquer um dos documentos do Systems Manager predefinidos (qualquer documento que comece com `AWS-`), ele também terá acesso de administrador ao nó. Para todos os outros usuários, você deve criar documentos restritivos e compartilhá-los com usuários específicos.

Tópicos

- Etapa 1: Conceitos básicos
- Etapa 2: Executar scripts de shell para exibir detalhes de recursos
- Etapa 3: Enviar comandos simples usando o documento AWS-RunShellScript
- Etapa 4: Executar um script Python simples usando o Run Command
- Etapa 5: Rodar um script Bash usandoRun Command

Etapa 1: Conceitos básicos

Você deve ter permissões de administrador sobre o nó gerenciado que deseja configurar ou deve ter recebido a permissão apropriada no AWS Identity and Access Management (IAM). Veja também que o exemplo usa região Leste dos EUA (Ohio) (us-east-2). O Run Command está disponível nas Regiões da AWS listadas em Systems Manager service endpoints no Referência geral da Amazon Web Services. Para obter mais informações, consulte Configurar nós gerenciados para o AWS Systems Manager.

Para executar comandos usando a AWS CLI

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Liste todos os documentos disponíveis.

   Esse comando lista todos os documentos disponíveis para sua conta com base em permissões do IAM.

   ```
   aws ssm list-documents
   ```

3. Verifique se um nó está pronto para receber comandos.

   A saída do comando a seguir mostra se os nós gerenciados estão online.

   Linux & macOS

   ```
   aws ssm describe-instance-information \
       --output text --query "InstanceInformationList[*]"
   ```

   Windows

   ```
   aws ssm describe-instance-information ^
       --output text --query "InstanceInformationList[*]"
   ```

4. Execute o comando a seguir para ver detalhes sobre um nó gerenciado específico.

   > **Note**
   >
   > Para executar os comandos nessa demonstração, substitua os IDs de instância e comando. Para os dispositivos gerenciados principais do AWS IoT Greengrass, use o mi-*ID_number* para o ID da instância. O ID do comando é retornado como uma resposta ao send-command. Os IDs de instância estão disponíveis no Fleet Manager, uma ferramenta do AWS Systems Manager.

   Linux & macOS

   ```
   aws ssm describe-instance-information \
       --instance-information-filter-list key=InstanceIds,valueSet=instance-ID
   ```

   Windows

   ```
   aws ssm describe-instance-information ^
   ```

```
    --instance-information-filter-list key=InstanceIds,valueSet=instance-ID
```

Etapa 2: Executar scripts de shell para exibir detalhes de recursos

Por meio do Run Command e do documento `AWS-RunShellScript`, você pode executar qualquer comando ou script em um nó gerenciado, como se estivesse conectado localmente.

Veja a descrição e os parâmetros disponíveis

Use o seguinte comando para visualizar uma descrição do documento JSON do Systems Manager.

Linux & macOS

```
aws ssm describe-document \
    --name "AWS-RunShellScript" \
    --query "[Document.Name,Document.Description]"
```

Windows

```
aws ssm describe-document ^
    --name "AWS-RunShellScript" ^
    --query "[Document.Name,Document.Description]"
```

Use o comando a seguir para visualizar os parâmetros e detalhes disponíveis sobre esses parâmetros.

Linux & macOS

```
aws ssm describe-document \
    --name "AWS-RunShellScript" \
    --query "Document.Parameters[*]"
```

Windows

```
aws ssm describe-document ^
    --name "AWS-RunShellScript" ^
    --query "Document.Parameters[*]"
```

Etapa 3: Enviar comandos simples usando o documento **`AWS-RunShellScript`**

Execute o comando a seguir para obter informações de IP para um nó do Linux.

Se você estiver segmentando um nó gerenciado do Windows Server, altere o `document-name` para `AWS-RunPowerShellScript` e altere o `command` de `ifconfig` para `ipconfig`.

Linux & macOS

```
aws ssm send-command \
    --instance-ids "instance-ID" \
    --document-name "AWS-RunShellScript" \
    --comment "IP config" \
    --parameters commands=ifconfig \
    --output text
```

Windows

```
aws ssm send-command ^
    --instance-ids "instance-ID" ^
    --document-name "AWS-RunShellScript" ^
    --comment "IP config" ^
    --parameters commands=ifconfig ^
    --output text
```

Obter informações de comando com dados de resposta

O comando a seguir usa o ID de Comando que foi retornado do comando anterior para obter os detalhes e os dados de resposta da execução do comando. O sistema retorna os dados da resposta se o comando for concluído. Se a execução do comando mostrar `"Pending"` ou `"InProgress"`, execute esse comando novamente para ver os dados de resposta.

Linux & macOS

```
aws ssm list-command-invocations \
    --command-id $sh-command-id \
    --details
```

Windows

```
aws ssm list-command-invocations ^
```

```
    --command-id $sh-command-id ^
    --details
```

Identificar usuário

O comando a seguir exibe o usuário padrão que executa os comandos.

Linux & macOS

```
sh_command_id=$(aws ssm send-command \
    --instance-ids "instance-ID" \
    --document-name "AWS-RunShellScript" \
    --comment "Demo run shell script on Linux managed node" \
    --parameters commands=whoami \
    --output text \
    --query "Command.CommandId")
```

Obter status do comando

O comando a seguir usa o ID de Comando para obter o status da execução do comando em nós gerenciados. Este exemplo usa o ID de comando que foi retornado no comando anterior.

Linux & macOS

```
aws ssm list-commands \
    --command-id "command-ID"
```

Windows

```
aws ssm list-commands ^
    --command-id "command-ID"
```

Obter detalhes do comando

O comando a seguir usa o ID do comando anterior para obter o status da execução do comando por nó gerenciado.

Linux & macOS

```
aws ssm list-command-invocations \
    --command-id "command-ID" \
    --details
```

Windows

```
aws ssm list-command-invocations ^
    --command-id "command-ID" ^
    --details
```

Obter informações de comando com dados de resposta para um nó gerenciado específico

O comando a seguir retorna a saída da solicitação `aws ssm send-command` original para um nó específico.

Linux & macOS

```
aws ssm list-command-invocations \
    --instance-id instance-ID \
    --command-id "command-ID" \
    --details
```

Windows

```
aws ssm list-command-invocations ^
    --instance-id instance-ID ^
    --command-id "command-ID" ^
    --details
```

Exibir versão do Python

O comando a seguir retorna a versão do Python em execução em um nó.

Linux & macOS

```
sh_command_id=$(aws ssm send-command \
    --instance-ids "instance-ID" \
```

```
    --document-name "AWS-RunShellScript" \
    --comment "Demo run shell script on Linux Instances" \
    --parameters commands='python -V' \
    --output text --query "Command.CommandId") \
    sh -c 'aws ssm list-command-invocations \
    --command-id "$sh_command_id" \
    --details \
    --query "CommandInvocations[].CommandPlugins[].{Status:Status,Output:Output}"'
```

Etapa 4: Executar um script Python simples usando o Run Command

O comando a seguir executa um script simples Python “Hello World” usando o Run Command.

Linux & macOS

```
sh_command_id=$(aws ssm send-command \
    --instance-ids "instance-ID" \
    --document-name "AWS-RunShellScript" \
    --comment "Demo run shell script on Linux Instances" \
    --parameters '{"commands":["#!/usr/bin/python","print \"Hello World from python
\""]}' \
    --output text \
    --query "Command.CommandId") \
    sh -c 'aws ssm list-command-invocations \
    --command-id "$sh_command_id" \
    --details \
    --query "CommandInvocations[].CommandPlugins[].{Status:Status,Output:Output}"'
```

Etapa 5: Rodar um script Bash usandoRun Command

Os exemplos nesta seção demonstram como executar o seguinte script bash usandoRun Command.

Para obter exemplos de uso do Run Command para executar scripts armazenados em locais remotos, consulte Executar scripts no Amazon S3 e Executar scripts do GitHub.

```
#!/bin/bash
yum -y update
yum install -y ruby
cd /home/ec2-user
curl -O https://aws-codedeploy-us-east-2.s3.amazonaws.com/latest/install
chmod +x ./install
```

```
./install auto
```

Este script instala o agente do AWS CodeDeploy no Amazon Linux e no Red Hat Enterprise Linux (RHEL), como descrito em Criar uma instância do Amazon EC2 para CodeDeploy no Guia do usuário do AWS CodeDeploy.

O script instala o agente CodeDeploy de um bucket do S3 gerenciado pela AWS na região Leste dos EUA (Ohio) (us-east-2), `aws-codedeploy-us-east-2`.

Execute um script bash em umAWS CLIComando da

O exemplo a seguir demonstra como incluir o script bash em um comando da CLI usando o comando`--parametersopção`.

Linux & macOS

```
aws ssm send-command \
    --document-name "AWS-RunShellScript" \
    --targets '[{"Key":"InstanceIds","Values":["instance-id"]}]' \
    --parameters '{"commands":["#!/bin/bash","yum -y update","yum
 install -y ruby","cd /home/ec2-user","curl -O https://aws-codedeploy-us-
east-2.s3.amazonaws.com/latest/install","chmod +x ./install","./install auto"]}'
```

Executar um script bash em um arquivo JSON

No exemplo a seguir, o conteúdo do script bash é armazenado em um arquivo JSON e o arquivo é incluído no comando usando o`--cli-input-jsonopção`.

Linux & macOS

```
aws ssm send-command \
    --document-name "AWS-RunShellScript" \
    --targets "Key=InstanceIds,Values=instance-id" \
    --cli-input-json file://installCodeDeployAgent.json
```

Windows

```
aws ssm send-command ^
    --document-name "AWS-RunShellScript" ^
    --targets "Key=InstanceIds,Values=instance-id" ^
```

```
    --cli-input-json file://installCodeDeployAgent.json
```

O conteúdo do arquivo `installCodeDeployAgent.json` mencionado é mostrado no exemplo a seguir.

```
{
    "Parameters": {
        "commands": [
            "#!/bin/bash",
            "yum -y update",
            "yum install -y ruby",
            "cd /home/ec2-user",
            "curl -O https://aws-codedeploy-us-east-2.s3.amazonaws.com/latest/install",
            "chmod +x ./install",
            "./install auto"
        ]
    }
}
```

Demonstração: Usar a AWS Tools for Windows PowerShell com o Run Command

Os exemplos a seguir mostram como usar o AWS Tools for Windows PowerShell para ver informações sobre comandos e parâmetros de comandos, como executar comandos e como visualizar o status desses comandos. Essa demonstração inclui um exemplo para cada um dos documentos do AWS Systems Manager predefinidos.

> **Important**
>
> Apenas administradores confiáveis devem ter permissão para usar os documentos pré-configurados do Systems Manager mostrados neste tópico. Os comandos ou scripts especificados em documentos do Systems Manager são executados com permissão administrativa em seus nós gerenciados. Se um usuário tiver permissão para executar qualquer um dos documentos do Systems Manager predefinidos (qualquer documento que comece com AWS), ele também terá acesso de administrador ao nó. Para todos os outros usuários, você deve criar documentos restritivos e compartilhá-los com usuários específicos.

Tópicos

- Definir configurações de sessão do AWS Tools for Windows PowerShell

- Listar todos os documentos disponíveis
- Executar comandos ou scripts do PowerShell
- Instalar uma aplicação usando o documento AWS-InstallApplication
- Instalar um módulo do PowerShell usando aAWS-InstallPowerShellModuleDocumento JSON
- Integrar um nó gerenciado a um domínio usando o documento JSON AWS-JoinDirectoryServiceDomain
- Enviar métricas do Windows ao Amazon CloudWatch Logs usando o documento AWS-ConfigureCloudWatch
- Atualizar o EC2Config usando aAWS-UpdateEC2Configdocument
- Ative ou desative a atualização automática do Windows usando oAWS-ConfigureWindowsUpdatedocument
- Gerenciar atualizações do Windows usando o Run Command

Definir configurações de sessão do AWS Tools for Windows PowerShell

Especificar suas credenciais

Abra as Tools for Windows PowerShell no computador local e execute o comando a seguir para especificar suas credenciais. Você deve ter permissões de administrador sobre os nós gerenciados que deseja configurar ou deve ter recebido a permissão apropriada no AWS Identity and Access Management (IAM). Para ter mais informações, consulte Configurar nós gerenciados para o AWS Systems Manager.

```
Set-AWSCredentials –AccessKey key-name –SecretKey key-name
```

Definir uma Região da AWS padrão

Execute o comando a seguir para definir a região da sua sessão do PowerShell. O exemplo usa região Leste dos EUA (Ohio) (us-east-2). O Run Command está disponível nas Regiões da AWS listadas em Systems Manager service endpoints no Referência geral da Amazon Web Services.

```
Set-DefaultAWSRegion `
    -Region us-east-2
```

Listar todos os documentos disponíveis

Esse comando lista todos os documentos disponíveis para a sua conta.

```
Get-SSMDocumentList
```

Executar comandos ou scripts do PowerShell

Por meio do Run Command e do documento `AWS-RunPowerShell`, você pode executar qualquer comando ou script em um nó gerenciado, como se estivesse conectado localmente. Você pode emitir comandos ou inserir um caminho para um script local para executar o comando.

> **Note**
>
> Para obter informações sobre a reinicialização de nós gerenciados ao usar o Run Command para chamar scripts, consulte Tratamento de reinicializações ao executar comandos.

Veja a descrição e os parâmetros disponíveis

```
Get-SSMDocumentDescription `
    -Name "AWS-RunPowerShellScript"
```

Veja mais informações sobre parâmetros

```
Get-SSMDocumentDescription `
    -Name "AWS-RunPowerShellScript" | Select -ExpandProperty Parameters
```

Enviar um comando usando o documento **`AWS-RunPowerShellScript`**

O comando a seguir mostra o conteúdo do diretório `"C:\Users"` e o conteúdo do diretório `"C:\"` em dois nós.

```
$runPSCommand = Send-SSMCommand `
    -InstanceIds @("instance-ID-1", "instance-ID-2") `
    -DocumentName "AWS-RunPowerShellScript" `
    -Comment "Demo AWS-RunPowerShellScript with two instances" `
    -Parameter @{'commands'=@('dir C:\Users', 'dir C:\')}
```

Obter detalhes da solicitação de comando

O comando a seguir usa o `CommandId` para obter o status da execução do comando em ambos os nós. Este exemplo usa o `CommandId` que foi retornado no comando anterior.

```
Get-SSMCommand `
    -CommandId $runPSCommand.CommandId
```

O status do comando neste exemplo pode ser Success, Pending ou InProgress.

Obter informações de comando por nó gerenciado

O comando a seguir usa o `CommandId` do comando anterior para obter o status da execução do comando por nó gerenciado.

```
Get-SSMCommandInvocation `
    -CommandId $runPSCommand.CommandId
```

Obter informações de comando com dados de resposta para um nó gerenciado específico

O comando a seguir retorna a saída da solicitação `Send-SSMCommand` original para um nó gerenciado específico.

```
Get-SSMCommandInvocation `
    -CommandId $runPSCommand.CommandId `
    -Details $true `
    -InstanceId instance-ID | Select -ExpandProperty CommandPlugins
```

Cancelar um comando

O comando a seguir cancela o `Send-SSMCommand` para o documento `AWS-RunPowerShellScript`.

```
$cancelCommand = Send-SSMCommand `
    -InstanceIds @("instance-ID-1","instance-ID-2") `
    -DocumentName "AWS-RunPowerShellScript" `
    -Comment "Demo AWS-RunPowerShellScript with two instances" `
    -Parameter @{'commands'='Start-Sleep -Seconds 120; dir C:\'}

Stop-SSMCommand -CommandId $cancelCommand.CommandId
```

Verificar o status do comando

O comando a seguir verifica o status do comando `Cancel`.

```
Get-SSMCommand `
```

```
    -CommandId $cancelCommand.CommandId
```

Instalar uma aplicação usando o documento **`AWS-InstallApplication`**

Usando o Run Command e o documento `AWS-InstallApplication`, é possível instalar, reparar ou desinstalar aplicações em nós gerenciados. O comando requer o caminho ou endereço para um MSI.

> **Note**
>
> Para obter informações sobre a reinicialização de nós gerenciados ao usar o Run Command para chamar scripts, consulte Tratamento de reinicializações ao executar comandos.

Veja a descrição e os parâmetros disponíveis

```
Get-SSMDocumentDescription `
    -Name "AWS-InstallApplication"
```

Veja mais informações sobre parâmetros

```
Get-SSMDocumentDescription `
    -Name "AWS-InstallApplication" | Select -ExpandProperty Parameters
```

Enviar um comando usando o documento **`AWS-InstallApplication`**

O comando a seguir instala uma versão do Python em seu nó gerenciado no modo autônomo e registra a saída em um arquivo de texto local na sua unidade `C:`.

```
$installAppCommand = Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-InstallApplication" `
    -Parameter @{'source'='https://www.python.org/ftp/python/2.7.9/python-2.7.9.msi';
 'parameters'='/norestart /quiet /log c:\pythoninstall.txt'}
```

Obter informações de comando por nó gerenciado

O comando a seguir usa o `CommandId` para obter o status da execução do comando.

```
Get-SSMCommandInvocation `
    -CommandId $installAppCommand.CommandId `
```

```
    -Details $true
```

Obter informações de comando com dados de resposta para um nó gerenciado específico

O comando a seguir retorna os resultados da instalação do Python.

```
Get-SSMCommandInvocation `
    -CommandId $installAppCommand.CommandId `
    -Details $true `
    -InstanceId instance-ID | Select -ExpandProperty CommandPlugins
```

Instalar um módulo do PowerShell usando a**AWS-InstallPowerShellModule**Documento JSON

Você pode usar o Run Command para instalar módulos do PowerShell em nós gerenciados. Para obter mais informações sobre os módulos PowerShell, consulte Módulos do Windows PowerShell.

Veja a descrição e os parâmetros disponíveis

```
Get-SSMDocumentDescription `
    -Name "AWS-InstallPowerShellModule"
```

Veja mais informações sobre parâmetros

```
Get-SSMDocumentDescription `
    -Name "AWS-InstallPowerShellModule" | Select -ExpandProperty Parameters
```

Instalar um módulo PowerShell

O comando a seguir faz download do arquivo EZOut.zip, instala o arquivo e depois executa um comando adicional para instalar o visualizador XPS. Por fim, a saída desse comando é carregada em um bucket do S3 chamado "amzn-s3-demo-bucket".

```
$installPSCommand = Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-InstallPowerShellModule" `
    -Parameter @{'source'='https://gallery.technet.microsoft.com/EZOut-33ae0fb7/
file/110351/1/EZOut.zip';'commands'=@('Add-WindowsFeature -name XPS-Viewer -restart')}
 `
    -OutputS3BucketName amzn-s3-demo-bucket
```

Obter informações de comando por nó gerenciado

O comando a seguir usa o `CommandId` para obter o status da execução do comando.

```
Get-SSMCommandInvocation `
    -CommandId $installPSCommand.CommandId `
    -Details $true
```

Obter informações de comando com dados de resposta para o nó gerenciado

O comando a seguir retorna a saída do `Send-SSMCommand` original para o `CommandId` específico.

```
Get-SSMCommandInvocation `
    -CommandId $installPSCommand.CommandId `
    -Details $true | Select -ExpandProperty CommandPlugins
```

Integrar um nó gerenciado a um domínio usando o documento JSON **`AWS-JoinDirectoryServiceDomain`**

Por meio do Run Command, você pode unir rapidamente um nó gerenciado a um domínio do AWS Directory Service. Antes de executar esse comando, você deverá criar um diretório. Recomendamos também que conheça mais sobre o AWS Directory Service. Para obter mais informações, consulte o Guia do Administrador do AWS Directory Service.

Você só pode unir um nó gerenciado a um domínio. Não é possível remover um nó de um domínio.

> **Note**
>
> Para obter informações sobre nós gerenciados ao usar o Run Command para chamar scripts, consulte Tratamento de reinicializações ao executar comandos.

Veja a descrição e os parâmetros disponíveis

```
Get-SSMDocumentDescription `
    -Name "AWS-JoinDirectoryServiceDomain"
```

Veja mais informações sobre parâmetros

```
Get-SSMDocumentDescription `
    -Name "AWS-JoinDirectoryServiceDomain" | Select -ExpandProperty Parameters
```

Integrar um nó gerenciado a um domínio

O comando a seguir integra um nó gerenciado a um domínio AWS Directory Service determinado e carrega qualquer saída gerada no exemplo do bucket do Amazon Simple Storage Service (Amazon S3).

```
$domainJoinCommand = Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-JoinDirectoryServiceDomain" `
    -Parameter @{'directoryId'='d-example01'; 'directoryName'='ssm.example.com';
 'dnsIpAddresses'=@('192.168.10.195', '192.168.20.97')} `
    -OutputS3BucketName amzn-s3-demo-bucket
```

Obter informações de comando por nó gerenciado

O comando a seguir usa o `CommandId` para obter o status da execução do comando.

```
Get-SSMCommandInvocation `
    -CommandId $domainJoinCommand.CommandId `
    -Details $true
```

Obter informações de comando com dados de resposta para o nó gerenciado

Este comando retorna a saída do `Send-SSMCommand` original para o `CommandId` específico.

```
Get-SSMCommandInvocation `
    -CommandId $domainJoinCommand.CommandId `
    -Details $true | Select -ExpandProperty CommandPlugins
```

Enviar métricas do Windows ao Amazon CloudWatch Logs usando o documento **`AWS-ConfigureCloudWatch`**

Você pode enviar mensagens do Windows Server nos logs do aplicativo, do sistema e do Rastreamento de Eventos para Windows (ETW) para o Amazon CloudWatch Logs. Quando você permite o registro pela primeira vez, o Systems Manager envia todos os logs gerados no prazo de um (1) minuto a partir do momento em que você inicia o upload de logs para a aplicação, o sistema, a segurança e os logs do ETW. Logs ocorridos antes dessa hora não são incluídos. Se você desativar o registro em log e depois reabilitá-lo, o Systems Manager enviará logs do ponto em que ele parou. Para todos os arquivos de log personalizados e logs do IIS (Serviços de Informações da Internet), o Systems Manager lê os arquivos de log desde o início. Além disso, o Systems Manager também pode enviar dados do contador de performance para o CloudWatch Logs.

Se você habilitou anteriormente a integração do CloudWatch no EC2Config, as configurações do Systems Manager substituirão todas as configurações armazenadas localmente em seu nó gerenciado no arquivo `C:\Program Files\Amazon\EC2ConfigService\Settings\AWS.EC2.Windows.CloudWatch.json`. Para obter mais informações sobre como usar o EC2Config para gerenciar o contador e os logs de performance em um único nó gerenciado, consulte Coletar métricas e logs de instâncias do Amazon EC2 em servidores on-premises com o agente do CloudWatch no Guia do usuário do Amazon CloudWatch.

Veja a descrição e os parâmetros disponíveis

```
Get-SSMDocumentDescription `
    -Name "AWS-ConfigureCloudWatch"
```

Veja mais informações sobre parâmetros

```
Get-SSMDocumentDescription `
    -Name "AWS-ConfigureCloudWatch" | Select -ExpandProperty Parameters
```

Enviar logs de aplicativo ao CloudWatch

O comando a seguir configura o nó gerenciado e move logs de aplicações do Windows para o CloudWatch.

```
$cloudWatchCommand = Send-SSMCommand `
    -InstanceID instance-ID `
    -DocumentName "AWS-ConfigureCloudWatch" `
    -Parameter @{'properties'='{"engineConfiguration": {"PollInterval":"00:00:15",
 "Components":[{"Id":"ApplicationEventLog",
 "FullName":"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWa
 "Parameters":{"LogName":"Application", "Levels":"7"}},{"Id":"CloudWatch",
 "FullName":"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
 "Parameters":{"Region":"region", "LogGroup":"my-log-group", "LogStream":"instance-
id"}}], "Flows":{"Flows":["ApplicationEventLog,CloudWatch"]}}}'}
```

Obter informações de comando por nó gerenciado

O comando a seguir usa o `CommandId` para obter o status da execução do comando.

```
Get-SSMCommandInvocation `
    -CommandId $cloudWatchCommand.CommandId `
```

```
    -Details $true
```

Obter informações de comando com dados de resposta para um nó gerenciado específico

O comando a seguir retorna os resultados da configuração do Amazon CloudWatch.

```
Get-SSMCommandInvocation `
    -CommandId $cloudWatchCommand.CommandId `
    -Details $true `
    -InstanceId instance-ID | Select -ExpandProperty CommandPlugins
```

Enviar contadores de performance ao CloudWatch usando o documento **AWS-ConfigureCloudWatch**

O comando de demonstração a seguir faz upload de contadores de performance para o CloudWatch. Para obter mais informações, consulte o Guia do usuário do Amazon CloudWatch.

```
$cloudWatchMetricsCommand = Send-SSMCommand `
    -InstanceID instance-ID `
    -DocumentName "AWS-ConfigureCloudWatch" `
    -Parameter @{'properties'='{"engineConfiguration": {"PollInterval":"00:00:15",
 "Components":[{"Id":"PerformanceCounter",
 "FullName":"AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInputComp
 "Parameters":{"CategoryName":"Memory", "CounterName":"Available
 MBytes", "InstanceName":"", "MetricName":"AvailableMemory",
 "Unit":"Megabytes","DimensionName":"", "DimensionValue":""}},{"Id":"CloudWatch",
 "FullName":"AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.Cl
 "Parameters":{"AccessKey":"", "SecretKey":"","Region":"region", "NameSpace":"Windows-
Default"}}], "Flows":{"Flows":["PerformanceCounter,CloudWatch"]}}}'}
```

Atualizar o EC2Config usando a**AWS-UpdateEC2Config**document

Usando o Run Command e o documento `AWS-EC2ConfigUpdate`, é possível atualizar o serviço EC2Config em execução em nós gerenciados do Windows Server. Esse comando pode atualizar o serviço EC2Config para a versão mais recente ou uma versão que você especificar.

Veja a descrição e os parâmetros disponíveis

```
Get-SSMDocumentDescription `
    -Name "AWS-UpdateEC2Config"
```

Veja mais informações sobre parâmetros

```
Get-SSMDocumentDescription `
    -Name "AWS-UpdateEC2Config" | Select -ExpandProperty Parameters
```

Atualizar EC2Config para a versão mais recente

```
$ec2ConfigCommand = Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-UpdateEC2Config"
```

Obter informações de comando com dados de resposta para o nó gerenciado

Esse comando retorna a saída do comando especificado do `Send-SSMCommand` anterior.

```
Get-SSMCommandInvocation `
    -CommandId $ec2ConfigCommand.CommandId `
    -Details $true `
    -InstanceId instance-ID | Select -ExpandProperty CommandPlugins
```

Atualizar EC2Config para uma versão específica

O comando a seguir faz o downgrade do EC2Config para uma versão anterior.

```
Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-UpdateEC2Config" `
    -Parameter @{'version'='4.9.3519'; 'allowDowngrade'='true'}
```

Ative ou desative a atualização automática do Windows usando o**`AWS-ConfigureWindowsUpdate`**document

Usando o Run Command e o documento `AWS-ConfigureWindowsUpdate`, é possível ativar ou desativar as atualizações automáticas do Windows em seus nós gerenciados do Windows Server. Esse comando configura o agente de atualização do Windows para baixar e instalar atualizações do Windows na data e hora no dia e hora que você especificar. Se uma atualização exigir uma reinicialização, o nó gerenciado reiniciará automaticamente 15 minutos após a instalação das atualizações. Com esse comando, você também pode configurar o Windows Update para verificar atualizações, mas não pode instalá-las. O documento `AWS-ConfigureWindowsUpdate` é compatível com o Windows Server 2008, 2008 R2, 2012, 2012 R2 e 2016.

Veja a descrição e os parâmetros disponíveis

```
Get-SSMDocumentDescription `
    -Name "AWS-ConfigureWindowsUpdate"
```

Veja mais informações sobre parâmetros

```
Get-SSMDocumentDescription `
    -Name "AWS-ConfigureWindowsUpdate" | Select -ExpandProperty Parameters
```

Ativar a atualização automática do Windows

O comando a seguir configura o Windows Update para baixar e instalar atualizações diariamente às 22h.

```
$configureWindowsUpdateCommand = Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-ConfigureWindowsUpdate" `
    -Parameters @{'updateLevel'='InstallUpdatesAutomatically';
 'scheduledInstallDay'='Daily'; 'scheduledInstallTime'='22:00'}
```

Visualizar o status do comando para permitir atualizações automáticas do Windows

O comando a seguir usa o `CommandId` para obter o status da execução do comando para ativar a Atualização Automática do Windows.

```
Get-SSMCommandInvocation `
    -Details $true `
    -CommandId $configureWindowsUpdateCommand.CommandId | Select -ExpandProperty
 CommandPlugins
```

Desativar a atualização automática do Windows

O seguinte comando reduz o nível de notificação do Windows Update para que o sistema verifique se há atualizações, mas não atualiza automaticamente o nó gerenciado.

```
$configureWindowsUpdateCommand = Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-ConfigureWindowsUpdate" `
    -Parameters @{'updateLevel'='NeverCheckForUpdates'}
```

Visualizar o status do comando para desativar a atualização automática do Windows

O comando a seguir usa o `CommandId` para obter o status da execução do comando para ativar a Atualização Automática do Windows.

```
Get-SSMCommandInvocation `
    -Details $true `
    -CommandId $configureWindowsUpdateCommand.CommandId | Select -ExpandProperty
 CommandPlugins
```

Gerenciar atualizações do Windows usando o Run Command

Usando o Run Command e do documento `AWS-InstallWindowsUpdates`, você pode gerenciar atualizações dos nós gerenciados do Windows Server. Esse comando verifica ou instala atualizações ausentes em nós gerenciados e, opcionalmente, reinicializa após a instalação. Você também pode especificar as classificações e os níveis de severidade adequados para as atualizações a serem instaladas no ambiente.

> **Note**
>
> Para obter informações sobre a reinicialização de nós gerenciados ao usar o Run Command para chamar scripts, consulte Tratamento de reinicializações ao executar comandos.

Os exemplos a seguir demonstram como realizar as tarefas de gerenciamento do Windows Update especificadas.

Procurar todas as atualizações do Windows ausentes

```
Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-InstallWindowsUpdates" `
    -Parameters @{'Action'='Scan'}
```

Instalar atualizações específicas do Windows

```
Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-InstallWindowsUpdates" `
```

```
    -Parameters @{'Action'='Install';'IncludeKbs'='kb-ID-1,kb-ID-2,kb-
ID-3';'AllowReboot'='True'}
```

Instalar atualizações importantes do Windows ausentes

```
Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-InstallWindowsUpdates" `
    -Parameters @{'Action'='Install';'SeverityLevels'='Important';'AllowReboot'='True'}
```

Instalar atualizações do Windows ausentes com exclusões específicas

```
Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-InstallWindowsUpdates" `
    -Parameters @{'Action'='Install';'ExcludeKbs'='kb-ID-1,kb-
ID-2';'AllowReboot'='True'}
```

Solução de problemas do Run Commando do Systems Manager

O Run Command, uma ferramenta do AWS Systems Manager, fornece detalhes de status com cada execução do comando. Para obter mais informações sobre os detalhes dos status de comando, consulte Noções básicas sobre status de comando. Você também pode usar as informações neste tópico para ajudar a solucionar problemas com a Run Command.

Tópicos

- Alguns dos meus nós gerenciados estão ausentes
- Uma etapa no meu script falhou, mas o status geral é "bem-sucedido".
- O SSM Agent não está sendo executado corretamente

Alguns dos meus nós gerenciados estão ausentes

Na página Run a command(Executar um comando), depois de escolher um documento do SSM para executar e selecionar Manually selecting instances (Selecionar instâncias manualmente) na seção Targets (Destinos), é exibida uma lista de nós gerenciados que você pode escolher para a execução do comando.

Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

Depois de criar, ativar, reinicializar ou reiniciar um nó gerenciado, instale o Run Command em um nó ou anexe um perfil da instância do IAM do AWS Identity and Access Management a um nó. Esse nó poderá levar alguns minutos para aparecer na lista.

Uma etapa no meu script falhou, mas o status geral é "bem-sucedido".

Com o Run Command, é possível definir como seus scripts manipulam os códigos de saída. Por padrão, o código de saída do último comando executado em um script é relatado como o código de saída de todo o script. No entanto, você pode incluir uma instrução condicional para sair do script se algum comando antes do final falhar. Para obter informações e exemplos, consulte Especifique códigos de saída nos comandos.

O SSM Agent não está sendo executado corretamente

Se você tiver problemas ao executar comandos usando o Run Command, isso significa que pode haver um problema com o SSM Agent. Para obter informações sobre como investigar problemas com o SSM Agent, consulte Solução de problemas do SSM Agent.

AWS Systems Manager Session Manager

O Session Manager é uma ferramenta do AWS Systems Manager totalmente gerenciada. Com o Session Manager, é possível gerenciar as instâncias, os dispositivos de borda, os servidores on-premises e as máquinas virtuais (VMs) do Amazon Elastic Compute Cloud (Amazon EC2). Use um shell interativo baseado em navegador acionado com um clique ou o AWS Command Line Interface (AWS CLI). O Session Manager fornece gerenciamento de nós seguro sem a necessidade de abrir portas de entrada, manter bastion hosts ou gerenciar chaves SSH. O Session Manager também facilita a conformidade com políticas corporativas que exigem acesso controlado aos nós gerenciados, práticas rígidas de segurança e logs com detalhes do acesso aos nós, sem deixar de fornecer aos usuários finais acesso com um clique aos nós gerenciados em várias plataformas. Para começar a usar o Session Manager, abra o Systems Manager console (Console do gerenciador de sistemas). No painel de navegação, escolha Session Manager.

Como o Session Manager beneficia minha organização?

Session Manager oferece estes benefícios:

- Controle de acesso centralizado aos nós gerenciados usando as políticas do IAM

  Os administradores têm um único lugar para conceder e revogar o acesso a nós gerenciados. Usando somente políticas do AWS Identity and Access Management (IAM), você pode controlar

quais usuários individuais ou grupos na sua organização podem usar o Session Manager e quais nós gerenciados eles podem acessar.

- Sem a necessidade de abrir portas de entrada e de gerenciar bastion hosts ou chaves SSH

  Deixar portas SSH de entrada e portas do PowerShell remotas abertas em seus nós gerenciados aumenta muito o risco de entidades executarem comandos não autorizados ou mal-intencionados nesses nós. O Session Manager ajuda a melhorar o procedimento de segurança, permitindo que você feche essas portas de entrada, eliminando o gerenciamento de chaves e certificados SSH, bastion hosts e caixas de diálogo.

- Acesso com um clique aos nós gerenciados do console e da CLI

  Usando o console do AWS Systems Manager ou do Amazon EC2, é possível iniciar uma sessão com um único clique. Usando a AWS CLI, você também pode iniciar uma sessão que executa um único comando ou uma sequência de comandos. Como as permissões para os nós gerenciados são fornecidas por meio de políticas do IAM em vez de chaves SSH ou outros mecanismos, o tempo de conexão é significativamente reduzido.

- Estabelecer conexão com instâncias do Amazon EC2 e nós gerenciados que não são do EC2 em ambientes híbridos e multinuvem

  É possível estabelecer conexão com instâncias do Amazon Elastic Compute Cloud (Amazon EC2) e nós que não são do EC2 em seu ambiente híbrido e multinuvem.

  Para estabelecer conexão com nós que não pertençam ao EC2 usando o Session Manager, primeiramente é necessário ativar o nível de instâncias avançadas. Há uma cobrança para o uso do nível de instâncias avançadas. No entanto, não há custo adicional para se conectar às instâncias do EC2 usando o Session Manager. Para mais informações, consulte Configurar níveis de instâncias.

- Encaminhamento de portas

  Redirecione todas as portas dentro do nó gerenciado para uma porta local em um cliente. Depois disso, conecte-se à porta local e acesse a aplicação do servidor que está sendo executado dentro do nó.

- Suporte entre plataformas para Windows, Linux e macOS

  O Session Manager fornece suporte para Windows, Linux e macOS em uma única ferramenta. Por exemplo, você não precisa usar um cliente SSH para nós gerenciados do Linux e do macOS ou uma conexão RDP para nós gerenciados do Windows Server.

- Registrar a atividade de sessão de registro em log

  Para satisfazer os requisitos de segurança ou operacionais em sua organização, pode ser necessário fornecer um registro das conexões feitas para seus nós gerenciados e os comandos que foram executados nelas. Você também pode receber notificações quando um usuário na sua organização começar ou terminar a atividade da sessão.

  Os recursos de registro em log são fornecidos por meio de integração com os seguintes Serviços da AWS:

  - AWS CloudTrail – o AWS CloudTrail captura informações sobre chamadas de API do Session Manager feitas na conta da Conta da AWS e grava-as em arquivos de log armazenados em um bucket do Amazon Simple Storage Service (Amazon S3) especificado. Um bucket é usado em todos os logs do CloudTrail para sua conta. Para obter mais informações, consulte Registrar em log chamadas de API do AWS Systems Manager com o AWS CloudTrail.
  - Amazon Simple Storage Service – Você pode optar por armazenar os dados de log da sessão em um bucket do Amazon S3 de sua escolha para fins de depuração e solução de problemas. Os dados de log podem ser enviados ao seu bucket do Amazon S3 com ou sem criptografia usando sua AWS KMS key. Para obter mais informações, consulte Registrar dados da sessão em log usando o Amazon S3 (console).
  - Amazon CloudWatch Logs: o CloudWatch Logs permite monitorar, armazenar e acessar os arquivos de log de vários Serviços da AWS. Você pode enviar dados de log da sessão a um grupo de logs do CloudWatch Logs para fins de depuração e solução de problemas. Os dados de log podem ser enviados ao seu grupo de logs com ou sem criptografia do AWS KMS usando sua chave do KMS. Para obter mais informações, consulte Registrar dados da sessão em log usando o Amazon CloudWatch Logs (console).
  - Amazon EventBridge e Amazon Simple Notification Service – O EventBridge permite configurar regras para detectar quando ocorrem alterações nos recursos da AWS que você especificar. Você pode criar uma regra para detectar quando um usuário na sua organização inicia ou interrompe uma sessão e, em seguida, receber uma notificação por meio do Amazon SNS (por exemplo, uma mensagem texto ou e-mail) sobre o evento. Você também pode configurar um evento do CloudWatch para iniciar outras respostas. Para obter mais informações, consulte Monitorar as atividades da sessão usando o Amazon EventBridge (console).

  > **Note**
  >
  > O registro em log não está disponível para sessões do Session Manager que se conectam por meio de encaminhamento de portas ou SSH. Isso ocorre porque o SSH criptografa

todos os dados da sessão e o Session Manager serve apenas como um túnel para conexões SSH.

Quem deve usar o Session Manager?

- Qualquer cliente da AWS que quiser melhorar sua postura de segurança, reduzir despesas operacionais centralizando o controle de acesso nos nós gerenciados e reduzir o acesso à entrada dos nós.
- Especialistas em segurança da informação que quiserem monitorar e acompanhar o acesso e a atividade de nós gerenciados, fechar portas de entrada em nós gerenciados ou permitir conexões a esses nós sem um endereço IP público.
- Administradores que desejam conceder e revogar acesso de um único lugar e querem fornecer uma solução aos usuários para nós gerenciados do Linux, macOS e Windows Server.
- Os usuários que desejam se conectar a um nó gerenciado com apenas um clique no navegador ou na AWS CLI, sem a necessidade de fornecer chaves SSH.

Quais são os principais recursos do Session Manager?

- Suporte para nós gerenciados do Windows Server, do Linux e do macOS

  O Session Manager permite que você estabeleça conexões seguras com as instâncias, os dispositivos de borda, os servidores on-premises e as máquinas virtuais (VMs) do Amazon Elastic Compute Cloud (EC2). Para obter uma lista dos tipos de sistema operacional compatíveis, consulte Configurar o Session Manager.

  > **Note**
  >
  > O suporte do Session Manager para máquinas on-premises é fornecido somente para o nível de instâncias avançadas. Para mais informações, consulte Ativar o nível de instâncias avançadas.

- Acesso do Console, CLI e SDK aos recursos do Session Manager

  Você pode trabalhar com Session Manager das seguintes formas:

O console do AWS Systems Manager inclui acesso a todos os recursos do Session Manager para ambos os administradores e usuários finais. Você pode executar qualquer tarefa relacionada às suas sessões usando o console do Systems Manager.

O console do Amazon EC2 fornece o recurso de os usuários finais se conectarem às instâncias do EC2 para as quais receberam permissões de sessão.

A AWS CLI inclui acesso a recursos do Session Manager para usuários finais. Você pode iniciar uma sessão, visualizar uma lista de sessões e encerrar permanentemente uma sessão usando a AWS CLI.

> **Note**
>
> Para usar a AWS CLI para executar comandos da sessão, você deve estar usando a versão 1.16.12 da CLI (ou posterior) e deve ter instalado o plugin do Session Manager em sua máquina local. Para mais informações, consulte Instalar o plug-in do Session Manager para a AWS CLI. Para visualizar o plug-in no GitHub, consulte session-manager-plugin.

- Controle de acesso do IAM

  Com o uso de políticas do IAM, você pode controlar quais membros da organização podem iniciar sessões para nós gerenciados e quais nós eles podem acessar. Você também pode fornecer acesso temporário aos nós gerenciados. Por exemplo, você pode fornecer um acesso a um engenheiro de plantão (ou um grupo de engenheiros de plantão) aos servidores de produção somente durante todo o período de rotação.

- Suporte a log

  O Session Manager fornece opções de histórico de sessões de log na sua Conta da AWS por meio da integração com uma série de outros Serviços da AWS. Para obter mais informações, consulte Registrar a atividade de sessão de registro em log e Habilitar e desabilitar o registro em log de sessão.

- Perfis de shell configuráveis

  O Session Manager fornece opções para configurar preferências dentro das sessões. Esses perfis personalizáveis permitem que você defina opções preferenciais, como preferências de shell, variáveis de ambiente, diretórios de trabalho e execução de vários comandos quando uma sessão é iniciada.

- Suporte para criptografia de dados de chaves do cliente

  É possível configurar o Session Manager para criptografar os logs de dados da sessão que você envia para um bucket do Amazon Simple Storage Service (Amazon S3) ou transmite para um grupo de logs do Amazon CloudWatch Logs. Você também pode configurar o Session Manager para criptografar dados transmitidos entre as máquinas clientes e os nós gerenciados durante as sessões. Para obter mais informações, consulte Habilitar e desabilitar o registro em log de sessão e Configurar preferências da sessão.
- Suporte ao AWS PrivateLink para nós gerenciados sem endereços IP públicos

  Você também pode configurar endpoints da VPC para o Systems Manager usando o AWS PrivateLink para proteger ainda mais suas sessões. O AWS PrivateLink limita todo o tráfego de rede entre os nós gerenciados, o Systems Manager, o Amazon EC2 e a rede da Amazon. Para obter mais informações, consulte Melhorar a segurança das instâncias do EC2 usando endpoints da VPC para o Systems Manager.
- Encapsulamento

  Em uma sessão, use um documento do AWS Systems Manager (SSM) do tipo Session para tráfego em túnel, como http ou um protocolo personalizado, entre uma porta local em uma máquina cliente e uma porta remota em um nó gerenciado.
- Comandos interativos

  Crie um documento SSM do tipo sessão que use uma sessão para executar interativamente um único comando, o que fornece uma maneira de gerenciar o que os usuários podem fazer em um nó gerenciado.

O que é uma sessão?

Uma sessão é uma conexão feita a um nó gerenciado usando o Session Manager. As sessões são baseadas em um canal de comunicação bidirecional seguro entre o cliente (você) e o nó gerenciado remoto que transmite entradas e saídas para comandos. O tráfego entre um cliente e um nó gerenciado é criptografado usando TLS 1.2, e as solicitações para criar a conexão são assinadas usando Sigv4. Essa comunicação bidirecional permite acesso interativo bash e PowerShell aos nós gerenciados. Também é possível usar uma chave do AWS Key Management Service (AWS KMS) para criptografar dados além da criptografia TLS padrão.

Por exemplo, digamos que John é um engenheiro de plantão do departamento de TI. Ele recebe a notificação de um problema que requer a conexão remota a um nó gerenciado, como uma falha que

requer a solução de problemas ou uma diretiva para alterar uma opção de configuração simples em um nó. Usando o console do AWS Systems Manager, o Amazon EC2 ou a AWS CLI, John inicia uma sessão conectando-o ao nó gerenciado, executa os comandos em um nó necessários para concluir a tarefa e encerra a sessão.

Quando John envia o primeiro comando para iniciar a sessão, o serviço Session Manager autentica o seu ID, verifica as permissões concedidas a ele por uma política do IAM, verifica as definições de configuração (como a verificação de limites permitidos para as sessões) e envia uma mensagem para SSM Agent para abrir a conexão bidirecional. Depois que a conexão é estabelecida e John digita o próximo comando, a saída do comando SSM Agent é carregada para esse canal de comunicação e enviada de volta para sua máquina local.

Tópicos

- Configurar o Session Manager
- Trabalhar com o Session Manager
- Registrar a atividade de sessão de registro em log
- Habilitar e desabilitar o registro em log de sessão
- Esquema do documento de sessão
- Solução de problemas do Session Manager

Configurar o Session Manager

Antes de usar AWS Systems Manager Session Manager para se conectar aos nós gerenciados na sua conta, conclua as etapas nos tópicos a seguir.

Tópicos

- Etapa 1: Concluir os pré-requisitos do Session Manager
- Etapa 2: verificar ou adicionar permissões de instância para o Session Manager
- Etapa 3: controlar o acesso da sessão pelos nós gerenciados
- Etapa 4: Configurar preferências de sessão
- Etapa 5: (Opcional) Restringir o acesso a comandos em uma sessão
- Etapa 6: (Opcional) Usar o AWS PrivateLink para configurar um endpoint da VPC para o Session Manager
- Etapa 7: (Opcional) Ativar ou desativar permissões administrativas da conta ssm-user.

- Etapa 8: (Opcional) Permitir e controlar permissões de conexões de SSH por meio do Session Manager

Etapa 1: Concluir os pré-requisitos do Session Manager

Antes de usar o Session Manager, verifique se seu ambiente atende aos requisitos a seguir.

Pré-requisitos da Session Manager

Requisito	Descrição
Sistemas operacionais compatíveis	O Session Manager é compatível com a conexão às instâncias do Amazon Elastic Compute Cloud (Amazon EC2), além de máquinas que não são do EC2, em seu ambiente híbrido e multinuvem que usa o nível de instâncias avançadas. O Session Manager oferece suporte às seguintes versões de sistemas operacionais: **Note** O Session Manager é compatível com as instâncias do EC2, com os dispositivos de borda, com os servidores on-premises e com as máquinas virtuais (VMs) no ambiente híbrido e multinuvem que usa o nível de instâncias avançadas. Para obter mais informações sobre instâncias avançadas, consulte Configurar níveis de instâncias. Linux e macOS O Session Manager é compatível com todas as versões do Linux e do macOS que forem

Requisito	Descrição
	compatíveis com o AWS Systems Manager. Para mais informações, consulte Sistemas operacionais e tipos de máquinas compatíveis. Windows O Session Manager é compatível com Windows Server 2012 até Windows Server 2022. **Note** O Microsoft Windows Server 2016 Nano não é compatível.

Requisito	Descrição
SSM Agent	No mínimo a versão 2.3.68.0 do AWS Systems Manager SSM Agent ou posterior deve ser instalada nos nós gerenciados aos quais você quer se conectar por meio de sessões. Para usar a opção de criptografar dados da sessão usando uma chave criada no AWS Key Management Service (AWS KMS), a versão 2.3.539.0 ou posterior do SSM Agent deve ser instalada em seu nó gerenciado. Para usar perfis de shell em uma sessão, a versão 3.0.161.0 ou posterior do SSM Agent deve ser instalada em seu nó gerenciado. Para iniciar um encaminhamento da porta do Session Manager ou uma sessão SSH, a versão 3.0.222.0 ou posterior do SSM Agent deve ser instalada em seu nó gerenciado. Para transmitir dados da sessão usando o Amazon CloudWatch Logs, a versão 3.0.284.0 ou posterior do SSM Agent deve ser instalada em seu nó gerenciado. Para obter informações sobre como determina r o número da versão em execução em uma instância, consulte Verificar o número de versão do SSM Agent. Para obter informações sobre como instalar manualmente ou atualizar automaticamente o SSM Agent, consulte Trabalhar com o SSM Agent. Sobre a conta ssm-user A partir da versão 2.3.50.0 do SSM Agent, o agente cria uma conta de usuário em seu

Requisito	Descrição
	nó gerenciado, com permissões de administrador ou de raiz, chamada de `ssm-user`. (Em versões anteriores a 2.3.612.0, a conta é criada quando o SSM Agent inicia ou reinicia. Na versão 2.3.612.0 e posteriores, o `ssm-user` é criado na primeira vez que uma sessão for iniciada em um nó gerenciado.) As sessões são executadas usando as credenciais administrativas da conta de usuário. Para obter mais informações sobre como restringir o controle administrativo para esta conta, consulte Desativar ou ativar permissões administrativas da conta ssm-user. ssm-user em controladores de domínio do Windows Server Começando com o SSM Agent versão 2.3.612.0, a conta do `ssm-user` não é criada automaticamente em nós usados como controladores de domínio do Windows Server. Para usar o Session Manager em uma máquina Windows Server usada como controlador de domínio, crie a conta `ssm-user` manualmente, caso ela ainda não esteja presente, e atribua permissões de Administrador do Domínio ao usuário. No Windows Server, o SSM Agent define uma nova senha para a conta `ssm-user` sempre que uma sessão é iniciada e, portanto, você não precisa especificar uma senha ao criar a conta.

Requisito	Descrição
Conectividade com endpoints	Os nós gerenciados aos quais você se conecta devem permitir o tráfego de saída HTTPS (porta 443) para os seguintes endpoints: • ec2messages.*region*.amazonaws.com • ssm.*region*.amazonaws.com • ssmmessages.*region*.amazonaws.com Para obter mais informações, consulte os tópicos a seguir. • Referência: ec2messages, ssmmessages e outras operações da API • Como criar endpoints da VPC para poder usar o Systems Manager para gerenciar instâncias do EC2 privadas sem acesso à Internet? no Centro de Conhecimento da AWS re:Post. Como alternativa, você pode se conectar aos endpoints necessários usando endpoints da interface. Para obter mais informações, consulte Etapa 6: (Opcional) Usar o AWS PrivateLink para configurar um endpoint da VPC para o Session Manager.

Requisito	Descrição
AWS CLI	(Opcional) Se você usar a AWS Command Line Interface (AWS CLI) para iniciar as sessões (em vez de usar o console da AWS Systems Manager ou o console do Amazon EC2), a versão 1.16.12 ou posterior da CLI deve estar instalada em sua máquina local. Você pode chamar o `aws --version` para verificar a versão. Se você precisar instalar ou atualizar a CLI, consulte Instalação do AWS Command Line Interface no Guia do usuário do AWS Command Line Interface. **Important** Uma versão atualizada do SSM Agent é lançada sempre que novas ferrament as são adicionadas ao Systems Manager ou sempre que atualizações são feitas nas ferramentas existentes. Deixar de usar a versão mais recente do agente pode impedir que seu nó gerenciado use várias ferrament as do Systems Manager. Por isso, recomendamos automatizar o processo de manter o SSM Agent atualizado em suas máquinas. Para mais informaçõ es, consulte Automatizar atualizações do SSM Agent. Inscreva-se na página Notas de versão do SSM Agent no GitHub para receber notificações sobre atualizações do SSM Agent.

Requisito	Descrição
	Além disso, para usar a CLI para gerenciar seus nós com o Session Manager, você deve primeiro instalar o plugin do Session Manager em sua máquina local. Para mais informações, consulte Instalar o plug-in do Session Manager para a AWS CLI.
Ative o nível de instâncias avançadas (ambientes híbridos e multinuvem)	Para conectar-se a máquinas que não são do EC2 usando o Session Manager, você deve ativar o nível de instâncias avançadas na Conta da AWS e Região da AWS onde você cria ativações híbridas para registrar máquinas que não são do EC2 como nós gerenciados. Há uma cobrança para o uso do nível de instâncias avançadas. Para obter mais informações sobre o nível de instância s avançadas, consulte Configurar níveis de instâncias.

Requisito	Descrição
Verifique as permissões de perfil de serviço do IAM (ambientes híbridos e multinuvem)	Os nós ativados para ambientes híbridos usam o perfil de serviço do AWS Identity and Access Management (IAM) especificado na ativação híbrida para se comunicar com as operações da API do Systems Manager. Esse perfil de serviço deve conter as permissões necessárias para se conectar às suas máquinas híbridas e multinuvem usando o Session Manager. Se sua função de serviço contém a política gerenciada `AmazonSSMManagedInstanceCore` da AWS, as permissões necessárias para o Session Manager já estão fornecidas. Se você achar que a função de serviço não contém as permissões necessárias, você deverá cancelar o registro da instância gerenciada e registrá-la com uma nova ativação híbrida que usa uma função de serviço do IAM com as permissões necessárias. Para obter mais informações sobre como cancelar o registro de instâncias gerenciadas, consulte Cancelar o registro de nós gerenciados em um ambiente híbrido e multinuvem. Para obter mais informações sobre como criar políticas do IAM com permissões do Session Manager, consulte Etapa 2: verificar ou adicionar permissões de instância para o Session Manager.

Etapa 2: verificar ou adicionar permissões de instância para o Session Manager

Por padrão, o AWS Systems Manager não tem permissão para executar ações em suas instâncias. É possível fornecer permissões de instância no nível da conta usando um perfil do AWS Identity and Access Management (IAM) ou no nível da instância usando um perfil de instância. Se o seu

caso de uso permitir, recomendamos conceder acesso no nível da conta usando a configuração de gerenciamento de host padrão. Se já definiu a configuração de gerenciamento do host padrão para sua conta usando a política `AmazonSSMManagedEC2InstanceDefaultPolicy`, você pode avançar para a próxima etapa. Para obter mais informações sobre a configuração de gerenciamento do host padrão, consulte Gerenciar instâncias do EC2 automaticamente com a Configuração de gerenciamento de hosts padrão.

Se preferir, você pode usar perfis de instância para fornecer as permissões necessárias às suas instâncias. Um perfil da instância passa uma função do IAM para uma instância do Amazon EC2. Você pode anexar um perfil da instância do IAM a uma instância do Amazon EC2 ao executá-la ou a uma instância executada anteriormente. Para obter mais informações, consulte Usar os perfis da instância.

Para servidores on-premises ou máquinas virtuais (VMs), as permissões são fornecidas pela função de serviço do IAM associada à ativação híbrida usada para registrar seus servidores e VMs on-premises com o Systems Manager. Servidores on-premises e VMs locais não usam perfis da instância.

Se você já usou outras ferramentas do Systems Manager, como o Run Command ou o Parameter Store, um perfil da instância com as permissões básicas necessárias para o Session Manager poderá já estar anexado às suas instâncias do Amazon EC2. Se um perfil da instância que contém a política gerenciada `AmazonSSMManagedInstanceCore` da AWS já estiver anexado às suas instâncias, as permissões para Session Manager já estarão fornecidas. Isso também é verdade se a função de serviço do IAM usada na ativação híbrida contiver a política gerenciada `AmazonSSMManagedInstanceCore`.

No entanto, em alguns casos, pode ser necessário modificar as permissões anexadas ao seu perfil de instância. Por exemplo, você quer fornecer um conjunto mais restrito de permissões de instância, você criou uma política personalizada para seu perfil da instância ou você quer usar as opções de criptografia do Amazon Simple Storage Service (Amazon S3) ou do AWS Key Management Service (AWS KMS) para proteger os dados da sessão. Nesses casos, para permitir que as ações do Session Manager sejam executadas em suas instâncias, proceda de uma das seguintes maneiras:

- Incorpore permissões para as ações do Session Manager em uma função do IAM personalizada

  Para adicionar permissões para ações do Session Manager a uma função do IAM existente que não dependa da política padrão `AmazonSSMManagedInstanceCore` fornecida pela AWS, siga as etapas em Adicionar permissões do Session Manager a um perfil do IAM existente.
- Crie uma função do IAM personalizada apenas com permissões do Session Manager

Para criar uma função do IAM que contenha permissões somente para ações do Session Manager, siga as etapas em Crie uma função do IAM personalizada para o Session Manager.

- Crie e use uma nova função do IAM com permissões para todas as ações do Systems Manager

  Para criar um perfil do IAM para as instâncias gerenciadas do Systems Manager que usam uma política padrão fornecida pela AWS para conceder todas as permissões do Systems Manager, siga as etapas em Configurar permissões de instância obrigatórias para o Systems Manager.

Tópicos

- Adicionar permissões do Session Manager a um perfil do IAM existente
- Crie uma função do IAM personalizada para o Session Manager

Adicionar permissões do Session Manager a um perfil do IAM existente

Use o procedimento a seguir para adicionar permissões do Session Manager a um perfil do AWS Identity and Access Management (IAM) já existente. Ao adicionar permissões a um perfil já existente, você pode aprimorar a segurança do ambiente de computação sem precisar usar a política `AmazonSSMManagedInstanceCore` da AWS para obter permissões de instância.

> **Note**
>
> Observe as seguintes informações:
>
> - Esse procedimento pressupõe que sua função existente já inclui outras permissões `ssm` do Systems Manager para ações que você deseja permitir o acesso. Essa política não é suficiente para usar o Session Manager.
> - O exemplo de política a seguir contém uma ação `s3:GetEncryptionConfiguration`. Essa ação será obrigatória se você escolher a opção Aplicar criptografia de log do S3 nas preferências de registro em log do Session Manager.

Para adicionar permissões do Session Manager a uma função existente (console)

1. Faça login no AWS Management Console e abra o console do IAM, em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Perfis.

3. Selecione o nome do perfil ao qual você está adicionando as permissões.
4. Escolha a aba Permissões.
5. Escolha Adicionar permissões e, em seguida, selecione Criar política em linha.
6. Selecione a guia JSON.
7. Substitua o conteúdo da política padrão pelo conteúdo a seguir. Substitua *key-name* pelo nome do recurso da Amazon (ARN) da chave do AWS Key Management Service (AWS KMS key) que você deseja usar.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssmmessages:CreateControlChannel",
                "ssmmessages:CreateDataChannel",
                "ssmmessages:OpenControlChannel",
                "ssmmessages:OpenDataChannel"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetEncryptionConfiguration"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-name"
        }
    ]
}
```

Para obter informações sobre como usar uma chave KMS para criptografar dados de sessão, consulte Ativar a criptografia de chaves do KMS de dados de sessão (console).

Se você não or usar a criptografia do AWS KMS para sua sessão de dados, poderá remover o seguinte conteúdo da política:

```
,
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "key-name"
        }
```

8. Escolha Próximo: tags.
9. (Opcional) Adicione tags escolhendo Add tag (Adicionar tag) e inserindo as tags preferenciais para a política.
10. Escolha Próximo: revisar.
11. Na página Revisar política, em Nome, digite um nome para a política em linha, como **`SessionManagerPermissions`**.
12. (Opcional) Em Descrição, digite uma descrição para a política.

    Escolha Criar política.

Para obter informações sobre as ações `ssmmessages`, consulte Referência: ec2messages, ssmmessages e outras operações da API.

Crie uma função do IAM personalizada para o Session Manager

É possível criar um perfil do AWS Identity and Access Management (IAM) que conceda ao Session Manager a permissão para realizar ações em suas instâncias gerenciadas do Amazon EC2. Você também pode incluir uma política para conceder as permissões necessárias para que os logs da sessão sejam enviados ao Amazon Simple Storage Service (Amazon S3) e ao Amazon CloudWatch Logs.

Depois de criar o perfil do IAM, para obter informações sobre como anexar o perfil a uma instância consulte Anexar ou substituir um perfil de instância no site do AWS re:Post. Para obter mais

informações sobre perfis de instância e perfis do IAM, consulte Usar perfis de instância no Guia do usuário do IAM e Perfis do IAM para Amazon EC2 no Guia do usuário do Amazon Elastic Compute Cloud para instâncias do Linux. Para obter mais informações sobre como criar um perfil de serviço do IAM para máquinas on-premises, consulte Criar o perfil de serviço do IAM obrigatório para o Systems Manager em ambientes híbridos e multinuvem.

Tópicos

- Crie uma função do IAM com permissões mínimas do Session Manager (console)
- Crie uma função do IAM com permissões para o Session Manager, Amazon S3 e CloudWatch Logs (console)

Crie uma função do IAM com permissões mínimas do Session Manager (console)

Use o procedimento a seguir para criar uma função do IAM personalizada com uma política que fornece permissões somente para ações do Session Manager em suas instâncias.

Para criar um perfil de instância com permissões mínimas do Session Manager (console)

1. Faça login no AWS Management Console e abra o console do IAM, em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Políticas e, em seguida, Criar política. (Se aparecer um botão Get Started (Iniciar), selecione-o e, em seguida, clique em Create Policy (Criar política).
3. Selecione a guia JSON.
4. Substitua o conteúdo padrão pela política a seguir. Para criptografar os dados de sessão usando o AWS Key Management Service (AWS KMS), substitua *key-name* pelo nome do recurso da Amazon (ARN) da AWS KMS key que você deseja usar.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:UpdateInstanceInformation",
                "ssmmessages:CreateControlChannel",
                "ssmmessages:CreateDataChannel",
```

```
                "ssmmessages:OpenControlChannel",
                "ssmmessages:OpenDataChannel"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-name"
        }
    ]
}
```

Para obter informações sobre como usar uma chave KMS para criptografar dados de sessão, consulte Ativar a criptografia de chaves do KMS de dados de sessão (console).

Se você não or usar a criptografia do AWS KMS para sua sessão de dados, poderá remover o seguinte conteúdo da política:

```
,
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "key-name"
        }
```

5. Escolha Próximo: tags.
6. (Opcional) Adicione tags escolhendo Add tag (Adicionar tag) e inserindo as tags preferenciais para a política.
7. Escolha Próximo: revisar.
8. Na página Revisar política, em Nome, digite um nome para a política em linha, como **`SessionManagerPermissions`**.
9. (Opcional) Em Descrição, digite uma descrição para a política.
10. Escolha Criar política.
11. No painel de navegação, escolha Perfis e Criar perfil.

12. Na página Criar perfil, escolha Serviço da AWS e, para Caso de uso, escolha EC2.
13. Escolha Próximo.
14. Na página Add permissions (Adicionar políticas), marque a caixa de seleção à esquerda do nome da política que acabou de criar, por exemplo, **`SessionManagerPermissions`**.
15. Escolha Próximo.
16. Na página Name, review, and create (Nomear, revisar e criar), em Role name (Nome da função), digite um nome para a função do IAM, por exemplo, **`MySessionManagerRole`**.
17. (Opcional) Em Role description (Descrição da função), digite uma descrição para o perfil de instância.
18. (Opcional) Adicione tags escolhendo Add tag (Adicionar tag) e inserindo as tags preferenciais para a função.

    Selecione Criar perfil.

Para obter informações sobre as ações `ssmmessages`, consulte Referência: ec2messages, ssmmessages e outras operações da API.

Crie uma função do IAM com permissões para o Session Manager, Amazon S3 e CloudWatch Logs (console)

Use o procedimento a seguir para criar uma função do IAM personalizada com uma política que fornece permissões para ações do Session Manager em suas instâncias. A política também fornece as permissões necessárias para que logs de sessão sejam armazenados em buckets do Amazon Simple Storage Service (Amazon S3) e em grupos de logs do Amazon CloudWatch Logs.

> **Important**
>
> Para gerar logs de sessão em um bucket do Amazon S3 que pertença a outra Conta da AWS, você deve adicionar a permissão do IAM `s3:PutObjectAcl` à política de perfil do IAM. Além disso, é necessário garantir que a política do bucket conceda acesso entre contas ao perfil do IAM usado pela conta proprietária para conceder permissões do Systems Manager para instâncias gerenciadas. Se o bucket usar a criptografia do Key Management Service (KMS), a política do KMS do bucket também deverá conceder esse acesso entre contas. Para obter mais informações sobre como configurar permissões de bucket entre contas no Amazon S3, consulte Granting cross-account bucket permissions no Guia do usuário do Amazon Simple Storage Service. Se as permissões entre contas não forem

adicionadas, a conta que possui o bucket do Amazon S3 não poderá acessar os logs de saída da sessão.

Para obter informações sobre como especificar as preferências para armazenar logs de sessão, consulte Habilitar e desabilitar o registro em log de sessão.

Para criar uma função do IAM com permissões para o Session Manager, Amazon S3 e CloudWatch Logs (console)

1. Faça login no AWS Management Console e abra o console do IAM, em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Políticas e, em seguida, Criar política. (Se aparecer um botão Get Started (Iniciar), selecione-o e, em seguida, clique em Create Policy (Criar política).
3. Selecione a guia JSON.
4. Substitua o conteúdo padrão pela política a seguir. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssmmessages:CreateControlChannel",
                "ssmmessages:CreateDataChannel",
                "ssmmessages:OpenControlChannel",
                "ssmmessages:OpenDataChannel",
                "ssm:UpdateInstanceInformation"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogGroups",
```

```
                "logs:DescribeLogStreams"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/s3-prefix/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetEncryptionConfiguration"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-name"
        },
        {
            "Effect": "Allow",
            "Action": "kms:GenerateDataKey",
            "Resource": "*"
        }
    ]
}
```

5. Escolha Próximo: tags.
6. (Opcional) Adicione tags escolhendo Add tag (Adicionar tag) e inserindo as tags preferenciais para a política.
7. Escolha Próximo: revisar.
8. Na página Revisar política, em Nome, digite um nome para a política em linha, como **`SessionManagerPermissions`**.
9. (Opcional) Em Descrição, digite uma descrição para a política.
10. Escolha Criar política.

11. No painel de navegação, escolha Perfis e Criar perfil.
12. Na página Criar perfil, escolha Serviço da AWS e, para Caso de uso, escolha EC2.
13. Escolha Próximo.
14. Na página Add permissions (Adicionar políticas), marque a caixa de seleção à esquerda do nome da política que acabou de criar, por exemplo, **`SessionManagerPermissions`**.
15. Escolha Próximo.
16. Na página Name, review, and create (Nomear, revisar e criar), em Role name (Nome da função), digite um nome para a função do IAM, por exemplo, **`MySessionManagerRole`**.
17. (Opcional) Em Role description (Descrição da função), digite uma descrição para a função.
18. (Opcional) Adicione tags escolhendo Add tag (Adicionar tag) e inserindo as tags preferenciais para a função.
19. Selecione Criar perfil.

Etapa 3: controlar o acesso da sessão pelos nós gerenciados

Você concede ou revoga o acesso do Session Manager a nós gerenciados usando políticas do AWS Identity and Access Management (IAM). É possível criar uma política e anexá-la a um usuário ou grupo do IAM que especifica a quais nós gerenciados o usuário ou grupo pode se conectar. Também é possível especificar as operações de API do Session Manager que o usuário ou os grupos podem realizar nesses nós gerenciados.

Para ajudar você a começar a usar as políticas de permissão do IAM para o Session Manager, criamos exemplos de políticas para um usuário final e um usuário administrador. Você pode usar essas políticas fazendo apenas pequenas alterações. Ou use esses exemplos como um guia para criar políticas personalizadas do IAM. Para ter mais informações, consulte Exemplo de políticas do IAM para Session Manager. Para obter informações sobre como criar políticas do IAM e anexá-las a usuários ou grupos, consulte Criação de políticas do IAM e Adição e remoção de políticas do IAM no Guia do usuário do IAM.

Sobre formatos de ARN do ID da sessão

Ao criar uma política do IAM para acesso do Session Manager, especifique um ID da sessão como parte do nome do recurso da Amazon (ARN). O ID da sessão contém o nome do usuário como variável. Para ajudar a ilustrar isso, veja o formato de um ARN do Session Manager e um exemplo:

```
arn:aws:ssm:region-id:account-id:session/session-id
```

Por exemplo:

```
arn:aws:ssm:us-east-2:123456789012:session/JohnDoe-1a2b3c4d5eEXAMPLE
```

Para obter mais informações sobre usar variáveis em políticas IAM, consulte Elementos da política do IAM: variáveis.

Tópicos

- Iniciar uma sessão de shell padrão especificando o documento da sessão padrão nas políticas do IAM
- Iniciar uma sessão de shell com um documento ao especificar os documentos da sessão nas políticas do IAM
- Exemplo de políticas do IAM para Session Manager
- Exemplos adicionais de políticas do IAM para o Session Manager

Iniciar uma sessão de shell padrão especificando o documento da sessão padrão nas políticas do IAM

Quando você configura o Session Manager para sua Conta da AWS ou altera as preferências de sessão no console do Systems Manager, o sistema cria um documento de sessão do SSM chamado `SSM-SessionManagerRunShell`. Trata-se do documento padrão da sessão. O Session Manager usa esse documento para armazenar suas preferências de sessão, que incluem informações como:

- Um local em que você deseja salvar os dados da sessão, como um bucket do Amazon Simple Storage Service (Amazon S3) ou um grupo de logs do Amazon CloudWatch Logs.
- Um ID de chave do AWS Key Management Service (AWS KMS) para criptografar os dados da sessão.
- Se o suporte para Executar como é permitido para suas sessões.

Este é um exemplo das informações contidas no documento de preferências da sessão `SSM-SessionManagerRunShell`.

```
{
  "schemaVersion": "1.0",
  "description": "Document to hold regional settings for Session Manager",
  "sessionType": "Standard_Stream",
  "inputs": {
```

```
    "s3BucketName": "amzn-s3-demo-bucket",
    "s3KeyPrefix": "MyS3Prefix",
    "s3EncryptionEnabled": true,
    "cloudWatchLogGroupName": "MyCWLogGroup",
    "cloudWatchEncryptionEnabled": false,
    "kmsKeyId": "1a2b3c4d",
    "runAsEnabled": true,
    "runAsDefaultUser": "RunAsUser"
  }
}
```

Por padrão, o Session Manager usa o documento de sessão padrão quando o usuário inicia a sessão pelo AWS Management Console. Isso se aplica ao Fleet Manager ou ao Session Manager no console do Systems Manager ou ao EC2 Connect no console do Amazon EC2. O Session Manager também usa o documento de sessão padrão quando um usuário inicia uma sessão usando um comando da AWS CLI, como neste exemplo:

```
aws ssm start-session \
    --target i-02573cafcfEXAMPLE
```

Para iniciar uma sessão padrão do shell, você deve especificar o documento da sessão padrão na política do IAM, conforme mostrado no exemplo a seguir.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnableSSMSession",
      "Effect": "Allow",
      "Action": [
        "ssm:StartSession"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:111122223333:instance/instance-id",
        "arn:aws:ssm:us-east-1:111122223333:document/SSM-SessionManagerRunShell"
      ]
    },
    {
      "Effect": "Allow",
```

```
      "Action": [
        "ssmmessages:OpenDataChannel"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Iniciar uma sessão de shell com um documento ao especificar os documentos da sessão nas políticas do IAM

Ao usar o comando start-session da AWS CLI usando o documento de sessão padrão, você poderá omitir o nome do documento. O sistema chama automaticamente o documento da sessão `SSM-SessionManagerRunShell`.

Em todos os outros casos é preciso especificar um valor para o parâmetro `document-name`. Quando o usuário especifica o nome de um documento de sessão em um comando, os sistemas verificam sua política do IAM para conferir se há permissão para acessar o documento. Se não houver permissão, a solicitação de conexão falhará. Os exemplos a seguir incluem o parâmetro `document-name` com o documento da sessão `AWS-StartPortForwardingSession`.

```
aws ssm start-session \
    --target i-02573cafcfEXAMPLE \
    --document-name AWS-StartPortForwardingSession \
    --parameters '{"portNumber":["80"], "localPortNumber":["56789"]}'
```

Para obter um exemplo de como especificar um documento de sessão do Session Manager em uma política do IAM, consulte Políticas do usuário final do Quickstart para o Session Manager.

> **Note**
>
> Para iniciar uma sessão usando SSH, é necessário concluir as etapas de configuração no nó gerenciado de destino e na máquina local do usuário. Para obter mais informações, consulte (Opcional) Permitir e controlar permissões para conexões SSH por meio do Session Manager.

Exemplo de políticas do IAM para Session Manager

Use os exemplos nesta seção para ajudar a criar políticas do AWS Identity and Access Management (IAM) que fornecem as permissões necessárias mais comuns para acesso ao Session Manager.

> **Note**
>
> Também é possível usar uma política de AWS KMS key para controlar quais entidades IAM (usuários ou perfis) e Contas da AWS recebem acesso à sua chave do KMS. Para obter informações, consulte Visão geral do gerenciamento do acesso aos seus recursos do AWS KMS e Usar políticas de chaves no AWS KMS no Manual do desenvolvedor do AWS Key Management Service.

Tópicos

- Políticas do usuário final do Quickstart para o Session Manager
- Política do administrador do Quickstart para o Session Manager

Políticas do usuário final do Quickstart para o Session Manager

Use os exemplos a seguir para criar políticas de usuário final do IAM para o Session Manager.

É possível criar uma política que permita que os usuários iniciem sessões apenas no console do Session Manager e na AWS Command Line Interface (AWS CLI), apenas no console do Amazon Elastic Compute Cloud (Amazon EC2) ou nos três.

Essas políticas fornecem aos usuários finais a capacidade de iniciar uma sessão de um nó gerenciado específico e encerrar apenas suas próprias sessões. Consulte o Exemplos adicionais de políticas do IAM para o Session Manager para obter exemplos de customização que você pode querer fazer na política.

Nas políticas de exemplo a seguir, substitua cada *`espaço reservado para recurso de exemplo`* por suas próprias informações.

Consulte as seções a seguir para exibir as políticas de exemplo para o intervalo de acesso à sessão que você deseja fornecer.

Session Manager and Fleet Manager

Use esta política de exemplo para conceder aos usuários a capacidade de iniciar e retomar sessões apenas dos consoles do Session Manager e do Fleet Manager.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ec2:us-
east-1:111122223333:instance/i-02573cafcfEXAMPLE",
                "arn:aws:ssm:us-east-1:111122223333:document/SSM-
SessionManagerRunShell"
            ]
        },
        {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
       },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeSessions",
                "ssm:GetConnectionStatus",
                "ssm:DescribeInstanceProperties",
                "ec2:DescribeInstances"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession",
                "ssm:ResumeSession"
            ],
```

```
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-name"
        }
    ]
}
```

Amazon EC2

Use esta política de exemplo para conceder aos usuários a capacidade de iniciar e retomar sessões apenas do console do Amazon EC2. Esta política não fornece todas as permissões necessárias para iniciar sessões do console do Session Manager e da AWS CLI.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession",
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ec2:us-
east-1:111122223333:instance/i-02573cafcfEXAMPLE",
                "arn:aws:ssm:us-east-1:111122223333:document/SSM-
SessionManagerRunShell"
            ]
        },
        {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
```

```
          "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetConnectionStatus",
                "ssm:DescribeInstanceInformation"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession",
                "ssm:ResumeSession"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:username}-*"
            ]
        }
    ]
}
```

AWS CLI

Use esta política de exemplo para conceder aos usuários a capacidade de iniciar e retomar sessões apenas via AWS CLI.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession",
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ec2:us-
east-1:111122223333:instance/i-02573cafcfEXAMPLE",
```

```
                "arn:aws:ssm:us-east-1:111122223333:document/SSM-
SessionManagerRunShell"
            ]
        },
        {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
       },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession",
                "ssm:ResumeSession"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-name"
        }
    ]
}
```

[1] `SSM-SessionManagerRunShell` é o nome padrão do documento do SSM que o Session Manager cria para armazenar suas preferências de configuração de sessão. Você pode criar um documento de sessão personalizado e especificá-lo nessa política. Você também pode especificar o documento `AWS-StartSSHSession` fornecido pela AWS para usuários que estão iniciando sessões usando SSH. Para obter informações sobre as etapas de configuração necessárias para oferecer suporte a sessões que usam SSH, consulte (Opcional) Permitir e controlar permissões para conexões SSH por meio do Session Manager.

[2] A permissão `kms:GenerateDataKey` permite a criação de uma chave de criptografia de dados que será usada para criptografar dados de sessão. Se você usar a criptografia do AWS Key Management Service (AWS KMS) para os dados da sessão, substitua *`key-name`* pelo nome do

recurso da Amazon (ARN) da chave KMS que você deseja usar, no formato `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-12345EXAMPLE`. Se você não for usar a criptografia de chaves do KMS para dados de sessão, remova o conteúdo da política a seguir.

```
{
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey"
            ],
            "Resource": "key-name"
        }
```

Para obter informações sobre o uso do AWS KMS para criptografar dados de sessão, consulte Ativar a criptografia de chaves do KMS de dados de sessão (console).

[3] A permissão para SendCommand é necessária nos casos em que um usuário tenta iniciar uma sessão via console do Amazon EC2, mas o SSM Agent deve ser atualizado para a versão mínima exigida do Session Manager primeiro. Run Command é usado para enviar um comando à instância para atualizar o agente.

Política do administrador do Quickstart para o Session Manager

Use os exemplos a seguir para criar políticas de administrador do IAM para o Session Manager.

Essas políticas fornecem aos administradores a capacidade de iniciar uma sessão para nós gerenciados marcados com `Key=Finance,Value=WebServers`, permissões para criar, atualizar e excluir preferências e permissões para encerrar apenas suas sessões próprias. Consulte o Exemplos adicionais de políticas do IAM para o Session Manager para obter exemplos de customização que você pode querer fazer na política.

É possível criar uma política que permita que os administradores executem essas tarefas apenas do console do Session Manager e da AWS CLI, apenas do console do Amazon EC2 ou dos três.

Nas políticas de exemplo a seguir, substitua cada *`espaço reservado para recurso de exemplo`* por suas próprias informações.

Consulte as seções a seguir para exibir as políticas de exemplo para os três cenários de permissões.

Session Manager and CLI

> Use esta política de exemplo para conceder aos administradores a capacidade de executar tarefas relacionadas à sessão somente do console do Session Manager e da AWS CLI. Esta

política não fornece todas as permissões necessárias para executar tarefas relacionadas à sessão do console do Amazon EC2.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ec2:*:111122223333:instance/*"
            ],
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/Finance": [
                        "WebServers"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssmmessages:OpenDataChannel"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeSessions",
                "ssm:GetConnectionStatus",
                "ssm:DescribeInstanceProperties",
                "ec2:DescribeInstances"
            ],
            "Resource": "*"
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "ssm:CreateDocument",
                "ssm:UpdateDocument",
                "ssm:GetDocument",
                "ssm:StartSession"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:document/SSM-
SessionManagerRunShell"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssmmessages:OpenDataChannel"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession",
                "ssm:ResumeSession"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
            ]
        }
    ]
}
```

Amazon EC2

Use esta política de exemplo para conceder aos administradores a capacidade de executar tarefas relacionadas à sessão somente do console do Amazon EC2. Essa política não fornece todas as permissões necessárias para executar tarefas relacionadas à sessão do console do Session Manager e da AWS CLI.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession",
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ec2:region:account-id:instance/*"
            ],
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/tag-key": [
                        "tag-value"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ssm:region:account-id:document/SSM-
SessionManagerRunShell"
            ]
        },
        {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
       },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetConnectionStatus",
                "ssm:DescribeInstanceInformation"
            ],
```

```
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession",
                "ssm:ResumeSession"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
            ]
        }
    ]
}
```

Session Manager, CLI, and Amazon EC2

Use esta política de exemplo para conceder aos administradores a capacidade de executar tarefas relacionadas à sessão do console do Session Manager, da AWS CLI e do console do Amazon EC2.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession",
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1:111122223333:instance/*"
            ],
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/tag-key": [
                        "tag-value"
                    ]
                }
```

```
            }
        },
        {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
       },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeSessions",
                "ssm:GetConnectionStatus",
                "ssm:DescribeInstanceInformation",
                "ssm:DescribeInstanceProperties",
                "ec2:DescribeInstances"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:CreateDocument",
                "ssm:UpdateDocument",
                "ssm:GetDocument",
                "ssm:StartSession"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:document/SSM-
SessionManagerRunShell"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession",
                "ssm:ResumeSession"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
            ]
        }
    ]
}
```

[1] A permissão para SendCommand é necessária nos casos em que um usuário tenta iniciar uma sessão do console do Amazon EC2, mas um comando deve ser enviado para atualizar o SSM Agent primeiro.

Exemplos adicionais de políticas do IAM para o Session Manager

Consulte os seguintes exemplos de políticas para ajudar você a criar uma política do AWS Identity and Access Management (IAM) personalizada para que qualquer usuário do Session Manager acesse cenários os quais deseja oferecer suporte.

Tópicos

- Exemplo 1: conceder aos usuários acesso a documentos do console
- Exemplo 2: restringir o acesso a nós gerenciados específicos
- Exemplo 3: restringir o acesso com base em etiquetas
- Exemplo 4: permitir que um usuário encerre somente sessões iniciadas por ele
- Exemplo 5: permitir acesso completo (administrativo) a todas as sessões

Exemplo 1: conceder aos usuários acesso a documentos do console

É possível permitir que os usuários especifiquem um documento personalizado ao iniciarem uma sessão usando o console do Gerenciador de Sessões. O exemplo de política do IAM a seguir concede permissão para acessar documentos com nomes que começam com **`SessionDocument-`** na Região da AWS e Conta da AWS especificadas.

Para usar essa política, substitua cada *`espaço reservado para recurso de exemplo`* por suas próprias informações.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetDocument",
                "ssm:ListDocuments"
```

```
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:document/SessionDocument-*"
            ]
        }
    ]
}
```

> **Note**
>
> O console do Gerenciador de Sessões oferece suporte somente a documentos de sessão que tenham um `sessionType` de `Standard_Stream` que sejam usados para definir as preferências da sessão. Para obter mais informações, consulte Esquema do documento de sessão.

Exemplo 2: restringir o acesso a nós gerenciados específicos

É possível criar uma política do IAM que defina a quais nós gerenciados um usuário pode se conectar usando o Gerenciador de Sessões. Por exemplo, a política a seguir concede a um usuário permissão para iniciar, encerrar e retomar as sessões em três nós específicos. A política restringe o usuário de se conectar a nós diferentes dos especificados.

> **Note**
>
> Para usuários federados, consulte Exemplo 4: permitir que um usuário encerre somente sessões iniciadas por ele.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
```

```
            ],
            "Resource": [
                "arn:aws:ec2:us-east-2:123456789012:instance/
i-1234567890EXAMPLE",
                "arn:aws:ec2:us-east-2:123456789012:instance/i-
abcdefghijEXAMPLE",
                "arn:aws:ec2:us-east-2:123456789012:instance/
i-0e9d8c7b6aEXAMPLE",
                "arn:aws:ssm:us-east-2:123456789012:document/SSM-
SessionManagerRunShell"
            ]
        },
        {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
       },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession",
                "ssm:ResumeSession"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
            ]
        },
        {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
       }
    ]
}
```

Exemplo 3: restringir o acesso com base em etiquetas

Você pode restringir o acesso a nós gerenciados com base em etiquetas específicas. No exemplo a seguir, o usuário pode iniciar e retomar sessões (`Effect: Allow, Action: ssm:StartSession, ssm:ResumeSession`) em qualquer nó gerenciado (`Resource: arn:aws:ec2:`*region*`:`*987654321098*`:instance/*`) com a condição de que o nó seja um Finance WebServer (`ssm:resourceTag/Finance: WebServer`). Se o usuário enviar um

comando para um nó gerenciado que não está etiquetado ou que tem qualquer outra etiqueta que não seja `Finance: WebServer`, o resultado do comando incluirá `AccessDenied`.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-2:123456789012:instance/*"
            ],
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/Finance": [
                        "WebServers"
                    ]
                }
            }
        },
        {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
       },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession",
                "ssm:ResumeSession"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
```

```
            ],
            "Resource": [
                "arn:aws:ssm:us-east-2:123456789012:document/SSM-
SessionManagerRunShell"
            ]
        }
    ]
}
```

Você pode criar políticas do IAM que permitem que um usuário inicie sessões para nós gerenciados que são marcados com várias etiquetas. A política a seguir permite que o usuário inicie sessões em nós gerenciados que tiverem ambas as etiquetas especificadas aplicadas a eles. Se um usuário enviar um comando para um nó gerenciado que não estiver marcado com ambas as etiquetas, o resultado do comando incluirá `AccessDenied`.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "ssm:StartSession"
         ],
         "Resource":"*",
         "Condition":{
            "StringLike":{
               "ssm:resourceTag/tag-key1":[
                  "tag-value1"
               ],
               "ssm:resourceTag/tag-key2":[
                  "tag-value2"
               ]
            }
         }
      },
      {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
```

```
        },
        {
              "Effect": "Allow",
              "Action": [
                  "ssm:StartSession"
              ],
              "Resource": [
                  "arn:aws:ssm:us-east-2:123456789012:document/SSM-
SessionManagerRunShell"
              ]
        }
    ]
}
```

Para obter mais informações sobre como criar políticas do IAM, consulte Políticas gerenciadas e em linha no Guia do usuário do IAM. Para obter mais informações sobre a marcação de nós gerenciados, consulte Marcar recursos do Amazon EC2 com tags no Guia do usuário do Amazon EC2 (o conteúdo é aplicável a nós gerenciados do Windows e do Linux). Para obter informações sobre como reforçar seu procedimento de segurança em relação a comandos em nível raiz não autorizados em seus nós gerenciados, consulte Restringir o acesso aos comandos em nível raiz por meio do SSM Agent

Exemplo 4: permitir que um usuário encerre somente sessões iniciadas por ele

O Session Manager fornece dois métodos para controlar quais sessões um usuário federado da conta da Conta da AWS tem permissão para encerrar.

- Use a variável `{aws:userid}` em uma política de permissões do AWS Identity and Access Management (IAM). Os usuários federados podem encerrar apenas as sessões que eles iniciaram. Para usuários não federados, use o Método 1. Para usuários federados, use o Método 2.
- Use as tags fornecidas pelas tags da AWS em uma política de permissões do IAM. Na política, inclua uma condição que permita que os usuários encerrem somente as sessões marcadas com as tags específicas fornecidas pela AWS. Esse método funciona para todas as contas, incluindo aquelas que usam IDs federados para conceder acesso à AWS.

Método 1: Conceder privilégios TerminateSession usando a variável **`{aws:username}`**

A política do IAM a seguir permite que um usuário exiba os IDs de todas as sessões na sua conta. No entanto, os usuários podem interagir com os nós gerenciados apenas por meio das sessões

que eles iniciaram. Um usuário que teve a política a seguir atribuída não poderá se conectar nem encerrar sessões de outros usuários. A política usa a variável `{aws:username}` para atingir isso.

> **Note**
>
> Esse método não funciona para contas que usam IDs federados para conceder acesso à AWS.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ssm:DescribeSessions"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Action": [
                "ssm:TerminateSession"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:username}-*"
            ]
        }
    ]
}
```

Método 2: Conceder privilégios TerminateSession usando tags fornecidas pela AWS

É possível controlar quais sessões um usuário poderá encerrar, incluindo variáveis de chave de etiqueta condicional em uma política do IAM. A condição determina que o usuário só pode

encerrar sessões marcadas com uma ou ambas as variáveis de chave de tag específicas e um valor especificado.

Quando um usuário em sua Conta da AWS inicia uma sessão, o Session Manager aplica duas tags de recurso a ela. A primeira tag de recurso é `aws:ssmmessages:target-id`, com a qual você especifica o ID do destino que o usuário tem permissão para encerrar. A outra tag de recurso é `aws:ssmmessages:session-id`, com um valor no formato de *role-id*:*caller-specified-role-name*.

> **Note**
>
> O Session Manager não oferece suporte a tags personalizadas para essa política de controle de acesso do IAM. Use as tags de recurso fornecidas pela AWS, descritas abaixo.

`aws:ssmmessages:target-id`

Com essa chave de etiqueta, inclua o ID do nó gerenciado como o valor na política. No bloco de política a seguir, a instrução da condição permite que um usuário encerre apenas o nó i-02573cafcfEXAMPLE.

JSON

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Action": [
                "ssm:TerminateSession"
             ],
             "Resource": "*",
             "Condition": {
                 "StringLike": {
                     "ssm:resourceTag/aws:ssmmessages:target-id": [
                        "i-02573cafcfEXAMPLE"
                     ]
                 }
             }
         }
     ]
}
```

Se o usuário tentar encerrar uma sessão para a qual não recebeu a permissão `TerminateSession`, ele receberá um erro `AccessDeniedException`.

`aws:ssmmessages:session-id`

Essa chave de tag inclui uma variável para o ID de sessão como o valor na solicitação para iniciar uma sessão.

O exemplo a seguir mostra uma política para casos em que o tipo de chamador é `User`. O valor fornecido em `aws:ssmmessages:session-id` é o ID do usuário. Neste exemplo, `AIDIODR4TAW7CSEXAMPLE` representa o ID de um usuário na sua conta da Conta da AWS. Para recuperar o ID de um usuário na sua Conta da AWS, use o comando `get-user` do IAM. Para obter mais informações, consulte get-user na seção AWS Identity and Access Management do Manual do usuário do IAM

JSON

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Action": [
                "ssm:TerminateSession"
             ],
             "Resource": "*",
             "Condition": {
                 "StringLike": {
                     "ssm:resourceTag/aws:ssmmessages:session-id": [
                        "AIDIODR4TAW7CSEXAMPLE"
                     ]
                 }
             }
         }
     ]
}
```

O exemplo a seguir mostra uma política para casos em que o tipo de chamador é `AssumedRole`. Você pode usar a variável `{aws:userid}` para o valor fornecido para

`aws:ssmmessages:session-id`. Como alternativa, você pode codificar um ID de função para o valor fornecido para `aws:ssmmessages:session-id`. Se você codificar um ID de função, deverá fornecer o valor no formato *`role-id`*`:`*`caller-specified-role-name`*. Por exemplo, `.AIDIODR4TAW7CSEXAMPLE:MyRole`

> **Important**
>
> Para que as tags do sistema sejam aplicadas, o ID de função fornecido pode conter apenas os seguintes caracteres: letras Unicode, 0–9, espaço, `_`, `.`, `:`, `/`, `=`, `+`, `-`, `@` e `\`.

Para recuperar o ID de uma função na sua Conta da AWS, use o comando `get-caller-identity`. Para obter informações, consulte get-caller-identity na Referência de comandos da AWS CLI.

JSON

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Action": [
                "ssm:TerminateSession"
             ],
             "Resource": "*",
             "Condition": {
                 "StringLike": {
                     "ssm:resourceTag/aws:ssmmessages:session-id": [
                        "${aws:userid}*"
                     ]
                 }
             }
         }
     ]
}
```

Se um usuário tentar encerrar uma sessão para a qual não recebeu a permissão `TerminateSession`, ele receberá um erro `AccessDeniedException`.

`aws:ssmmessages:target-id` e `aws:ssmmessages:session-id`

Você também pode criar políticas do IAM que permitem que um usuário encerre sessões marcadas com ambas as tags do sistema, conforme este exemplo.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "ssm:TerminateSession"
         ],
         "Resource":"*",
         "Condition":{
            "StringLike":{
               "ssm:resourceTag/aws:ssmmessages:target-id":[
                  "i-02573cafcfEXAMPLE"
               ],
               "ssm:resourceTag/aws:ssmmessages:session-id":[
                  "${aws:userid}*"
               ]
            }
         }
      }
   ]
}
```

Exemplo 5: permitir acesso completo (administrativo) a todas as sessões

A seguinte política do IAM permite que um usuário interaja totalmente com todos os nós gerenciados e todas as sessões criadas por todos os usuários para todos os nós. Ela deve ser concedida somente a um administrador que precisa de controle total sobre as atividades do Session Manager da sua organização.

JSON

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ssm:StartSession",
                "ssm:TerminateSession",
                "ssm:ResumeSession",
                "ssm:DescribeSessions",
                "ssm:GetConnectionStatus"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
       }
    ]
}
```

Etapa 4: Configurar preferências de sessão

Os usuários que receberam permissões administrativas em suas políticas do AWS Identity and Access Management (IAM) podem configurar as preferências da sessão, inclusive:

- Ative o suporte para Executar como para os nós gerenciados do Linux. Isso possibilita iniciar sessões usando as credenciais de um usuário do sistema operacional especificado, em vez de usar as credenciais de uma conta `ssm-user` gerada pelo sistema que o Session Manager do AWS Systems Manager pode criar em um nó gerenciado.
- Configure o Session Manager para usar a criptografia do AWS KMS key para fornecer proteção adicional aos dados transmitidos entre máquinas clientes e nós gerenciados.
- Configure o Session Manager para criar e enviar logs de histórico de sessão a um bucket do Amazon Simple Storage Service (Amazon S3) ou a um grupo de logs do Amazon CloudWatch Logs. Os dados de log armazenados podem ser usados para relatar as conexões da sessão feitas em seus nós gerenciados e os comandos executados neles durante as sessões.
- Configurar os limites de tempo da sessão. Você pode usar essa configuração para especificar quando terminar uma sessão após um período de inatividade.

- Configure o Session Manager para usar perfis de shell configuráveis. Esses perfis personalizáveis permitem que você defina opções preferenciais dentro das sessões, como preferências de shell, variáveis de ambiente, diretórios de trabalho e execução de vários comandos quando uma sessão é iniciada.

Para obter mais informações sobre as permissões necessárias para configurar preferências do Session Manager, consulte the section called “Conceder ou negar permissões a um usuário para atualizar as preferências do Session Manager”.

Tópicos

- Conceder ou negar permissões a um usuário para atualizar as preferências do Session Manager
- Especificar um valor de tempo limite de sessão ociosa
- Especifique a duração máxima da sessão
- Permitir perfis de shell configuráveis
- Ative o suporte a Executar como para nós gerenciados do Linux e do macOS
- Ativar a criptografia de chaves do KMS de dados de sessão (console)
- Criar um documento de preferências (linha de comando) do Session Manager
- Atualizar preferências do Session Manager (linha de comando)

Para obter informações sobre como usar o console do Systems Manager para configurar as opções de registro em log dos dados da sessão, consulte os seguintes tópicos:

- Registrar dados da sessão em log usando o Amazon S3 (console)
- Transmitir dados da sessão usando o Amazon CloudWatch Logs (console)
- Registrar dados da sessão em log usando o Amazon CloudWatch Logs (console)

Conceder ou negar permissões a um usuário para atualizar as preferências do Session Manager

As preferências da conta são armazenadas como documentos do AWS Systems Manager SSM para Região da AWS. Antes que um usuário possa atualizar as preferências de conta para sessões em sua conta, eles devem receber as permissões necessárias para acessar o tipo de documento do SSM onde essas preferências estão armazenadas. Essas permissões são concedidas por meio de uma política do AWS Identity and Access Management (IAM).

A política de administrador para permitir que as preferências sejam criadas e atualizadas

Um administrador pode ter a seguinte política para criar e atualizar as preferências a qualquer momento. A política a seguir dá permissão para acessar e atualizar o documento `SSM-SessionManagerRunShell` em uma conta 123456789012 na us-east-2.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ssm:CreateDocument",
                "ssm:GetDocument",
                "ssm:UpdateDocument",
                "ssm:DeleteDocument"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:ssm:us-east-2:123456789012:document/SSM-
SessionManagerRunShell"
            ]
        }
    ]
}
```

Política de usuário para impedir que as preferências sejam atualizados

Use a política a seguir para impedir que os usuários finais da sua conta atualizem ou substitua as preferências do Session Manager.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ssm:CreateDocument",
                "ssm:GetDocument",
                "ssm:UpdateDocument",
```

```
                "ssm:DeleteDocument"
            ],
            "Effect": "Deny",
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:document/SSM-
SessionManagerRunShell"
            ]
        }
    ]
}
```

Especificar um valor de tempo limite de sessão ociosa

O Session Manager, uma ferramenta do AWS Systems Manager, permite especificar o tempo para permitir que um usuário fique inativo antes do sistema encerrar uma sessão. Por padrão, as sessões expiram após 20 minutos de inatividade. Você pode modificar essa configuração para especificar que uma sessão expira entre 1 e 60 minutos de inatividade. Algumas agências profissionais de segurança de computação recomendam definir tempos limite de sessão ociosa para um máximo de 15 minutos.

Para permitir o tempo limite da sessão ociosa (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Session Manager.
3. Escolha a guia Preferences (Preferências) e, em seguida, escolha Edit (Editar).
4. Especifique a quantidade de tempo para permitir que um usuário fique inativo antes que uma sessão termine no campo minutes (minutos) em Idle session timeout (Tempo limite de sessão ociosa).
5. Escolha Salvar.

Especifique a duração máxima da sessão

O Session Manager, uma ferramenta do AWS Systems Manager, permite que você especifique a duração máxima de uma sessão antes que ela termine. Por padrão, sessões não têm duração máxima. O valor especificado para a duração máxima da sessão deve estar entre 1 e 1.440 minutos.

Para especificar a duração máxima da sessão (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Session Manager.
3. Escolha a guia Preferences (Preferências) e, em seguida, escolha Edit (Editar).
4. Marque a caixa de seleção ao lado de Enable maximum session duration (Ativar a duração máxima da sessão).
5. Especifique a duração máxima da sessão antes que ela termine, no campo minutes (minutos) em Maximum session duration (Duração máxima da sessão).
6. Escolha Salvar.

Permitir perfis de shell configuráveis

Por padrão, as sessões nas instâncias do EC2 para Linux começam a usar o shell Bourne (sh). No entanto, talvez prefira usar outro shell como bash. Ao permitir perfis de shell configuráveis, você pode personalizar preferências dentro de sessões como preferências de shell, variáveis de ambiente, diretórios de trabalho e executar vários comandos quando uma sessão é iniciada.

Important

O Systems Manager não verifica os comandos ou scripts em seu perfil de shell para ver quais alterações eles fariam em uma instância antes de serem executados. Para restringir a capacidade de um usuário modificar comandos ou scripts inseridos em seu perfil de shell, recomendamos o seguinte:

- Crie um documento personalizado do tipo de sessão para os usuários e funções do AWS Identity and Access Management (IAM). Em seguida, modifique a política do IAM para esses usuários e funções para que a operação da API `StartSession` só possa usar o documento do tipo Sessão que você criou para eles. Para obter informações, consulte Criar um documento de preferências (linha de comando) do Session Manager e Políticas do usuário final do Quickstart para o Session Manager.
- Modifique a política do IAM para seus usuários e funções do IAM para negar acesso à operação da API `UpdateDocument` do recurso de documento do tipo Sessão que você criar. Isso permite que seus usuários e funções usem o documento que você criou para suas preferências de sessão sem permitir que eles modifiquem qualquer uma das configurações.

Para ativar perfis de shell configuráveis

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Session Manager.
3. Escolha a guia Preferences (Preferências) e, em seguida, escolha Edit (Editar).
4. Especifique as variáveis de ambiente, preferências de shell ou comandos que você deseja executar quando a sessão for iniciada nos campos dos sistemas operacionais aplicáveis.
5. Escolha Salvar.

Os comandos a seguir são exemplos que podem ser adicionados ao seu perfil do shell.

Altere para o shell bash e para o diretório /usr nas instâncias do Linux.

```
exec /bin/bash
cd /usr
```

Emita um carimbo de data/hora e uma mensagem de boas-vindas no início de uma sessão.

Linux & macOS

```
timestamp=$(date '+%Y-%m-%dT%H:%M:%SZ')
user=$(whoami)
echo $timestamp && echo "Welcome $user"'!'
echo "You have logged in to a production instance. Note that all session activity is
 being logged."
```

Windows

```
$timestamp = (Get-Date).ToString("yyyy-MM-ddTH:mm:ssZ")
$splitName = (whoami).Split("\")
$user = $splitName[1]
Write-Host $timestamp
Write-Host "Welcome $user!"
Write-Host "You have logged in to a production instance. Note that all session
 activity is being logged."
```

Visualizar a atividade dinâmica do sistema no início de uma sessão.

Linux & macOS

```
top
```

Windows

```
while ($true) { Get-Process | Sort-Object -Descending CPU | Select-Object -First 30;
 `
Start-Sleep -Seconds 2; cls
Write-Host "Handles  NPM(K)    PM(K)      WS(K) VM(M)   CPU(s)     Id ProcessName";
Write-Host "-------  ------    -----      ----- -----   ------     -- -----------"}
```

Ative o suporte a Executar como para nós gerenciados do Linux e do macOS

Por padrão, o Session Manager autentica as conexões ao usar as credenciais da conta `ssm-user` gerada pelo sistema que é criada em um nó gerenciado. (Em máquinas macOS e Linux, a conta é adicionada ao `/etc/sudoers/`). Se desejar, em vez disso, é possível autenticar as sessões usando as credenciais de uma conta de usuário do sistema operacional (SO) ou um usuário de domínio para instâncias ingressadas para um Active Directory. Nesse caso, o Session Manager verifica se a conta do SO especificada existe no nó ou no domínio antes de iniciar a sessão. Se você tentar iniciar uma sessão usando uma conta do SO que não existe no nó ou no domínio, a conexão falhará.

> **Note**
>
> O Gerenciador de Sessões não oferece suporte ao uso de uma conta de usuário `root` do sistema operacional para autenticar conexões. Para sessões que são autenticadas usando uma conta de usuário do SO, o nível do sistema operacional e as políticas de diretório do nó, como restrições de login ou restrições de uso de recursos do sistema, podem não se aplicar.

Como funciona

Se você ativar o suporte a Run As (Executar como) para sessões, o sistema verificará a existência de permissões de acesso da seguinte forma:

1. Para o usuário que está iniciando a sessão, a entidade do IAM (usuário ou perfil) foi marcada com `SSMSessionRunAs = os user account name`?

Em caso afirmativo, o nome do usuário do SO existe no nó gerenciado? Se a resposta for sim, iniciar a sessão. Se isso não acontecer, não permita que uma sessão seja iniciada.

Se a entidade do IAM não tiver sido marcada com `SSMSessionRunAs = os user account name`, continue para a etapa 2.

2. Se a entidade do IAM não foi marcada com `SSMSessionRunAs = os user account name`, um nome de usuário do SO foi especificado nas preferências do Session Manager da Conta da AWS?

   Em caso afirmativo, o nome do usuário do SO existe no nó gerenciado? Se a resposta for sim, iniciar a sessão. Se isso não acontecer, não permita que uma sessão seja iniciada.

> **Note**
>
> Quando você ativa o suporte “Executar como”, ele evita que o Gerenciador de Sessões inicie sessões usando a conta `ssm-user` em um nó gerenciado. Isso significa que, se o Session Manager falhar ao se conectar usando a conta de usuário do SO especificada, ele não voltará a se conectar pelo método padrão.
> Se você ativar “Executar como” sem especificar uma conta do SO ou marcar uma entidade do IAM e não tiver especificado uma conta do SO nas preferências do Gerenciador de Sessões, as tentativas de conexão da sessão falharão.

Para ativar o suporte para Executar como para nós gerenciados do Linux e do macOS

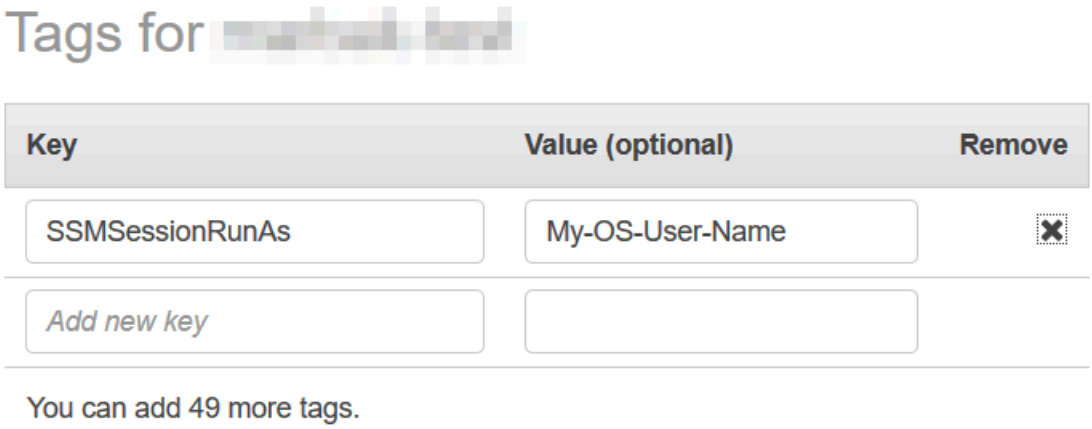
1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Session Manager.
3. Escolha a guia Preferences (Preferências) e, em seguida, escolha Edit (Editar).
4. Marque a caixa de seleção ao lado de Habilitar o suporte a Executar como para instâncias do Linux.
5. Execute um destes procedimentos:

   - Opção 1: no campo Nome de usuário do sistema operacional, insira o nome da conta de usuário do SO que você deseja usar para iniciar as sessões. Usando essa opção, todas as sessões são executadas pelo mesmo usuário do SO para todos os usuários em sua Conta da AWS que se conectam usando o Session Manager.

- Opção 2 (recomendada): escolha o link Abrir o console do IAM. No painel de navegação, selecione Users (Usuários) ou Roles (Funções). Escolha a entidade (usuário ou função) à qual adicionar tags e escolha a guia Tags. Insira `SSMSessionRunAs` para o nome da chave. Insira o nome de uma conta de usuário do SO para o valor da chave. Escolha Salvar alterações.

  Usando esta opção, é possível especificar usuários únicos do SO para diferentes entidades do IAM, se desejar. Para obter mais informações sobre a marcação de entidades do IAM (usuários ou perfis), consulte Recursos de marcação do IAM no Guia do usuário do IAM.

  Veja um exemplo a seguir.



6. Escolha Salvar.

Ativar a criptografia de chaves do KMS de dados de sessão (console)

Use o AWS Key Management Service (AWS KMS) para criar e gerenciar chaves de criptografia. Com o AWS KMS, você pode controlar o uso da criptografia em uma grande variedade de Serviços da AWS e nos seus aplicativos. Você pode especificar que os dados da sessão transmitidos entre os nós gerenciados e os computadores locais dos usuários na sua Conta da AWS sejam criptografados usando a criptografia de chaves do KMS. (Isso é um complemento à criptografia TLS 1.2/1.3 que já é fornecida pela AWS por padrão.) Para criptografar os dados da sessão do Session Manager, crie uma chave do KMS simétrica usando o AWS KMS.

A criptografia do AWS KMS está disponível para os tipos de sessão `Standard_Stream`, `InteractiveCommands` e `NonInteractiveCommands`. Para usar a opção de criptografar dados da sessão usando uma chave criada no AWS KMS (AWS Systems Manager), a versão 2.3.539.0 ou posterior do SSM Agent deve ser instalada em seu nó gerenciado.

> **Note**
>
> Você deve permitir a criptografia do AWS KMS para redefinir senhas em seus nós gerenciados no console do AWS Systems Manager. Para ter mais informações, consulte Redefina uma senha em um nó gerenciado.

Você pode usar uma chave que criou na sua Conta da AWS. Também pode usar uma chave criada em uma Conta da AWS diferente. O criador da chave em uma Conta da AWS diferente deve fornecer as permissões necessárias para usar essa chave.

Depois que você habilitar a criptografia de chaves do KMS para dados de sessão, os usuários que iniciarem sessões e nós gerenciados, às quais eles estiverem conectados, deverão ter permissão para usar essa chave. Você fornece permissão para usar a chave KMS com o Session Manager por meio de políticas do IAM AWS Identity and Access Management. Para obter informações, consulte os seguintes tópicos:

- Adicione permissões de AWS KMS para usuários na sua conta: Exemplo de políticas do IAM para Session Manager
- Adicione permissões do AWS KMS para nós gerenciados na sua conta: Etapa 2: verificar ou adicionar permissões de instância para o Session Manager

Para obter mais informações sobre como criar e gerenciar chaves KMS, consulte o Guia do desenvolvedor do AWS Key Management Service.

Para obter informações sobre como usar a AWS CLI para ativar a criptografia de chaves do KMS de dados de sessão na sua conta, consulte Criar um documento de preferências (linha de comando) do Session Manager ou Atualizar preferências do Session Manager (linha de comando).

> **Note**
>
> Há cobrança para usar chaves do KMS. Para obter mais informações, consulte Definição de preço do AWS Key Management Service.

Para ativar a criptografia de chaves do KMS de dados de sessão (console)

1. Abra o console do AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Session Manager.
3. Escolha a guia Preferences (Preferências) e, em seguida, escolha Edit (Editar).
4. Marque a caixa de seleção ao lado de Enable KMS encryption (Ativar criptografia KMS).
5. Execute um destes procedimentos:

   - Selecione o botão ao lado de Select an KMS key in my current account (Selecione uma chave KMS em minha conta atual) e escolha uma chave da lista.

     - ou -

     Escolha o botão ao lado de Enter a KMS key alias or KMS key ARN (Insira um alias de chave KMS ou um ARN de chave KMS). Insira manualmente um alias de chave do KMS para uma chave criada na sua conta atual ou insira o nome do recurso da Amazon (ARN) de uma chave em outra conta. Veja os exemplos a seguir:

     - Alias da chave: `alias/my-kms-key-alias`
     - Nome de região da Amazon (ARN) do alias da chave: `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-12345EXAMPLE`

     - ou -

     Escolha Create new key (Criar nova chave) para criar uma nova chave do KMS na sua conta. Depois de criar a nova chave, retorne à guia Preferences (Preferências) e selecione a chave para criptografar dados de sessão na sua conta.

   Para obter mais informações sobre como compartilhar chaves, consulte Allowing External Contas da AWS to Access a key no Guia do desenvolvedor do AWS Key Management Service.
6. Escolha Salvar.

Criar um documento de preferências (linha de comando) do Session Manager

Use o procedimento a seguir para criar documentos do SSM que definam suas preferências para sessões do Session Manager do AWS Systems Manager. É possível usar o documento para configurar as opções da sessão, inclusive criptografia de dados, duração da sessão e registro em

log. Por exemplo, você pode especificar se armazenará dados de log de sessão em um bucket do Amazon Simple Storage Service (Amazon S3) ou em um grupo de logs do Amazon CloudWatch Logs. Você pode criar documentos que definam preferências gerais para todas as sessões de uma Conta da AWS e Região da AWS ou que definam preferências para sessões individuais.

> **Note**
>
> Também é possível configurar as preferências gerais da sessão usando o console do Gerenciador de Sessões.

Os documentos usados para definir as preferências do Gerenciador de Sessões devem ter um `sessionType` de `Standard_Stream`. Para obter mais informações sobre documentos de sessões, consulte the section called “Esquema do documento de sessão”.

Para obter informações sobre como usar a linha de comando para atualizar as preferências atuais do Session Manager, consulte Atualizar preferências do Session Manager (linha de comando).

Para obter um exemplo de como criar preferências de sessão usando o AWS CloudFormation, consulte Create a Systems Manager document for Session Manager preferences (Criar um documento do Systems Manager para as preferências do Session Manager no Manual do usuário do AWS CloudFormation.

> **Note**
>
> Esse procedimento descreve como criar documentos para definir preferências do Session Manager no nível da Conta da AWS. Para criar documentos que serão usados para definir as preferências no nível da sessão, especifique um valor diferente de `SSM-SessionManagerRunShell` das entradas de comando relacionadas ao nome do arquivo. Para usar seu documento para definir preferências para sessões iniciadas por meio da AWS Command Line Interface (AWS CLI), forneça o nome do documento como valor do parâmetro `--document-name`. Para definir preferências para sessões iniciadas no console do Gerenciador de Sessões, você pode digitar ou selecionar o nome do documento em uma lista.

Como criar preferências do Session Manager (linha de comando)

1. Crie um arquivo JSON em sua máquina local com um nome, como `SessionManagerRunShell.json` e, em seguida, cole o conteúdo a seguir nesse arquivo.

```
{
    "schemaVersion": "1.0",
    "description": "Document to hold regional settings for Session Manager",
    "sessionType": "Standard_Stream",
    "inputs": {
        "s3BucketName": "",
        "s3KeyPrefix": "",
        "s3EncryptionEnabled": true,
        "cloudWatchLogGroupName": "",
        "cloudWatchEncryptionEnabled": true,
        "cloudWatchStreamingEnabled": false,
        "kmsKeyId": "",
        "runAsEnabled": false,
        "runAsDefaultUser": "",
        "idleSessionTimeout": "",
        "maxSessionDuration": "",
        "shellProfile": {
            "windows": "date",
            "linux": "pwd;ls"
        }
    }
}
```

Também é possível passar valores para suas preferências de sessão usando parâmetros em vez de codificar os valores, conforme mostrado no exemplo a seguir.

```
{
   "schemaVersion":"1.0",
   "description":"Session Document Parameter Example JSON Template",
   "sessionType":"Standard_Stream",
   "parameters":{
      "s3BucketName":{
         "type":"String",
         "default":""
      },
      "s3KeyPrefix":{
         "type":"String",
```

```
            "default":""
         },
         "s3EncryptionEnabled":{
            "type":"Boolean",
            "default":"false"
         },
         "cloudWatchLogGroupName":{
            "type":"String",
            "default":""
         },
         "cloudWatchEncryptionEnabled":{
            "type":"Boolean",
            "default":"false"
         }
      },
      "inputs":{
         "s3BucketName":"{{s3BucketName}}",
         "s3KeyPrefix":"{{s3KeyPrefix}}",
         "s3EncryptionEnabled":"{{s3EncryptionEnabled}}",
         "cloudWatchLogGroupName":"{{cloudWatchLogGroupName}}",
         "cloudWatchEncryptionEnabled":"{{cloudWatchEncryptionEnabled}}",
         "kmsKeyId":""
      }
}
```

2. Especifique em que local você deseja enviar dados de sessão. Você pode especificar o nome de um bucket do S3 (com um prefixo opcional) ou um nome de grupo de logs do CloudWatch Logs. Se você quiser criptografar ainda mais os dados entre o cliente local e os nós gerenciados, forneça a chave KMS a ser usada para a criptografia. Veja um exemplo a seguir.

```
{
  "schemaVersion": "1.0",
  "description": "Document to hold regional settings for Session Manager",
  "sessionType": "Standard_Stream",
  "inputs": {
    "s3BucketName": "amzn-s3-demo-bucket",
    "s3KeyPrefix": "MyS3Prefix",
    "s3EncryptionEnabled": true,
    "cloudWatchLogGroupName": "MyLogGroupName",
    "cloudWatchEncryptionEnabled": true,
    "cloudWatchStreamingEnabled": false,
    "kmsKeyId": "MyKMSKeyID",
    "runAsEnabled": true,
```

```
    "runAsDefaultUser": "MyDefaultRunAsUser",
    "idleSessionTimeout": "20",
    "maxSessionDuration": "60",
    "shellProfile": {
        "windows": "MyCommands",
        "linux": "MyCommands"
    }
  }
}
```

> **Note**
>
> Se você não quiser criptografar dados de log da sessão, altere `true` para `false` em `s3EncryptionEnabled`.
> Se você não estiver enviando logs para um bucket do Amazon S3 ou para um grupo de logs do CloudWatch Logs, se não quiser criptografar os dados da sessão ativa ou não quiser habilitar o suporte a Run As (Executar como) para as sessões em sua conta, exclua as linhas para essas opções. Certifique-se de que a última linha na seção `inputs` não termine com uma vírgula.
> Se você adicionar um ID de chave do KMS para criptografar dados de sessão, os usuários que iniciarem sessões e os nós gerenciados aos quais eles se conectarem deverão ter permissão para usar essa chave. Você fornece permissão para usar a chave KMS com o Session Manager por meio de políticas do IAM. Para obter informações, consulte os seguintes tópicos:
>
> - Adicione permissões de AWS KMS para usuários na sua conta: Exemplo de políticas do IAM para Session Manager
> - Adicione permissões do AWS KMS para nós gerenciados na sua conta: Etapa 2: verificar ou adicionar permissões de instância para o Session Manager

3. Salve o arquivo.
4. No diretório em que você criou o arquivo JSON, execute o seguinte comando:

   Linux & macOS

```
aws ssm create-document \
    --name SSM-SessionManagerRunShell \
    --content "file://SessionManagerRunShell.json" \
    --document-type "Session" \
```

```
    --document-format JSON
```

Windows

```
aws ssm create-document ^
    --name SSM-SessionManagerRunShell ^
    --content "file://SessionManagerRunShell.json" ^
    --document-type "Session" ^
    --document-format JSON
```

PowerShell

```
New-SSMDocument `
    -Name "SSM-SessionManagerRunShell" `
    -Content (Get-Content -Raw SessionManagerRunShell.json) `
    -DocumentType "Session" `
    -DocumentFormat JSON
```

Se houver êxito, o comando gerará uma saída semelhante à seguinte.

```
{
    "DocumentDescription": {
        "Status": "Creating",
        "Hash": "ce4fd0a2ab9b0fae759004ba603174c3ec2231f21a81db8690a33eb66EXAMPLE",
        "Name": "SSM-SessionManagerRunShell",
        "Tags": [],
        "DocumentType": "Session",
        "PlatformTypes": [
            "Windows",
            "Linux"
        ],
        "DocumentVersion": "1",
        "HashType": "Sha256",
        "CreatedDate": 1547750660.918,
        "Owner": "111122223333",
        "SchemaVersion": "1.0",
        "DefaultVersion": "1",
        "DocumentFormat": "JSON",
        "LatestVersion": "1"
    }
```

```
}
```

Atualizar preferências do Session Manager (linha de comando)

O procedimento a seguir descreve como usar sua ferramenta da linha de comando preferencial para fazer alterações nas preferências do Session Manager do AWS Systems Manager para sua Conta da AWS na Região da AWS selecionada. Use o Session Manager para especificar opções para o registro em log de dados de sessão em um bucket do Amazon Simple Storage Service (Amazon S3) ou em um grupo de logs do Amazon CloudWatch Logs. Você também pode usar preferências do Session Manager para criptografar seus dados de sessão.

Como atualizar preferências do Session Manager (linha de comando)

1. Crie um arquivo JSON em sua máquina local com um nome, como `SessionManagerRunShell.json` e, em seguida, cole o conteúdo a seguir nesse arquivo.

```
{
    "schemaVersion": "1.0",
    "description": "Document to hold regional settings for Session Manager",
    "sessionType": "Standard_Stream",
    "inputs": {
        "s3BucketName": "",
        "s3KeyPrefix": "",
        "s3EncryptionEnabled": true,
        "cloudWatchLogGroupName": "",
        "cloudWatchEncryptionEnabled": true,
        "cloudWatchStreamingEnabled": false,
        "kmsKeyId": "",
        "runAsEnabled": true,
        "runAsDefaultUser": "",
        "idleSessionTimeout": "",
        "maxSessionDuration": "",
        "shellProfile": {
            "windows": "date",
            "linux": "pwd;ls"
        }
    }
}
```

2. Especifique em que local você deseja enviar dados de sessão. Você pode especificar o nome de um bucket do S3 (com um prefixo opcional) ou um nome de grupo de logs do CloudWatch Logs.

Se você quiser criptografar ainda mais os dados entre o cliente local e os nós gerenciados, forneça o AWS KMS key a ser usado para a criptografia. Veja um exemplo a seguir.

```
{
  "schemaVersion": "1.0",
  "description": "Document to hold regional settings for Session Manager",
  "sessionType": "Standard_Stream",
  "inputs": {
    "s3BucketName": "amzn-s3-demo-bucket",
    "s3KeyPrefix": "MyS3Prefix",
    "s3EncryptionEnabled": true,
    "cloudWatchLogGroupName": "MyLogGroupName",
    "cloudWatchEncryptionEnabled": true,
    "cloudWatchStreamingEnabled": false,
    "kmsKeyId": "MyKMSKeyID",
    "runAsEnabled": true,
    "runAsDefaultUser": "MyDefaultRunAsUser",
    "idleSessionTimeout": "20",
    "maxSessionDuration": "60",
    "shellProfile": {
        "windows": "MyCommands",
        "linux": "MyCommands"
    }
  }
}
```

Note

Se você não quiser criptografar dados de log da sessão, altere `true` para `false` em `s3EncryptionEnabled`.

Se você não estiver enviando logs para um bucket do Amazon S3 ou para um grupo de logs do CloudWatch Logs, se não quiser criptografar os dados da sessão ativa ou não quiser habilitar o suporte a Run As (Executar como) para as sessões em sua conta, exclua as linhas para essas opções. Certifique-se de que a última linha na seção `inputs` não termine com uma vírgula.

Se você adicionar um ID de chave do KMS para criptografar dados de sessão, os usuários que iniciarem sessões e os nós gerenciados aos quais eles se conectarem deverão ter permissão para usar essa chave. Você fornece permissão para usar a chave KMS com o Session Manager por meio de políticas do IAM AWS Identity and Access Management. Para obter informações, consulte os seguintes tópicos:

- Adicione permissões de AWS KMS para usuários na sua conta: Exemplo de políticas do IAM para Session Manager
- Adicione permissões do AWS KMS para nós gerenciados na sua conta: Etapa 2: verificar ou adicionar permissões de instância para o Session Manager

3. Salve o arquivo.
4. No diretório em que você criou o arquivo JSON, execute o seguinte comando:

   Linux & macOS

   ```
   aws ssm update-document \
       --name "SSM-SessionManagerRunShell" \
       --content "file://SessionManagerRunShell.json" \
       --document-version "\$LATEST"
   ```

   Windows

   ```
   aws ssm update-document ^
       --name "SSM-SessionManagerRunShell" ^
       --content "file://SessionManagerRunShell.json" ^
       --document-version "$LATEST"
   ```

   PowerShell

   ```
   Update-SSMDocument `
       -Name "SSM-SessionManagerRunShell" `
       -Content (Get-Content -Raw SessionManagerRunShell.json) `
       -DocumentVersion '$LATEST'
   ```

   Se houver êxito, o comando gerará uma saída semelhante à seguinte.

   ```
   {
       "DocumentDescription": {
           "Status": "Updating",
           "Hash": "ce4fd0a2ab9b0fae759004ba603174c3ec2231f21a81db8690a33eb66EXAMPLE",
           "Name": "SSM-SessionManagerRunShell",
           "Tags": [],
           "DocumentType": "Session",
   ```

```
        "PlatformTypes": [
            "Windows",
            "Linux"
        ],
        "DocumentVersion": "2",
        "HashType": "Sha256",
        "CreatedDate": 1537206341.565,
        "Owner": "111122223333",
        "SchemaVersion": "1.0",
        "DefaultVersion": "1",
        "DocumentFormat": "JSON",
        "LatestVersion": "2"
    }
}
```

Etapa 5: (Opcional) Restringir o acesso a comandos em uma sessão

Você pode restringir os comandos que um usuário pode executar em uma sessão do AWS Systems Manager Session Manager usando um documento do AWS Systems Manager (SSM) do tipo `Session` personalizado. No documento, defina o comando que é executado quando o usuário inicia uma sessão e os parâmetros que o usuário pode fornecer ao comando. A `schemaVersion` do documento do `Session` deve ser 1.0 e o `sessionType` do documento deve ser `InteractiveCommands`. É possível criar políticas do AWS Identity and Access Management (IAM) que permitem que os usuários acessem apenas os documentos do `Session` definidos por você. Para obter mais informações sobre como usar políticas do IAM para restringir o acesso a comandos em uma sessão, consulte Exemplos de políticas do IAM para comandos interativos.

Documentos com o `sessionType` de `InteractiveCommands` são compatíveis somente com sessões iniciadas pela AWS Command Line Interface (AWS CLI). O usuário fornece o nome do documento personalizado como o valor do parâmetro `--document-name` e fornece qualquer valor de parâmetro do comando usando a opção `--parameters`. Para obter mais informações sobre como executar comandos interativos, consulte Iniciar uma sessão (comandos interativos e não interativos).

Use o procedimento a seguir para criar um documento do SSM tipo `Session` personalizado que define o comando que um usuário tem permissão para executar.

Restringir acesso a comandos em uma sessão (console)

Para restringir os comandos que um usuário pode executar em uma sessão do Session Manager (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Documents.
3. Escolha Create command or session (Criar comando ou sessão).
4. Em Name (Nome), insira um nome descritivo para o documento.
5. Em Document Type (Tipo de documento), escolha Session document (Documento de sessão).
6. Insira o conteúdo do documento que define o comando que um usuário pode executar em uma sessão do Session Manager usando JSON ou YAML, conforme mostrado no exemplo a seguir.

   YAML

```
---
schemaVersion: '1.0'
description: Document to view a log file on a Linux instance
sessionType: InteractiveCommands
parameters:
  logpath:
    type: String
    description: The log file path to read.
    default: "/var/log/amazon/ssm/amazon-ssm-agent.log"
    allowedPattern: "^[a-zA-Z0-9-_/]+(.log)$"
properties:
  linux:
    commands: "tail -f {{ logpath }}"
    runAsElevated: true
```

   JSON

```
{
    "schemaVersion": "1.0",
    "description": "Document to view a log file on a Linux instance",
    "sessionType": "InteractiveCommands",
    "parameters": {
        "logpath": {
            "type": "String",
            "description": "The log file path to read.",
```

```
            "default": "/var/log/amazon/ssm/amazon-ssm-agent.log",
            "allowedPattern": "^[a-zA-Z0-9-_/]+(.log)$"
        }
    },
    "properties": {
        "linux": {
            "commands": "tail -f {{ logpath }}",
            "runAsElevated": true
        }
    }
}
```

7. Escolha Criar documento.

Restringir acesso a comandos em uma sessão (linha de comando)

Antes de começar

Caso ainda não tenha feito isso, instale e configure a AWS Command Line Interface (AWS CLI) ou o Ferramentas da AWS para PowerShell. Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI e Instalar o Ferramentas da AWS para PowerShell.

Para restringir os comandos que um usuário pode executar em uma sessão do Session Manager (linha de comando)

1. Crie um arquivo JSON ou YAML para o conteúdo do documento que define o comando que um usuário pode executar em uma sessão do Session Manager, conforme mostrado no exemplo a seguir.

   YAML

```
---
schemaVersion: '1.0'
description: Document to view a log file on a Linux instance
sessionType: InteractiveCommands
parameters:
  logpath:
    type: String
    description: The log file path to read.
    default: "/var/log/amazon/ssm/amazon-ssm-agent.log"
    allowedPattern: "^[a-zA-Z0-9-_/]+(.log)$"
properties:
```

```
  linux:
    commands: "tail -f {{ logpath }}"
    runAsElevated: true
```

JSON

```
{
    "schemaVersion": "1.0",
    "description": "Document to view a log file on a Linux instance",
    "sessionType": "InteractiveCommands",
    "parameters": {
        "logpath": {
            "type": "String",
            "description": "The log file path to read.",
            "default": "/var/log/amazon/ssm/amazon-ssm-agent.log",
            "allowedPattern": "^[a-zA-Z0-9-_/]+(.log)$"
        }
    },
    "properties": {
        "linux": {
            "commands": "tail -f {{ logpath }}",
            "runAsElevated": true
        }
    }
}
```

2. Execute os seguintes comandos para criar um documento do SSM usando seu conteúdo que define o comando que um usuário pode executar em uma sessão do Session Manager.

   Linux & macOS

   ```
   aws ssm create-document \
       --content file://path/to/file/documentContent.json \
       --name "exampleAllowedSessionDocument" \
       --document-type "Session"
   ```

   Windows

   ```
   aws ssm create-document ^
       --content file://C:\path\to\file\documentContent.json ^
       --name "exampleAllowedSessionDocument" ^
       --document-type "Session"
   ```

PowerShell

```
$json = Get-Content -Path "C:\path\to\file\documentContent.json" | Out-String
New-SSMDocument `
    -Content $json `
    -Name "exampleAllowedSessionDocument" `
    -DocumentType "Session"
```

Parâmetros de comando interativos e a AWS CLI

Usando a AWS CLI, os parâmetros de comandos interativos podem ser fornecidos de várias maneiras. Dependendo do sistema operacional (SO) da máquina cliente que você usa para se conectar aos nós gerenciados com a AWS CLI, a sintaxe fornecida para comandos que contêm caracteres especiais ou de escape poderá ser diferente. Os exemplos a seguir mostram algumas das maneiras de fornecer parâmetros de comandos ao usar a AWS CLI e como lidar com caracteres especiais ou de escape.

Os parâmetros armazenados em Parameter Store podem ser referenciados na AWS CLI para seus parâmetros de comandos conforme mostrado no exemplo a seguir.

Linux & macOS

```
aws ssm start-session \
    --target instance-id \
    --document-name MyInteractiveCommandDocument \
    --parameters '{"command":["{{ssm:mycommand}}"]}'
```

Windows

```
aws ssm start-session ^
    --target instance-id ^
    --document-name MyInteractiveCommandDocument ^
    --parameters '{"command":["{{ssm:mycommand}}"]}'
```

O exemplo a seguir mostra como é possível usar uma sintaxe abreviada com a AWS CLI para passar parâmetros.

Linux & macOS

```
aws ssm start-session \
    --target instance-id \
    --document-name MyInteractiveCommandDocument \
    --parameters command="ifconfig"
```

Windows

```
aws ssm start-session ^
    --target instance-id ^
    --document-name MyInteractiveCommandDocument ^
    --parameters command="ipconfig"
```

Você também pode fornecer parâmetros em JSON como mostrado no exemplo a seguir.

Linux & macOS

```
aws ssm start-session \
    --target instance-id \
    --document-name MyInteractiveCommandDocument \
    --parameters '{"command":["ifconfig"]}'
```

Windows

```
aws ssm start-session ^
    --target instance-id ^
    --document-name MyInteractiveCommandDocument ^
    --parameters '{"command":["ipconfig"]}'
```

Os parâmetros também podem ser armazenados em um arquivo JSON e fornecidos para a AWS CLI como mostrado no exemplo a seguir. Para obter mais informações sobre como usar parâmetros da AWS CLI em um arquivo, consulte Carregar parâmetros da AWS CLI em um arquivo no Manual do usuário do AWS Command Line Interface.

```
{
    "command": [
        "my command"
    ]
```

```
}
```

Linux & macOS

```
aws ssm start-session \
    --target instance-id \
    --document-name MyInteractiveCommandDocument \
    --parameters file://complete/path/to/file/parameters.json
```

Windows

```
aws ssm start-session ^
    --target instance-id ^
    --document-name MyInteractiveCommandDocument ^
    --parameters file://complete/path/to/file/parameters.json
```

Também é possível gerar um esqueleto da AWS CLI de um arquivo de entrada JSON conforme mostrado no exemplo a seguir. Para obter mais informações sobre a geração de esqueletos da AWS CLI com base em arquivos de entrada JSON, consulte Gerar um esqueleto da AWS CLI e parâmetros de entrada usando um arquivo de entrada JSON ou YAML no Guia do usuário da AWS Command Line Interface.

```
{
    "Target": "instance-id",
    "DocumentName": "MyInteractiveCommandDocument",
    "Parameters": {
        "command": [
            "my command"
        ]
    }
}
```

Linux & macOS

```
aws ssm start-session \
    --cli-input-json file://complete/path/to/file/parameters.json
```

Windows

```
aws ssm start-session ^
```

```
    --cli-input-json file://complete/path/to/file/parameters.json
```

Para escapar caracteres dentro das aspas, adicione barras invertidas adicionais aos caracteres de escape, conforme mostrado no exemplo a seguir.

Linux & macOS

```
aws ssm start-session \
    --target instance-id \
    --document-name MyInteractiveCommandDocument \
    --parameters '{"command":["printf \"abc\\\\tdef\""]}'
```

Windows

```
aws ssm start-session ^
    --target instance-id ^
    --document-name MyInteractiveCommandDocument ^
    --parameters '{"command":["printf \"abc\\\\tdef\""]}'
```

Para obter informações sobre o uso de aspas com parâmetros formatados na AWS CLI, consulte Usar aspas com strings na AWS CLI, no Manual do usuário da AWS Command Line Interface.

Exemplos de políticas do IAM para comandos interativos

Você pode criar políticas do IAM que permitem que os usuários acessem somente os documentos de `Session` definidos por você. Isso restringe os comandos que um usuário pode executar em uma sessão do Session Manager apenas aos comandos definidos em seus documentos personalizados do SSM do tipo `Session`.

Permitir que um usuário execute um comando interativo em um único nó gerenciado

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ssm:StartSession",
         "Resource":[
```

```
            "arn:aws:ec2:us-east-1:444455556666:instance/i-02573cafcfEXAMPLE",
            "arn:aws:ssm:us-east-1:444455556666:document/allowed-session-
document"
         ]
      },
      {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
      }
   ]
}
```

Permitir que um usuário execute um comando interativo em todos os nós gerenciados

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ssm:StartSession",
         "Resource":[
            "arn:aws:ec2:us-east-1:444455556666:instance/*",
            "arn:aws:ssm:us-east-1:444455556666:document/allowed-session-
document"
         ]
      },
      {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
      }
   ]
}
```

Permitir que um usuário execute vários comandos interativos em todos os nós gerenciados

JSON

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

```
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ssm:StartSession",
         "Resource":[
            "arn:aws:ec2:us-east-1:444455556666:instance/*",
            "arn:aws:ssm:us-east-1:444455556666:document/allowed-session-
document",
            "arn:aws:ssm:us-east-1:444455556666:document/allowed-session-
document-2"
         ]
      },
      {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
      }
   ]
}
```

Etapa 6: (Opcional) Usar o AWS PrivateLink para configurar um endpoint da VPC para o Session Manager

Você pode melhorar o procedimento de segurança dos nós gerenciados configurando o AWS Systems Manager para usar um endpoint de nuvem privada virtual (VPC) da interface. Os endpoints da interface são habilitados pelo AWS PrivateLink, uma tecnologia que permite que você acesse privadamente APIs do Amazon Elastic Compute Cloud (Amazon EC2) e do Systems Manager usando endereços IP privados.

O AWS PrivateLink limita todo o tráfego de rede entre os nós gerenciados, o Systems Manager, o Amazon EC2 e a rede da Amazon. (Os nós gerenciados não têm acesso à Internet.) Além disso, você não precisa de um Internet gateway, de um dispositivo NAT ou de um gateway privado virtual.

Para obter informações sobre como criar um endpoint da VPC, consulte Melhorar a segurança das instâncias do EC2 usando endpoints da VPC para o Systems Manager.

A alternativa ao uso de um endpoint da VPC é permitir o acesso à Internet de saída em seus nós gerenciados. Nesse caso, os nós gerenciados também devem permitir tráfego de saída HTTPS (porta 443) para os seguintes endpoints:

- `ec2messages.region.amazonaws.com`

- `ssm.region.amazonaws.com`
- `ssmmessages.region.amazonaws.com`

O Systems Manager usa o último desses endpoints, `ssmmessages.region.amazonaws.com`, para fazer chamadas do SSM Agent para o serviço Session Manager na nuvem.

Para usar recursos opcionais como criptografia AWS Key Management Service (AWS KMS), transmissão de logs para o Amazon CloudWatch Logs (CloudWatch Logs) e envio de logs para o Amazon Simple Storage Service (Amazon S3), você deve permitir o tráfego de saída HTTPS (porta 443) para os seguintes endpoints:

- `kms.region.amazonaws.com`
- `logs.region.amazonaws.com`
- `s3.region.amazonaws.com`

Para obter mais informações sobre os endpoints necessários para o Systems Manager, consulte Referência: ec2messages, ssmmessages e outras operações da API.

Etapa 7: (Opcional) Ativar ou desativar permissões administrativas da conta ssm-user.

A partir da versão 2.3.50.0 do SSM Agent do AWS Systems Manager, o agente cria uma conta de usuário local chamada `ssm-user` e a adiciona ao `/etc/sudoers` (Linux e macOS) ou ao grupo de Administradores (Windows). Em versões do agente anteriores a 2.3.612.0, a conta é criada na primeira vez que o SSM Agent é iniciado ou reiniciado após a instalação. Na versão 2.3.612.0 e posteriores, a conta `ssm-user` é criada na primeira vez que uma sessão é iniciada em um nó. Esse `ssm-user` é o usuário padrão do sistema operacional (SO) quando uma sessão do AWS Systems Manager Session Manager é iniciada. A versão 2.3.612.0 do SSM Agent foi lançada em 8 de maio de 2019.

Para impedir que os usuários do Session Manager executem comandos de administrador em um nó, você pode atualizar as permissões da conta `ssm-user`. Você também pode restaurar essas permissões depois de serem removidas.

Tópicos

- Gerenciar as permissões da conta sudo do ssm-user no Linux e no macOS
- Gerenciar as permissões da conta de administrador ssm-user no Windows Server

Gerenciar as permissões da conta sudo do ssm-user no Linux e no macOS

Use um dos procedimentos a seguir para ativar ou desativar as permissões sudo da conta ssm-user em nós gerenciados do Linux e do macOS.

Usar o Run Command para modificar permissões sudo de ssm-user (console)

- Use o procedimento em Executar comandos no console com os seguintes valores:
  - Para Command document (Documento de comando), escolha `AWS-RunShellScript`.
  - Para remover o acesso sudo, na área Command parameters (Parâmetros de comando), cole o trecho a seguir na caixa Commands (Comandos).

    ```
    cd /etc/sudoers.d
    echo "#User rules for ssm-user" > ssm-agent-users
    ```

    - ou -

    Para restaurar o acesso sudo, na área Command parameters (Parâmetros de comando), cole o trecho a seguir na caixa Commands (Comandos).

    ```
    cd /etc/sudoers.d
    echo "ssm-user ALL=(ALL) NOPASSWD:ALL" > ssm-agent-users
    ```

Usar a linha de comando para modificar permissões sudo de ssm-user (AWS CLI)

1. Conecte-se ao nó gerenciado e execute o seguinte comando:

   ```
   sudo -s
   ```

2. Altere o diretório de trabalho usando o seguinte comando:

   ```
   cd /etc/sudoers.d
   ```

3. Abra o arquivo chamado `ssm-agent-users` para edição.
4. Para remover o acesso sudo, exclua a linha a seguir.

   ```
   ssm-user ALL=(ALL) NOPASSWD:ALL
   ```

- ou -

Para restaurar o acesso sudo, adicione a linha a seguir.

```
ssm-user ALL=(ALL) NOPASSWD:ALL
```

5. Salve o arquivo.

Gerenciar as permissões da conta de administrador ssm-user no Windows Server

Use um dos procedimentos a seguir para ativar ou desativar as permissões de Administrador da conta ssm-user em nós gerenciados do Windows Server.

Usar o Run Command para modificar permissões de Administrador (console)

- Use o procedimento em Executar comandos no console com os seguintes valores:

  Para Command document (Documento de comando), escolha `AWS-RunPowerShellScript`.

  Para remover o acesso de administrador, na área Command parameters (Parâmetros de comando), cole o trecho a seguir na caixa Commands (Comandos).

```
net localgroup "Administrators" "ssm-user" /delete
```

  - ou -

  Para restaurar o acesso de administrador, na área Command parameters (Parâmetros de comando), cole o trecho a seguir na caixa Commands (Comandos).

```
net localgroup "Administrators" "ssm-user" /add
```

Use a janela do prompt de comando ou o PowerShell para modificar permissões de administrador

1. Conecte-se ao nó gerenciado e abra o PowerShell ou a janela do prompt de comando.
2. Para remover o acesso administrativo, execute o comando a seguir.

```
net localgroup "Administrators" "ssm-user" /delete
```

- ou -

Para restaurar o acesso administrativo, execute o comando a seguir.

```
net localgroup "Administrators" "ssm-user" /add
```

Usar o console do Windows para modificar permissões de Administrador

1. Conecte-se ao nó gerenciado e abra o PowerShell ou a janela do prompt de comando.
2. Na linha de comando, execute `lusrmgr.msc` para abrir o console Local Users and Groups (Usuários locais e grupos).
3. Abra o diretório Users (Usuários) e, em seguida, abra ssm-user.
4. Na guia Member Of (Membro do), faça o seguinte:
   - Para remover o acesso administrativo, selecione Administrators (Administradores)e, em seguida, escolha Remove (Remover).

     - ou -

     Para restaurar o acesso administrativo, digite **`Administrators`** na caixa de texto e, em seguida, escolha Add (Adicionar).
5. Escolha OK.

Etapa 8: (Opcional) Permitir e controlar permissões de conexões de SSH por meio do Session Manager

Você pode permitir que os usuários da sua Conta da AWS usem a AWS Command Line Interface (AWS CLI) para estabelecer conexões Secure Shell (SSH) com nós gerenciados que usam o Session Manager do AWS Systems Manager. Os usuários que se conectam usando SSH também podem copiar arquivos entre suas máquinas locais e os nós gerenciados usando o Secure Copy Protocol (SCP). Você pode usar essa funcionalidade para se conectar aos nós gerenciados sem abrir portas de entrada nem manter bastion hosts.

Depois de permitir conexões SSH, você pode usar as políticas do AWS Identity and Access Management (IAM) para permitir ou negar explicitamente que usuários, grupos ou funções façam conexões de SSH usando o Session Manager.

> **Note**
>
> O registro em log não está disponível para sessões do Session Manager que se conectam por meio de encaminhamento de portas ou SSH. Isso ocorre porque o SSH criptografa todos os dados da sessão e o Session Manager serve apenas como um túnel para conexões SSH.

Tópicos

- Permitir conexões de SSH para o Session Manager
- Controlar permissões do usuário para conexões SSH por meio do Session Manager

Permitir conexões de SSH para o Session Manager

Use as etapas a seguir para permitir conexões de SSH por meio do Session Manager em um nó gerenciado.

Para permitir conexões de SSH para o Session Manager

1. Nesse nó gerenciado, para a qual você quer permitir conexões SSH, faça o seguinte:

   - Certifique-se de que o SSH está em execução em um nó gerenciado. (Você pode fechar portas de entrada em um nó.)
   - Certifique-se de que o SSM Agent versão 2.3.672.0 ou posterior esteja instalado em seu nó gerenciado.

     Para obter informações sobre como instalar ou atualizar o SSM Agent em um nó gerenciado, consulte os seguintes tópicos:

     - Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Windows Server.
     - Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux
     - Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para macOS
     - Como instalar o SSM Agent em nós híbridos do Windows
     - Como instalar o SSM Agent em nós híbridos do Linux

> **Note**
>
> Para usar o Session Manager com servidores on-premises, dispositivos de borda e máquinas virtuais (VMs) que você ativou como nós gerenciados, você deve usar o nível de instâncias avançadas. Para obter mais informações sobre instâncias avançadas, consulte Configurar níveis de instâncias.

2. Na máquina local da qual você deseja se conectar a um nó gerenciado usando SSH, faça o seguinte:

   - Certifique-se de que a versão 1.1.23.0 ou posterior do plugin do Session Manager esteja instalada.

     Para obter informações sobre como instalar o plugin do Session Manager, consulte Instalar o plug-in do Session Manager para a AWS CLI.

   - Atualize o arquivo de configuração do SSH para permitir a execução de um comando proxy que inicia uma sessão do Session Manager e transferir todos os dados por meio da conexão.

     Linux e macOS

     > **Tip**
     >
     > O arquivo de configuração do SSH normalmente está localizado em `~/.ssh/config`.

     Adicione o seguinte ao arquivo de configuração na máquina local:

     ```
     # SSH over Session Manager
     Host i-* mi-*
         ProxyCommand sh -c "aws ssm start-session --target %h --document-name AWS-StartSSHSession --parameters 'portNumber=%p'"
         User ec2-user
     ```

     Windows

> **Tip**
>
> O arquivo de configuração do SSH normalmente está localizado em `C:\Users\<username>\.ssh\config`.

Adicione o seguinte ao arquivo de configuração na máquina local:

```
# SSH over Session Manager
Host i-* mi-*
    ProxyCommand C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe "aws
 ssm start-session --target %h --document-name AWS-StartSSHSession --parameters
 portNumber=%p"
```

- Crie ou verifique se você tem um certificado de Privacy Enhanced Mail (um arquivo PEM) ou, no mínimo, uma chave pública para usar ao estabelecer conexões com os nós gerenciados. Essa deve ser uma chave que já esteja associada ao nó gerenciado. As permissões do seu arquivo de chave privada devem ser definidas para que somente você possa lê-lo. Você pode usar o seguinte comando para definir as permissões do arquivo de chave privada para que somente você possa lê-lo.

  ```
  chmod 400 <my-key-pair>.pem
  ```

  Por exemplo, para uma instância do Amazon Elastic Compute Cloud (Amazon EC2), o arquivo de par de chaves que você criou ou selecionou quando criou a instância. (Você especifica o caminho para o certificado ou a chave como parte do comando para iniciar uma sessão. Para obter informações sobre começar uma seção usando o SSH, consulte Iniciar uma sessão (SSH).)

Controlar permissões do usuário para conexões SSH por meio do Session Manager

Depois de habilitar conexões SSH por meio do Session Manager em um nó gerenciado, você poderá usar políticas do IAM para permitir ou recusar que usuários, grupos ou funções estabeleçam conexões SSH por meio do Session Manager.

Para usar uma política do IAM para permitir conexões SSH por meio do Session Manager

- Use uma das seguintes opções:

- Opção 1: abra o console do IAM em https://console.aws.amazon.com/iam/.

  No painel de navegação, escolha Policies (Políticas) e atualize a política de permissões para o usuário ou função ao qual você deseja permitir iniciar conexões SSH por meio do Session Manager.

  Por exemplo, adicione o elemento a seguir à política do Quickstart que você criou em Políticas do usuário final do Quickstart para o Session Manager. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

  JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ssm:StartSession",
            "Resource": [
                "arn:aws:ec2:us-east-1:111122223333:instance/instance-id",
                "arn:aws:ssm:*:*:document/AWS-StartSSHSession"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "ssmmessages:OpenDataChannel",
            "Resource": "arn:aws:ssm:*:*:session/${aws:userid}-*"
        }
    ]
}
```

- Opção 2: anexe uma política em linha a uma política de usuário usando o AWS Management Console, a AWS CLI ou a API da AWS.

  Usando o método de sua escolha, anexe a declaração de política na Opção 1 à política de um usuário, grupo ou função da AWS.

  Para obter informações, consulte Adicionar e remover permissões de identidade do IAM no Manual do usuário do IAM.

Para usar uma política do IAM para negar conexões SSH por meio do Session Manager

- Use uma das seguintes opções:

  - Opção 1: abra o console do IAM em https://console.aws.amazon.com/iam/. No painel de navegação, escolha Policies (Políticas) e, então, atualize a política de permissões para que o usuário ou a função não possa iniciar sessões do Session Manager.

    Por exemplo, adicione o elemento a seguir à política do Quickstart que você criou em Políticas do usuário final do Quickstart para o Session Manager.

    JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "ssm:StartSession",
            "Resource": "arn:aws:ssm:*:*:document/AWS-StartSSHSession"
        },
        {
            "Effect": "Allow",
            "Action": "ssmmessages:OpenDataChannel",
            "Resource": "arn:aws:ssm:*:*:session/${aws:userid}-*"
        }
    ]
}
```

  - Opção 2: anexe uma política em linha a uma política de usuário usando o AWS Management Console, a AWS CLI ou a API da AWS.

    Usando o método de sua escolha, anexe a declaração de política na Opção 1 à política de um usuário, grupo ou função da AWS.

    Para obter informações, consulte Adicionar e remover permissões de identidade do IAM no Manual do usuário do IAM.

Trabalhar com o Session Manager

Você pode usar o console do AWS Systems Manager, o console do Amazon Elastic Compute Cloud (Amazon EC2) ou a AWS Command Line Interface (AWS CLI) para iniciar sessões que conectem

você aos nós gerenciados do Amazon EC2 aos quais o administrador do sistema concedeu acesso a você usando políticas do AWS Identity and Access Management (IAM). Dependendo das suas permissões, você também poderá visualizar as informações sobre as sessões, retomar sessões inativas que não tenham expirado e encerrar sessões. Depois que uma sessão é estabelecida, ela não é afetada pela duração da sessão de perfil do IAM. Para obter informações sobre como limitar a duração da sessão com Session Manager, consulte Especificar um valor de tempo limite de sessão ociosa e Especifique a duração máxima da sessão.

Para obter mais informações sobre sessões, consulte O que é uma sessão?

Tópicos

- Instalar o plug-in do Session Manager para a AWS CLI
- Iniciar uma sessão
- Encerrar uma sessão
- Visualizar o histórico da sessão

Instalar o plug-in do Session Manager para a AWS CLI

Para iniciar sessões do Session Manager com seus nós gerenciados usando a AWS Command Line Interface (AWS CLI), é necessário instalar o plug-in do Session Manager em sua máquina local. O plug-in pode ser instalado em versões com suporte do Microsoft Windows Server, macOS, Linux e Ubuntu Server.

> **Note**
>
> Para usar o plug-in do Session Manager, é necessário ter a AWS CLI versão 1.16.12 ou posterior instalada na máquina local. Para obter mais informações, consulte Instalar ou atualizar a versão mais recente da AWS Command Line Interface.

Tópicos

- Versão mais recente e histórico de lançamentos do plugin Session Manager
- Instalar o plugin do Session Manager no Windows
- Instalar o plugin do Session Manager no macOS
- Instalar o plug-in Session Manager no Linux
- Verifique a instalação do plugin do Session Manager

- Plugin do Session Manager no GitHub
- (Opcional) Ative o plugin de registro em log do Session Manager

Versão mais recente e histórico de lançamentos do plugin Session Manager

Suas máquinas locais devem estar executando uma versão com suporte para o plugin Session Manager. A versão mínima compatível atual é 1.1.17.0. Se você estiver executando uma versão anterior, as operações do Session Manager podem não ser bem-sucedidas.

Para ver se você tem a versão mais recente, execute o seguinte comando no AWS CLI.

> **Note**
>
> O comando retornará resultados somente se o plugin estiver localizado no diretório de instalação padrão do tipo de sistema operacional. Você também pode verificar a versão no conteúdo do arquivo do `VERSION` no diretório onde você instalou o plugin.

```
session-manager-plugin --version
```

A tabela a seguir lista todas as versões do plugin do Session Manager, bem como os recursos e aprimoramentos incluídos em cada versão.

> **Important**
>
> Recomendamos que você sempre execute a versão mais recente. A versão mais recente inclui aprimoramentos que melhoram a experiência de uso do plug-in.

Versão	Data de lançamento	Detalhes
1.2.707.0	6 de fevereiro de 2025	Aprimoramento: atualização da versão do Go para 1.23 no Dockerfile. Atualização da etapa de configuração da versão no arquivo README.

Versão	Data de lançamento	Detalhes
1.2.694.0	20 de novembro de 2024	Correção de bug: uma alteração que adicionava credenciais às solicitações do OpenDataChannel foi revertida.
1.2.688.0	6 de novembro de 2024	Esta versão de runtime foi descontinuada em 20/11/2024. Melhorias: • Adição de credenciais às solicitações do OpenDataChannel. • Atualização dos pacotes dependentes `testify` e `objx`.
1.2.677.0	10 de outubro de 2024	Aperfeiçoamento: adição de suporte para passar a versão do plug-in com solicitações de OpenDataChannel.
1.2.650.0	2 de julho de 2024	Aprimoramento: o aws-sdk-go foi atualizado para 1.54.10. Correção de bug: comentários reformatados para gofmt check.
1.2.633.0	30 de maio de 2024	Aprimoramento: o Dockerfile foi atualizado para usar uma imagem do Amazon Elastic Container Registry (Amazon ECR).
1.2.553.0	10 de janeiro de 2024	Aprimoramento: atualização do aws-sdk-go e dos pacotes Golang dependentes.
1.2.536.0	4 de dezembro de 2023	Aprimoramento: adicionado suporte para transmitir uma resposta da API StartSession como variável de ambiente a session-manager-plugin.
1.2.497.0	1º de agosto de 2023	Aprimoramento: o Go SDK foi atualizado para v1.44.302.
1.2.463.0	15 de março de 2023	Aprimoramento: adição de suporte a Mac with Apple silicon para Apple Mac (M1) no instalador do pacote macOS e no instalador assinado.

Versão	Data de lançamento	Detalhes
1.2.398.0	14 de outubro de 2022	Aprimoramento: compatibilidade com a versão 1.17 do golang. Atualizar o programa de execução do session-manager-plugin padrão para macOS para usar o python3. Atualizar o caminho de importação de SSMCLI para session-manager-plugin.
1.2.339.0	16 de junho de 2022	Correção de bug: corrigir o tempo limite da sessão ociosa para sessões de porta.
1.2.331.0	27 de maio de 2022	Correção de bug: corrigir sessões de porta fechando prematuramente quando o servidor local não se conecta antes do tempo limite se esgotar.
1.2.323.0	19 de maio de 2022	Correção de bug: desativar o smux keep live para usar o recurso de tempo limite de sessão ociosa.
1.2.312.0	31 de março de 2022	Aprimoramento: Suporta mais tipos de carga útil de mensagem de saída.
1.2.295.0	12 de janeiro de 2022	Correção de bugs: sessões suspensas causadas pelo reenvio de dados de fluxo do cliente quando o agente se torna inativo e registros incorretos de mensagens `start_publication` e `pause_publication` .
1.2.279.0	27 de outubro de 2021	Aprimoramento: compactação em zip para a plataforma Windows.
1.2.245.0	19 de agosto de 2021	Melhoria: atualize o `aws-sdk-go` para a versão mais recente (v1.40.17) para suporte a AWS IAM Identity Center.
1.2.234.0	26 de julho de 2021	Correção de bugs: administre o encerramento abrupto da sessão no tipo de sessão interativa.
1.2.205.0	10 de junho de 2021	Melhoria: adição de suporte para o instalador do macOS

Versão	Data de lançamento	Detalhes
1.2.54.0	29 de janeiro de 2021	Melhoria: inclusão do suporte a sessões em execução no modo de execução NonInteractiveCommands.
1.2.30.0	24 de novembro de 2020	Melhoria: (somente sessões de encaminhamento de porta) melhoria geral da performance.
1.2.7.0	15 de outubro de 2020	Melhoria: (somente sessões de encaminhamento de porta) latência reduzida e melhoria geral da performance.
1.1.61.0	17 de abril de 2020	Aprimoramento: adição de suporte a ARM para Linux e Ubuntu Server.
1.1.54.0	6 de janeiro de 2020	Correção de erros: administre a condição de disputa dos pacotes que estão sendo descartados quando o plugin Session Manager ainda não estiver pronto.
1.1.50.0	19 de novembro de 2019	Melhoria: adição de suporte para encaminhar uma porta para um soquete unix local.
1.1.35.0	7 de novembro de 2019	Melhoria: (sessões de encaminhamento de porta somente) envie um comando TerminateSession para o SSM Agent quando o usuário local pressionar `Ctrl+C`.
1.1.33.0	26 de setembro de 2019	Melhoria: (somente sessões de encaminhamento de porta) envio de um sinal de desconexão para o servidor quando o cliente interromper a conexão TCP.
1.1.31.0	6 de setembro de 2019	Melhoria: atualização para manter a sessão de encaminhamento de porta aberta até que o servidor remoto encerre a conexão.
1.1.26.0	30 de julho de 2019	Melhoria: atualização para limitar a throughput de dados durante uma sessão.
1.1.23.0	9 de julho de 2019	Melhoria: adição de suporte à execução de sessões SSH usando o Session Manager.

Versão	Data de lançamento	Detalhes
1.1.17.0	4 de abril de 2019	Melhoria: adição de suporte para a criptografia adicional de dados da sessão usando o AWS Key Management Service (AWS KMS).
1.0.37.0	20 de setembro de 2018	Melhoria: correção de erros na versão do Windows.
1.0.0.0	11 de setembro de 2018	Versão inicial do plugin do Session Manager.

Instalar o plugin do Session Manager no Windows

Você pode instalar o plug-in do Session Manager no Windows Vista ou em uma versão posterior usando o instalador independente.

Quando as atualizações são lançadas, repita o processo de instalação para instalar a versão mais recente do plugin do Session Manager.

> **Note**
>
> Observe as seguintes informações:
>
> - O instalador do plug-in do Session Manager necessita de direitos de administrador para instalar o plug-in.
> - Para obter melhores resultados, recomendamos iniciar sessões em clientes Windows usando a aplicação o Windows PowerShell versão 5 ou posterior. É possível também usar o shell do Command no Windows 10. O plug-in do Session Manager é compatível apenas com o PowerShell e o shell do Command. Ferramentas da linha de comando de terceiros podem não ser compatíveis com o plugin.

Para instalar o plugin do Session Manager usando o instalador EXE

1. Baixe o instalador usando a seguinte URL:

```
https://s3.amazonaws.com/session-manager-downloads/plugin/latest/windows/
SessionManagerPluginSetup.exe
```

Como alternativa, você pode baixar uma versão compactada do instalador usando o seguinte URL:

```
https://s3.amazonaws.com/session-manager-downloads/plugin/latest/windows/
SessionManagerPlugin.zip
```

2. Execute o instalador baixado e siga as instruções na tela. Se você baixou a versão compactada do instalador, você deve descompactar o instalador primeiro.

   Deixe a caixa do local de instalação em branco para instalar o plugin no diretório padrão.

   - `%PROGRAMFILES%\Amazon\SessionManagerPlugin\bin\`

3. Verifique se a instalação foi bem-sucedida. Para mais informações, consulte Verifique a instalação do plugin do Session Manager.

> **Note**
>
> Se o Windows não consegue encontrar o executável, pode ser necessário abrir o prompt de comando ou adicionar o diretório de instalação a sua variável de ambiente PATH manualmente. Para obter informações, consulte o tópico sobre solução de problemas O plug-in Session Manager não é adicionado automaticamente ao caminho de linha de comando (Windows).

Instalar o plugin do Session Manager no macOS

Escolha um dos tópicos a seguir para instalar o plug-in do Session Manager no macOS.

> **Note**
>
> O instalador assinado é um arquivo `.pkg` assinado. O pacote de instalador usa um arquivo `.zip`. Depois que o arquivo for descompactado, você poderá instalar o plug-in usando o binário.

Instale o plugin Session Manager no macOS usando o instalador assinado.

Esta seção descreve como instalar o plug-in do Session Manager no macOS usando o instalador assinado.

Para instalar o plugin do Session Manager usando o instalador assinado (macOS)

1. Baixe o instalador assinado.

   x86_64

   ```
   curl "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/mac/
   session-manager-plugin.pkg" -o "session-manager-plugin.pkg"
   ```

   Mac com chip Apple

   ```
   curl "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/
   mac_arm64/session-manager-plugin.pkg" -o "session-manager-plugin.pkg"
   ```

2. Execute os comandos de instalação. Se o comando falhar, verifique se a pasta `/usr/local/bin` existe. Se ela não existir, crie-a e execute o comando novamente.

   ```
   sudo installer -pkg session-manager-plugin.pkg -target /
   sudo ln -s /usr/local/sessionmanagerplugin/bin/session-manager-plugin /usr/local/
   bin/session-manager-plugin
   ```

3. Verifique se a instalação foi bem-sucedida. Para mais informações, consulte Verifique a instalação do plugin do Session Manager.

Instalar o plugin do Session Manager no macOS

Esta seção descreve como instalar o plug-in do Session Manager no macOS usando o instalador associado.

> **Important**
>
> Observe as seguintes informações importantes:
>
> - Por padrão, o instalador requer acesso sudo para executar porque o script instala o plug-in no diretório de sistema `/usr/local/sessionmanagerplugin`. Se não desejar instalar

o plug-in usando sudo, atualize manualmente o script do instalador para instalar o plug-in em um diretório que não exija acesso sudo.

- O pacote de instalador fornecido não é compatível com a instalação em caminhos com espaços.

Para instalar o plugin Session Manager usando o pacote de instalador (macOS)

1. Baixe o instalador em pacote.

   x86_64

   ```
   curl "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/mac/sessionmanager-bundle.zip" -o "sessionmanager-bundle.zip"
   ```

   Mac com chip Apple

   ```
   curl "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/mac_arm64/sessionmanager-bundle.zip" -o "sessionmanager-bundle.zip"
   ```

2. Descompacte o pacote.

   ```
   unzip sessionmanager-bundle.zip
   ```

3. Execute o comando de instalação.

   ```
   sudo ./sessionmanager-bundle/install -i /usr/local/sessionmanagerplugin -b /usr/local/bin/session-manager-plugin
   ```

   Note

   O plugin exige Python 2.6.5 ou posterior ou Python 3.3 ou posterior. Por padrão, o script de instalação é executado sob a versão padrão do sistema do Python. Se uma versão diferente do Python estiver instalada, mas deseja usá-la para instalar o plugin do Session Manager, execute o script de instalação com essa versão por caminho absoluto para o Python executável. Veja um exemplo a seguir.

```
sudo /usr/local/bin/python3.8 sessionmanager-bundle/install -i /usr/local/
sessionmanagerplugin -b /usr/local/bin/session-manager-plugin
```

O instalador instala o plugin do Session Manager em `/usr/local/sessionmanagerplugin` e cria o symlink `session-manager-plugin` no diretório `/usr/local/bin`. Isso elimina a necessidade de especificar o diretório de instalação na variável `$PATH` do usuário.

Para ver uma explicação das opções `-i` e `-b`, use a opção `-h`.

```
./sessionmanager-bundle/install -h
```

4. Verifique se a instalação foi bem-sucedida. Para mais informações, consulte Verifique a instalação do plugin do Session Manager.

> **Note**
>
> Para desinstalar o plug-in, execute os comandos a seguir, um de cada vez.
>
> ```
> sudo rm -rf /usr/local/sessionmanagerplugin
> ```
>
> ```
> sudo rm /usr/local/bin/session-manager-plugin
> ```

Instalar o plug-in Session Manager no Linux

Esta seção inclui informações sobre como verificar a assinatura do pacote do instalador do plug-in do Session Manager e instalá-lo nas seguintes distribuições Linux:

- Amazon Linux 2
- AL2023
- RHEL
- Debian
- Ubuntu

Tópicos

- Verificar a assinatura do plug-in Session Manager
- Instalar o plug-in do Session Manager no Amazon Linux 2, no Amazon Linux 2023 e nas distribuições do Red Hat Enterprise Linux
- Instalar o plugin do Session Manager no Debian Server e no Ubuntu Server

Verificar a assinatura do plug-in Session Manager

Os pacotes do instalador RPM e Debian do plug-in do Session Manager para instâncias Linux são assinados criptograficamente. É possível usar uma chave pública para verificar se o binário e o pacote do plug-in são originais e não modificados. Se o arquivo for alterado ou danificado, a verificação falhará. A assinatura do pacote do instalador pode ser verificada com a ferramenta GNU Privacy Guard (GPG). As informações a seguir aplicam-se ao plug-in do Session Manager versões 1.2.707.0 ou posteriores.

Conclua as etapas a seguir para verificar a assinatura do pacote do instalador do plug-in do Session Manager.

Tópicos

- Etapa 1: baixar o pacote do instalador do plug-in do Session Manager
- Etapa 2: baixar o arquivo de assinatura associado
- Etapa 3: instalar a ferramenta GPG
- Etapa 4: verificar o pacote do instalador do plug-in do Session Manager em um servidor Linux

Etapa 1: baixar o pacote do instalador do plug-in do Session Manager

Baixe o pacote do instalador do plug-in do Session Manager que deseja verificar.

Pacotes do RPM para Amazon Linux 2, AL2023 e RHEL

x86_64

```
curl -o "session-manager-plugin.rpm" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/linux_64bit/session-manager-plugin.rpm"
```

x86

```
curl -o "session-manager-plugin.rpm" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/linux_32bit/session-manager-plugin.rpm"
```

ARM64

```
curl -o "session-manager-plugin.rpm" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/linux_arm64/session-manager-plugin.rpm"
```

Pacotes do Debian e Ubuntu Deb

x86_64

```
curl -o "session-manager-plugin.deb" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/ubuntu_64bit/session-manager-plugin.deb"
```

x86

```
curl -o "session-manager-plugin.deb" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/ubuntu_32bit/session-manager-plugin.deb"
```

ARM64

```
curl -o "session-manager-plugin.deb" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/ubuntu_arm64/session-manager-plugin.deb"
```

Etapa 2: baixar o arquivo de assinatura associado

Após baixar o pacote do instalador, baixe o arquivo de assinatura associado para verificação do pacote. Para fornecer uma camada extra de proteção contra cópia ou uso não autorizados do arquivo binário session-manager-plugin dentro do pacote, também oferecemos assinaturas binárias que podem ser usadas para validar arquivos binários individuais. Você pode optar por usar essas assinaturas binárias com base em suas necessidades de segurança.

Pacotes de assinatura para Amazon Linux 2, AL2023 e RHEL

x86_64

Pacote:

```
curl -o "session-manager-plugin.rpm.sig" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/linux_64bit/session-manager-plugin.rpm.sig"
```

Binário:

```
curl -o "session-manager-plugin.sig" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/linux_64bit/session-manager-plugin.sig"
```

x86

Pacote:

```
curl -o "session-manager-plugin.rpm.sig" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/linux_32bit/session-manager-plugin.rpm.sig"
```

Binário:

```
curl -o "session-manager-plugin.sig" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/linux_32bit/session-manager-plugin.sig"
```

ARM64

Pacote:

```
curl -o "session-manager-plugin.rpm.sig" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/linux_arm64/session-manager-plugin.rpm.sig"
```

Binário:

```
curl -o "session-manager-plugin.sig" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/linux_arm64/session-manager-plugin.sig"
```

Pacotes de assinatura para Debian e Ubuntu Deb

x86_64

Pacote:

```
curl -o "session-manager-plugin.deb.sig" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/ubuntu_64bit/session-manager-plugin.deb.sig"
```

Binário:

```
curl -o "session-manager-plugin.sig" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/ubuntu_64bit/session-manager-plugin.sig"
```

x86

Pacote:

```
curl -o "session-manager-plugin.deb.sig" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/ubuntu_32bit/session-manager-plugin.deb.sig"
```

Binário:

```
curl -o "session-manager-plugin.sig" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/ubuntu_32bit/session-manager-plugin.sig"
```

ARM64

Pacote:

```
curl -o "session-manager-plugin.deb.sig" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/ubuntu_arm64/session-manager-plugin.deb.sig"
```

Binário:

```
curl -o "session-manager-plugin.sig" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/ubuntu_arm64/session-manager-plugin.sig"
```

Etapa 3: instalar a ferramenta GPG

Para verificar a assinatura do plug-in Session Manager, você deve ter a ferramenta GNU Privacy Guard (GPG) instalada no seu sistema. Esse processo de verificação requer o uso do GPG versão 2.1 ou posterior. Para verificar a versão do GPG, execute o seguinte comando:

```
gpg --version
```

Se sua versão do GPG for anterior à 2.1, atualize-a antes de prosseguir com o processo de verificação. Na maioria dos sistemas, é possível atualizar a ferramenta GPG usando seu gerenciador de pacotes. Por exemplo, em sistemas Amazon Linux e RHEL, é possível usar os seguintes comandos:

```
sudo yum update
sudo yum install gnupg2
```

Em sistemas Ubuntu ou Debian, é possível usar os seguintes comandos:

```
sudo apt-get update
sudo apt-get install gnupg2
```

Certifique-se de ter a versão do GPG necessária antes de continuar com o processo de verificação.

Etapa 4: verificar o pacote do instalador do plug-in do Session Manager em um servidor Linux

Use o procedimento a seguir para verificar o pacote do instalador do plug-in do Session Manager em um servidor Linux.

> **Note**
>
> O Amazon Linux 2 não é compatível com a ferramenta GPG versão 2.1 ou superior. Se o procedimento a seguir não funcionar em suas instâncias do Amazon Linux 2, verifique a assinatura em uma plataforma diferente antes de instalá-la em suas instâncias do Amazon Linux 2.

1. Copie a chave pública a seguir e salve-a em um arquivo chamado session-manager-plugin.gpg.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mFIEZ5ERQxMIKoZIzj0DAQcCAwQjuZy+IjFoYg57sLTGhF3aZLBaGpzB+gY6j7Ix
P7NqbpXyjVj8a+dy79gSd64OEaMxUb7vw/jug+CfRXwVGRMNtIBBV1MgU1NNIFNl
c3Npb24gTWFuYWdlciA8c2Vzc2lvbi1tYW5hZ2VyLXBsdWdpbi1zaWduZXJAYW1h
em9uLmNvbT4gKEFXUyBTeXN0ZW1zIE1hbmFnZXIgU2Vzc2lvbiBNYW5hZ2VyIFBs
```

```
dWdpbiBMaW51eCBTaWduZXIgS2V5KYkBAAQQEwgAqAUCZ5ERQ4EcQVdTIFNTTSBT
ZXNzaW9uIE1hbmFnZXIgPHNlc3Npb24tbWFuYWdlci1wbHVnaW4tc2lnbmVyQGFt
YXpvbi5jb20+IChBV1MgU3lzdGVtcyBNYW5hZ2VyIFNlc3Npb24gTWFuYWdlciBQ
bHVnaW4gTGludXggU2lnbmVyIEtleSkWIQR5WWNxJM4JOtUB1HosTUr/b2dX7gIe
AwIbAwIVCAAKCRAsTUr/b2dX7rO1AQCa1kig3lQ78W/QHGU76uHx3XAyv0tfpE9U
oQBCIwFLSgEA3PDHt3lZ+s6m9JLGJsy+Cp5ZFzpiF6RgluR/2gA861M=
=2DQm
-----END PGP PUBLIC KEY BLOCK-----
```

2. Importe a chave pública em seu token de autenticação. O valor da chave retornada deve ser 2C4D4AFF6F6757EE.

```
$ gpg --import session-manager-plugin.gpg
gpg: key 2C4D4AFF6F6757EE: public key "AWS SSM Session Manager <session-manager-
plugin-signer@amazon.com> (AWS Systems Manager Session Manager Plugin Linux Signer
 Key)" imported
gpg: Total number processed: 1
gpg:               imported: 1
```

3. Execute o comando a seguir para verificar a impressão digital.

```
gpg --fingerprint 2C4D4AFF6F6757EE
```

A impressão digital da saída do comando deve corresponder à mostrada a seguir.

```
7959 6371 24CE 093A D501 D47A 2C4D 4AFF 6F67 57EE
```

```
pub   nistp256 2025-01-22 [SC]
      7959 6371 24CE 093A D501  D47A 2C4D 4AFF 6F67 57EE
uid           [ unknown] AWS SSM Session Manager <session-manager-plugin-
signer@amazon.com> (AWS Systems Manager Session Manager Plugin Linux Signer Key)
```

Se a impressão digital não coincidir, não instale o plug-in. Entre em contato com a AWS Support.

4. Verifique a assinatura do pacote do instalador. Substitua *signature-filename* e *downloaded-plugin-filename* pelos valores especificados ao baixar o arquivo de assinatura e session-manager-plugin, conforme listados na tabela anteriormente neste tópico.

```
gpg --verify signature-filename downloaded-plugin-filename
```

Por exemplo, para a arquitetura x86_64 no Amazon Linux 2, o comando é o seguinte:

```
gpg --verify session-manager-plugin.rpm.sig session-manager-plugin.rpm
```

Esse comando retorna uma saída semelhante à seguinte:

```
gpg: Signature made Mon Feb 3 20:08:32 2025 UTC gpg: using ECDSA key
 2C4D4AFF6F6757EE
gpg: Good signature from "AWS Systems Manager Session Manager <session-manager-
plugin-signer@amazon.com> (AWS Systems Manager Session Manager Plugin Linux Signer
 Key)" [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: 7959 6371 24CE 093A D501 D47A 2C4D 4AFF 6F67 57EE
```

Se a saída inclui a frase `BAD signature`, verifique se você executou o procedimento corretamente. Se você continuar recebendo essa resposta, entre em contato com o AWS Support e não instale o pacote. A mensagem de aviso sobre a relação de confiança não significa que a assinatura não é válida, apenas que você não verificou a chave pública. Uma chave somente será confiável se você ou alguém em quem você confia a tiver assinado. Se a saída incluir a frase `Can't check signature: No public key`, verifique se você baixou o plug-in do Session Manager versão 1.2.707.0 ou posterior.

Instalar o plug-in do Session Manager no Amazon Linux 2, no Amazon Linux 2023 e nas distribuições do Red Hat Enterprise Linux

Use o procedimento a seguir para instalar o plug-in do Session Manager no Amazon Linux 2, no Amazon Linux 2023 (AL2023) e nas distribuições do RHEL.

> **Note**
>
> Não há suporte ao plug-in do Session Manager no Amazon Linux 1. Ele é compatível com o Amazon Linux 2 e distribuições posteriores.

1. Baixe e instale o pacote RPM do plug-in do Session Manager.

   x86_64

   No Amazon Linux 2 e no RHEL 7, execute o seguinte comando:

```
sudo yum install -y https://s3.amazonaws.com/session-manager-downloads/plugin/
latest/linux_64bit/session-manager-plugin.rpm
```

No AL2023 e no RHEL 8 e 9, execute o seguinte comando:

```
sudo dnf install -y https://s3.amazonaws.com/session-manager-downloads/plugin/
latest/linux_64bit/session-manager-plugin.rpm
```

x86

No RHEL 7, execute o seguinte comando:

```
sudo yum install -y https://s3.amazonaws.com/session-manager-downloads/plugin/
latest/linux_32bit/session-manager-plugin.rpm
```

No RHEL 8 e 9, execute o seguinte comando:

```
sudo dnf install -y https://s3.amazonaws.com/session-manager-downloads/plugin/
latest/linux_32bit/session-manager-plugin.rpm
```

ARM64

No Amazon Linux 2 e no RHEL 7, execute o seguinte comando:

```
sudo yum install -y https://s3.amazonaws.com/session-manager-downloads/plugin/
latest/linux_arm64/session-manager-plugin.rpm
```

No AL2023 e no RHEL 8 e 9, execute o seguinte comando:

```
sudo dnf install -y https://s3.amazonaws.com/session-manager-downloads/plugin/
latest/linux_arm64/session-manager-plugin.rpm
```

2. Verifique se a instalação foi bem-sucedida. Para mais informações, consulte Verifique a instalação do plugin do Session Manager.

> **Note**
>
> Se desejar desinstalar o plug-in, execute `sudo yum erase session-manager-plugin -y`

Instalar o plugin do Session Manager no Debian Server e no Ubuntu Server

1. Baixe o pacote deb do plugin do Session Manager:

   x86_64

   ```
   curl "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/ubuntu_64bit/session-manager-plugin.deb" -o "session-manager-plugin.deb"
   ```

   x86

   ```
   curl "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/ubuntu_32bit/session-manager-plugin.deb" -o "session-manager-plugin.deb"
   ```

   ARM64

   ```
   curl "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/ubuntu_arm64/session-manager-plugin.deb" -o "session-manager-plugin.deb"
   ```

2. Execute o comando de instalação.

   ```
   sudo dpkg -i session-manager-plugin.deb
   ```

3. Verifique se a instalação foi bem-sucedida. Para mais informações, consulte Verifique a instalação do plugin do Session Manager.

> **Note**
>
> Se você não deseja desinstalar o plugin, execute `sudo dpkg -r session-manager-plugin`

Verifique a instalação do plugin do Session Manager

Execute os comandos a seguir para verificar se a instalação do plugin do Session Manager foi bem-sucedida.

```
session-manager-plugin
```

Se a instalação foi bem-sucedida, a mensagem a seguir é retornada.

```
The Session Manager plugin is installed successfully. Use the AWS CLI to start a
 session.
```

Você também pode testar a instalação executando o comando start-session no AWS Command Line Interface (AWS CLI). No comando a seguir, substitua *instance-id* por suas próprias informações.

```
aws ssm start-session --target instance-id
```

Este comando funcionará somente se você tiver instalado e configurado o AWS CLI e o administrador do Session Manager tiver concedido as permissões do IAM necessárias para acessar o nó gerenciado de destino usando o Session Manager.

Plugin do Session Manager no GitHub

O código-fonte do plugin do Session Manager está disponível no GitHub e você poderá adaptá-lo de acordo com suas necessidades. Incentivamos você a enviar solicitações pull sobre alterações que gostaria que fosse incluídas. Porém, a Amazon Web Services não oferece suporte à execução de cópias modificadas desse software.

(Opcional) Ative o plugin de registro em log do Session Manager

O plugin do Session Manager inclui uma opção para permitir o registro em log para sessões que você executar. Por padrão, o registro em log está desativado.

Se você permitir o registro em log, o plugin Session Manager criará arquivos de log para ambas as atividades (`session-manager-plugin.log`) e erros (`errors.log`) da aplicação em sua máquina local.

Tópicos

- Ative o registro em log do plug-in do Session Manager (Windows)

- Habilite o registro em log para o plug-in do Session Manager (Linux e macOS)

Ative o registro em log do plug-in do Session Manager (Windows)

1. Localize o arquivo `seelog.xml.template` para o plugin.

   O local padrão é `C:\Program Files\Amazon\SessionManagerPlugin\seelog.xml.template`.

2. Altere o nome do arquivo para `seelog.xml`.
3. Abra o arquivo e altere `minlevel="off"` para `minlevel="info"` ou `minlevel="debug"`.

   > **Note**
   >
   > Por padrão, as entradas de log sobre como abrir um canal de dados e reconectar sessões são registrados no nível INFO. As entradas de fluxo de dados (pacotes e confirmação) são registradas no nível DEBUG.

4. Altere outras opções de configuração as quais deseja modificar. As opções que você pode alterar incluem:

   - Depurar nível: você pode alterar o nível de depuração de `formatid="fmtinfo"` para `formatid="fmtdebug"`.
   - Opções de arquivo de log: você pode fazer alterações nas opções de arquivo de log, incluindo onde os logs são armazenados, com exceção dos nomes dos arquivos.

   > **Important**
   >
   > Não altere os nomes de arquivos ou o registro não funcionará corretamente.

   ```
   <rollingfile type="size" filename="C:\Program Files\Amazon\SessionManagerPlugin
   \Logs\session-manager-plugin.log" maxsize="30000000" maxrolls="5"/>
   <filter levels="error,critical" formatid="fmterror">
   <rollingfile type="size" filename="C:\Program Files\Amazon\SessionManagerPlugin
   \Logs\errors.log" maxsize="10000000" maxrolls="5"/>
   ```

5. Salve o arquivo.

Habilite o registro em log para o plug-in do Session Manager (Linux e macOS)

1. Localize o arquivo `seelog.xml.template` para o plugin.

   O local padrão é `/usr/local/sessionmanagerplugin/seelog.xml.template`.

2. Altere o nome do arquivo para `seelog.xml`.

3. Abra o arquivo e altere `minlevel="off"` para `minlevel="info"` ou `minlevel="debug"`.

   > **Note**
   >
   > Por padrão, as entradas de log sobre como abrir canais de dados e reconectar sessões são registrados no nível INFO. As entradas de fluxo de dados (pacotes e confirmação) são registradas no nível DEBUG.

4. Altere outras opções de configuração as quais deseja modificar. As opções que você pode alterar incluem:

   - Depurar nível: você pode alterar o nível de depuração de `formatid="fmtinfo"` para `outputs formatid="fmtdebug"`
   - Opções de arquivo de log: você pode fazer alterações nas opções de arquivo de log, incluindo onde os logs são armazenados, com exceção dos nomes dos arquivos.

   > **Important**
   >
   > Não altere os nomes de arquivos ou o registro não funcionará corretamente.

   ```
   <rollingfile type="size" filename="/usr/local/sessionmanagerplugin/logs/session-
   manager-plugin.log" maxsize="30000000" maxrolls="5"/>
   <filter levels="error,critical" formatid="fmterror">
   <rollingfile type="size" filename="/usr/local/sessionmanagerplugin/logs/
   errors.log" maxsize="10000000" maxrolls="5"/>
   ```

   > **Important**
   >
   > Se você usar o diretório padrão para armazenar logs, você deve executar comandos de sessão usando sudo ou atribuir o diretório onde o plugin está instalado permissões

totais de leitura e gravação. Para ignorar essas restrições, altere o local onde os logs são armazenados.

5. Salve o arquivo.

Iniciar uma sessão

Você pode usar o console do AWS Systems Manager, o console do Amazon Elastic Compute Cloud (Amazon EC2), a AWS Command Line Interface (AWS CLI) ou o SSH para iniciar uma sessão.

Tópicos

- Iniciar uma sessão (console do Systems Manager)
- Início de uma sessão (console do Amazon EC2)
- Iniciar uma sessão (AWS CLI)
- Iniciar uma sessão (SSH)
- Iniciar uma sessão (encaminhamento de portas)
- Iniciar uma sessão (encaminhamento de portas para host remoto)
- Iniciar uma sessão (comandos interativos e não interativos)

Iniciar uma sessão (console do Systems Manager)

Você pode usar o console do AWS Systems Manager para iniciar uma sessão com um nó gerenciado na sua conta.

> **Note**
>
> Antes de iniciar uma sessão, conclua as etapas de configuração para o Session Manager. Para mais informações, consulte Configurar o Session Manager.

Para iniciar uma sessão (console do Systems Manager)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Session Manager.
3. Selecione Start session (Iniciar sessão).
4. (Opcional) Insira uma descrição para a sessão no campo Motivo da sessão.

5. Na lista Instâncias de destino, escolha o botão de opção à esquerda do nó gerenciado ao qual você deseja se conectar.

   Se o nó desejado não estiver na lista ou se você selecionar um nó e receber um erro de configuração, consulte Nó gerenciado não disponível ou não está configurado para o Session Manager para ver etapas de solução de problemas.
6. Escolha Iniciar sessão para iniciar a sessão imediatamente.

   - ou -

   Escolha Próximo para ver as opções da sessão.
7. (Opcional) Em Documento da sessão, selecione o documento que deseja executar quando a sessão começar. Se seu documento oferecer suporte a parâmetros de runtime, você poderá inserir um ou mais valores separados por vírgula em cada campo de parâmetro.
8. Escolha Próximo.
9. Selecione Start session (Iniciar sessão).

Depois que a conexão for feita, você poderá executar os comandos bash (Linux e macOS) ou PowerShell (Windows) como faria com qualquer outro tipo de conexão.

> **Important**
>
> Para permitir que os usuários especifiquem um documento ao iniciar sessões no console do Gerenciador de Sessões, observe o seguinte:
>
> - É necessário conceder aos usuários as permissões `ssm:GetDocument` e `ssm:ListDocuments` em suas políticas do IAM. Para obter mais informações, consulte Conceder acesso a documentos personalizados da sessão no console.
> - O console só oferece suporte a documentos de sessão que tenham `sessionType` definido como `Standard_Stream`. Para obter mais informações, consulte Esquema do documento de sessão.

Início de uma sessão (console do Amazon EC2)

Você pode usar o console do Amazon Elastic Compute Cloud (Amazon EC2) para iniciar uma sessão com uma instância na sua conta.

> **Note**
>
> Se você receber uma mensagem de erro informando que não está autorizado a executar uma ou mais ações do Systems Manager (`ssm:command-name`), entre em contato com o administrador para obter assistência. Caso seu administrador seja a pessoa que forneceu suas credenciais de início de sessão. Solicite que essa pessoa atualize suas políticas para permitir que você inicie sessões do console do Amazon EC2. Se você for um administrador, consulte Exemplo de políticas do IAM para Session Manager para obter mais informações.

Para iniciar uma sessão (console do Amazon EC2)

1. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, escolha Instances (Instâncias).
3. Selecione a instância e escolha Connect (Conectar).
4. Em Connection Method (Método de conexão), selecione Session Manager.
5. Selecione Conectar.

Depois que a conexão for feita, você poderá executar os comandos bash (Linux e macOS) ou PowerShell (Windows) como faria com qualquer outro tipo de conexão.

Iniciar uma sessão (AWS CLI)

Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

Antes de iniciar uma sessão, conclua as etapas de configuração para o Session Manager. Para mais informações, consulte Configurar o Session Manager.

Para usar a AWS CLI para executar comandos de sessão, o plugin do Session Manager também deve ser instalado em sua máquina local. Para mais informações, consulte Instalar o plug-in do Session Manager para a AWS CLI.

Para iniciar uma sessão usando o AWS CLI, execute o seguinte comando substituindo *instance-id* por suas próprias informações.

```
aws ssm start-session \
```

```
    --target instance-id
```

Para obter informações sobre outras opções que podem ser usadas com o comando start-session, consulte start-session na seção AWS Systems Manager da AWS CLI Command Reference.

Iniciar uma sessão (SSH)

Para iniciar uma sessão SSH do Session Manager, a versão 2.3.672.0 ou posterior do SSM Agent deve ser instalada no seu nó gerenciado.

Requisitos para a conexão SSH

Anote os seguintes requisitos e limitações para conexões de sessão usando SSH:

- O nó gerenciado de destino deve estar configurado para oferecer suporte a conexões SSH. Para obter informações, consulte (Opcional) Permitir e controlar permissões para conexões SSH por meio do Session Manager.
- Você deve se conectar usando a conta do nó gerenciado associada ao certificado PEM (Privacy Enhanced Mail), não a conta `ssm-user` usada para outros tipos de conexões de sessão. Por exemplo, em instâncias do EC2 para Linux e macOS, o usuário padrão é `ec2-user`. Para obter informações sobre como identificar o usuário padrão para cada tipo de instância, consulte Obter informações sobre a instância no Guia do usuário do Amazon EC2.
- O registro em log não está disponível para sessões do Session Manager que se conectam por meio de encaminhamento de portas ou SSH. Isso ocorre porque o SSH criptografa todos os dados da sessão e o Session Manager serve apenas como um túnel para conexões SSH.

> **Note**
>
> Antes de iniciar uma sessão, conclua as etapas de configuração para o Session Manager. Para mais informações, consulte Configurar o Session Manager.

Para iniciar uma sessão usando SSH, execute o comando a seguir. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

```
ssh -i /path/my-key-pair.pem username@instance-id
```

> **Tip**
>
> Ao iniciar uma sessão usando SSH, você pode copiar arquivos locais em seus nós de destino usando o seguinte formato de comando.
>
> ```
> scp -i /path/my-key-pair.pem /path/ExampleFile.txt username@instance-id:~
> ```

Para obter informações sobre outras opções que podem ser usadas com o comando start-session, consulte start-session na seção AWS Systems Manager da AWS CLI Command Reference.

Iniciar uma sessão (encaminhamento de portas)

Para iniciar uma sessão de encaminhamento da porta do Session Manager, a versão 2.3.672.0 ou posterior do SSM Agent deve ser instalada em seu nó gerenciado.

> **Note**
>
> Antes de iniciar uma sessão, conclua as etapas de configuração para o Session Manager. Para mais informações, consulte Configurar o Session Manager.
> Para usar a AWS CLI para executar comandos de sessão, o plugin do Session Manager deve ser instalado em sua máquina local. Para mais informações, consulte Instalar o plug-in do Session Manager para a AWS CLI.
> Dependendo do sistema operacional e da ferramenta de linha de comando, o posicionamento das aspas pode ser diferente e caracteres de escape podem ser necessários.

Para iniciar uma sessão de encaminhamento de portas, execute o comando a seguir na CLI. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

Linux & macOS

```
aws ssm start-session \
    --target instance-id \
    --document-name AWS-StartPortForwardingSession \
    --parameters '{"portNumber":["80"], "localPortNumber":["56789"]}'
```

Windows

```
aws ssm start-session ^
    --target instance-id ^
    --document-name AWS-StartPortForwardingSession ^
    --parameters portNumber="3389",localPortNumber="56789"
```

`portNumber` representa a porta remota no nó gerenciado para a qual você deseja que o tráfego da sessão seja redirecionado. Por exemplo, é possível especificar a porta `3389` para se conectar a um nó do Windows usando o protocolo da Área de Trabalho Remota (RDP). Se você não especificar o parâmetro `portNumber`, o Session Manager usará `80` como valor padrão.

`localPortNumber` é a porta no computador local onde o tráfego começa, como `56789`. Esse valor é o que você insere ao se conectar a um nó gerenciado usando um cliente. Por exemplo, .**`localhost:56789`**

Para obter informações sobre outras opções que podem ser usadas com o comando start-session, consulte start-session na seção AWS Systems Manager da AWS CLI Command Reference.

Para obter mais informações sobre sessões de encaminhamento de portas, consulte Encaminhamento de portas usando o AWS Systems Manager Session Manager no Blog de notícias da AWS.

Iniciar uma sessão (encaminhamento de portas para host remoto)

Para iniciar uma sessão de encaminhamento da portas do Session Manager para um host remoto, a versão 3.1.1374.0 ou posterior do SSM Agent deve ser instalada em seu nó gerenciado. O host remoto não precisa ser gerenciado pelo Systems Manager.

> **Note**
>
> Antes de iniciar uma sessão, conclua as etapas de configuração para o Session Manager. Para mais informações, consulte Configurar o Session Manager.
> Para usar a AWS CLI para executar comandos de sessão, o plugin do Session Manager deve ser instalado em sua máquina local. Para mais informações, consulte Instalar o plug-in do Session Manager para a AWS CLI.
> Dependendo do sistema operacional e da ferramenta de linha de comando, o posicionamento das aspas pode ser diferente e caracteres de escape podem ser necessários.

Para iniciar uma sessão de encaminhamento de portas, execute o comando a seguir na AWS CLI. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

Linux & macOS

```
aws ssm start-session \
    --target instance-id \
    --document-name AWS-StartPortForwardingSessionToRemoteHost \
    --parameters '{"host":["mydb.example.us-east-2.rds.amazonaws.com"],"portNumber":
["3306"], "localPortNumber":["3306"]}'
```

Windows

```
aws ssm start-session ^
    --target instance-id ^
    --document-name AWS-StartPortForwardingSessionToRemoteHost ^
    --parameters host="mydb.example.us-
east-2.rds.amazonaws.com",portNumber="3306",localPortNumber="3306"
```

O valor de `host` representa o nome do host ou endereço IP do host remoto com o qual você deseja estabelecer conexão. Requisitos gerais de conectividade e resolução de nomes entre o nó gerenciado e o host remoto ainda se aplicam.

`portNumber` representa a porta remota no nó gerenciado para a qual você deseja que o tráfego da sessão seja redirecionado. Por exemplo, é possível especificar a porta `3389` para se conectar a um nó do Windows usando o protocolo da Área de Trabalho Remota (RDP). Se você não especificar o parâmetro `portNumber`, o Session Manager usará `80` como valor padrão.

`localPortNumber` é a porta no computador local onde o tráfego começa, como `56789`. Esse valor é o que você insere ao se conectar a um nó gerenciado usando um cliente. Por exemplo, .**`localhost:56789`**

Para obter informações sobre outras opções que podem ser usadas com o comando start-session, consulte start-session na seção AWS Systems Manager da AWS CLI Command Reference.

Iniciar uma sessão com uma tarefa do Amazon ECS

O Session Manager oferece suporte ao início de sessões de encaminhamento de portas com uma tarefa dentro de um cluster do Amazon Elastic Container Service (Amazon ECS). Para fazer isso, é necessário atualizar o perfil da tarefa no IAM para incluir as seguintes permissões:

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
       "Effect": "Allow",
       "Action": [
            "ssmmessages:CreateControlChannel",
            "ssmmessages:CreateDataChannel",
            "ssmmessages:OpenControlChannel",
            "ssmmessages:OpenDataChannel"
       ],
      "Resource": "*"
      }
   ]
}
```

Para iniciar uma sessão de encaminhamento de portas com uma tarefa do Amazon ECS, execute o comando a seguir na AWS CLI. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

> **Note**
>
> Remova os símbolos < e > do parâmetro `target`. Esses símbolos são fornecidos apenas para fins de esclarecimento do leitor.

Linux & macOS

```
aws ssm start-session \
    --target ecs:<ECS_cluster_name>_<ECS_container_ID>_<container_runtime_ID> \
    --document-name AWS-StartPortForwardingSessionToRemoteHost \
    --parameters '{"host":["URL"],"portNumber":["port_number"], "localPortNumber":
["port_number"]}'
```

Windows

```
aws ssm start-session ^
    --target ecs:<ECS_cluster_name>_<ECS_container_ID>_<container_runtime_ID> ^
```

```
    --document-name AWS-StartPortForwardingSessionToRemoteHost ^
    --parameters host="URL",portNumber="port_number",localPortNumber="port_number"
```

Iniciar uma sessão (comandos interativos e não interativos)

Antes de iniciar uma sessão, conclua as etapas de configuração para o Session Manager. Para mais informações, consulte Configurar o Session Manager.

Para usar a AWS CLI para executar comandos de sessão, o plugin do Session Manager também deve ser instalado em sua máquina local. Para mais informações, consulte Instalar o plug-in do Session Manager para a AWS CLI.

Para iniciar uma sessão de comandos interativos, execute o seguinte comando: Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

Linux & macOS

```
aws ssm start-session \
    --target instance-id \
    --document-name CustomCommandSessionDocument \
    --parameters '{"logpath":["/var/log/amazon/ssm/amazon-ssm-agent.log"]}'
```

Windows

```
aws ssm start-session ^
    --target instance-id ^
    --document-name CustomCommandSessionDocument ^
    --parameters logpath="/var/log/amazon/ssm/amazon-ssm-agent.log"
```

Para obter informações sobre outras opções que podem ser usadas com o comando start-session, consulte start-session na seção AWS Systems Manager da AWS CLI Command Reference.

Mais informações

- Usar o encaminhamento de portas no AWS Systems Manager Session Manager para conectar a hosts remotos
- Encaminhamento de portas de instâncias do Amazon EC2 com o AWS Systems Manager
- Gerenciar recursos do Microsoft AD gerenciados pela AWS com encaminhamento de portas do Session Manager

- Port Forwarding Using AWS Systems ManagerSession Manager no Blog de notícias da AWS.

Encerrar uma sessão

Você pode encerrar uma sessão que foi iniciada em sua conta usando o console do AWS Systems Manager ou a AWS Command Line Interface (AWS CLI). Se você clicar no botão Encerrar de uma sessão no console ou chamar a ação da API TerminateSession usando a AWS CLI, o Session Manager encerrará permanentemente a sessão e fechará a conexão de dados entre o cliente do Session Manager e SSM Agent no nó gerenciado. Você não pode retomar uma sessão encerrada.

Se não houver atividade do usuário em uma sessão aberta por 20 minutos, o estado ocioso acionará um tempo limite. O Session Manager não chamará o `TerminateSession`, mas fechará o canal subjacente. Você não pode retomar uma sessão fechada devido ao tempo limite de inatividade.

Recomendamos sempre encerrar explicitamente uma sessão usando o comando `terminate-session`, ao usar a AWS CLI ou o botão Encerrar ao usar o console. (Os botões de Encerrar estão localizados na janela da sessão e na página principal do console do Session Manager.) Se você fechar somente um navegador ou uma janela de comando, a sessão permanecerá listada como Ativa no console por 30 dias. Quando você não encerra explicitamente uma sessão ou quando ela expira, todos os processos que estavam em execução no nó gerenciado no momento continuarão sendo executados.

Tópicos

- Encerrar uma sessão (console)
- Encerrar uma sessão (AWS CLI)

Encerrar uma sessão (console)

Você pode usar o console do AWS Systems Manager para encerrar uma sessão em sua conta.

Como encerrar uma sessão (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Session Manager.
3. Em Sessions (Sessões), escolha o botão de opção à esquerda da sessão que deseja encerrar.
4. Escolha Encerrar.

Encerrar uma sessão (AWS CLI)

Para encerrar uma sessão usando a AWS CLI, execute o comando a seguir. Substitua *session-id* por sua própria informação.

```
aws ssm terminate-session \
    --session-id session-id
```

Para obter mais informações sobre o comando terminate-session, consulte terminate-session na seção AWS Systems Manager da AWS CLI Command Reference.

Visualizar o histórico da sessão

Você pode usar o console do AWS Systems Manager ou a AWS Command Line Interface (AWS CLI) para visualizar as informações sobre sessões da sua conta. No console, você pode visualizar detalhes da sessão, como os seguintes:

- O ID da sessão
- Qual usuário se conectou a um nó gerenciado por uma sessão
- O ID do nó gerenciado.
- Quando a sessão foi iniciada e encerrada
- O status da sessão
- O local especificado para armazenar logs de sessão (se ativado)

Usando o AWS CLI, você pode visualizar uma lista de sessões em sua conta, mas não os detalhes adicionais que estão disponíveis no console.

Para obter informações sobre registro do histórico da sessão, consulte Habilitar e desabilitar o registro em log de sessão.

Tópicos

- Visualizar o histórico da sessão (console)
- Visualizar o histórico da sessão (AWS CLI)

Visualizar o histórico da sessão (console)

Você pode usar o console do AWS Systems Manager para visualizar detalhes sobre sessões da sua conta.

Para visualizar o histórico da sessão (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Session Manager.
3. Escolha a guia Session history (Histórico da sessão).

   - ou -

   Se a página inicial do Session Manager for aberta primeiro, escolha Configurar preferências e, em seguida, selecione a guia Histórico da sessão.

Visualizar o histórico da sessão (AWS CLI)

Para visualizar uma lista de sessões em sua conta usando a AWS CLI, execute o seguinte comando:

```
aws ssm describe-sessions \
    --state History
```

> **Note**
>
> Este comando retorna somente os resultados para conexões com destinos iniciados usando o Session Manager. Ele não lista conexões feitas por outros meios, como o protocolo RDP (Remote Desktop Protocol) ou o protocolo SSH (Secure Shell Protocol).

Para obter informações sobre outras opções que podem ser usadas com o comando describe-sessions, consulte describe-sessions na seção AWS Systems Manager da AWS CLI Command Reference.

Registrar a atividade de sessão de registro em log

Além de fornecer informações sobre sessões atuais e concluídas no console do Systems Manager, o Session Manager fornece a habilidade de registrar em log as atividades de sessões em sua conta da Conta da AWS usando o AWS CloudTrail.

O CloudTrail captura chamadas à API feitas no console do Systems Manager, na AWS Command Line Interface (AWS CLI) e no SDK do Systems Manager. Você pode visualizar as informações no console do CloudTrail ou em um bucket do Amazon Simple Storage Service (Amazon S3), onde elas estão armazenadas. Um bucket do Amazon S3 é usado em todos os logs do CloudTrail para sua

conta. Para ter mais informações, consulte Registrar em log chamadas de API do AWS Systems Manager com o AWS CloudTrail.

> **Note**
>
> Para uma análise recorrente, histórica e analítica dos seus arquivos de log, considere consultar os logs do CloudTrail usando o CloudTrail Lake ou uma tabela mantida por você. Para obter mais informações, consulte Consultar logs do AWS CloudTrail no Guia do usuário do AWS CloudTrail.

Monitorar as atividades da sessão usando o Amazon EventBridge (console)

Com o EventBridge, é possível configurar regras para detectar quando ocorrem alterações nos recursos da AWS. Você pode criar uma regra que detecta quando um usuário em sua organização inicia ou encerra uma sessão e, então, por exemplo, receber uma notificação por meio do Amazon SNS sobre o evento.

O suporte ao EventBridge para o Session Manager conta com registros de operações de API que foram registradas pelo CloudTrail. (Você pode usar integração do CloudTrail com EventBridge para responder à maioria dos eventos do AWS Systems Manager.) Ações que ocorrem dentro de uma sessão, como um comando `exit`, que não fazem uma chamada de API não são detectados pelo EventBridge.

As etapas a seguir descrevem como iniciar notificações por meio do Amazon Simple Notification Service (Amazon SNS) quando ocorre um evento da API do Session Manager, como StartSession.

Para monitorar atividades de sessão usando o Amazon EventBridge (console)

1. Crie um tópico do Amazon SNS a ser usado para enviar notificações, quando o evento do Session Manager ocorrer.

   Para obter mais informações, consulte Criar um tópico no Guia do desenvolvedor do Amazon Simple Notification Service.
2. Crie uma regra do EventBridge para invocar o destino do Amazon SNS para o tipo de evento do Session Manager que você quiser monitorar.

   Para obter informações sobre como criar regras de eventos, consulte Criar regras do Amazon EventBridge que reagem a eventos, no Guia do usuário do Amazon EventBridge.

À medida que você seguir as etapas para criar uma regra, selecione o seguinte:

- Em Serviço da AWS, escolha Systems Manager.
- Em Tipo de evento, escolha Chamada de API da AWS por meio do CloudTrail.
- Escolha Specific operations(s) (Operação(ões) específica(s)) e, em seguida, insira o(s) comando(s) do Session Manager (um de cada vez) que você deseja receber notificações. Você também pode escolher StartSession, ResumeSession e TerminateSession. (O EventBridge não oferece suporte aos comandos `Get*`, `List*` e `Describe*`).
- Em Select a target (Selecionar um destino), escolha SNS topic (Tópico do SNS). Em Topic (Tópico), escolha o nome do tópico do Amazon SNS criado na Etapa 1.

Para obter mais informações, consulte Manual do usuário do Amazon EventBridge User e o Guia de conceitos básicos do Amazon Simple Notification Service.

Habilitar e desabilitar o registro em log de sessão

O registro em log de sessão registra informações sobre sessões atuais e concluídas no console do Systems Manager. Você também pode registrar detalhes sobre os comandos executados durante as sessões na sua Conta da AWS. O registro em log de sessão permite que você:

- Criar e armazenar logs de sessão para fins de arquivamento.
- Gere um relatório que mostre detalhes de cada conexão feita para os nós gerenciados usando o Session Manager nos últimos 30 dias.
- Gere notificações de registro em log de sessão na Conta da AWS, como notificações do Amazon Simple Notification Service (Amazon SNS).
- Inicie automaticamente outra ação em um recurso da AWS como resultado das ações realizadas durante uma sessão, como executar uma função AWS Lambda, iniciar um pipeline do AWS CodePipeline ou executar um documento do AWS Systems Manager Run Command.

Important

Anote os seguintes requisitos e limitações para o Session Manager:

- O Session Manager registra os comandos inseridos e o resultados deles durante uma sessão, dependendo das suas preferências para a sessão. Para evitar que dados

confidenciais, como senhas, sejam exibidos em seus logs de sessão, recomendamos usar os seguintes comandos ao inserir dados confidenciais durante uma sessão.

Linux & macOS

```
stty -echo; read passwd; stty echo;
```

Windows

```
$Passwd = Read-Host -AsSecureString
```

- Se você estiver usando o Windows Server 2012 ou versões anteriores, os dados em seus logs poderão não ser formatados corretamente. Recomendamos usar o Windows Server 2012 R2 e posterior para formatos de log ideais.
- Se você estiver usando nós gerenciados do Linux ou do macOS, instale o utilitário de tela. Se não estiver, seus dados de log podem ser truncados. No Amazon Linux 1, Amazon Linux 2, AL2023 e Ubuntu Server, o utilitário de tela é instalado por padrão. Para instalar a tela manualmente, dependendo da versão do Linux, execute `sudo yum install screen` ou `sudo apt-get install screen`.
- O registro em log não está disponível para sessões do Session Manager que se conectam por meio de encaminhamento de portas ou SSH. Isso ocorre porque o SSH criptografa todos os dados da sessão e o Session Manager serve apenas como um túnel para conexões SSH.

Para obter mais informações sobre as permissões necessárias para usar o Amazon S3 ou o Amazon CloudWatch Logs para registrar dados da sessão em log, consulte Crie uma função do IAM com permissões para o Session Manager, Amazon S3 e CloudWatch Logs (console).

Consulte os tópicos a seguir para obter mais informações sobre as opções de registro do Session Manager.

Tópicos

- Transmitir dados da sessão usando o Amazon CloudWatch Logs (console)
- Registrar dados da sessão em log usando o Amazon S3 (console)
- Registrar dados da sessão em log usando o Amazon CloudWatch Logs (console)
- Configurar o registro em log da sessão em disco

- Ajustar por quanto tempo o arquivo de log temporário Session Manager é armazenado no disco
- Desabilitar o registro em log do Session Manager no CloudWatch Logs e no Amazon S3

Transmitir dados da sessão usando o Amazon CloudWatch Logs (console)

Você pode fazer uma transmissão contínua de logs de dados de sessão ao Amazon CloudWatch Logs. Detalhes essenciais, como os comandos que um usuário executou em uma sessão, o ID do usuário que executou os comandos e carimbos de data/hora para quando os dados da sessão são transmitidos para o CloudWatch Logs, são incluídos ao transmitir dados da sessão. Ao transmitir dados de sessão, os logs são formatados em JSON para ajudar você a integrar com suas soluções de log existentes. A transmissão de dados de sessão não é compatível com comandos interativos.

> **Note**
>
> Para transmitir dados de sessão dos nós gerenciados do Windows Server, você deve ter o PowerShell 5.1 ou posterior instalado. Por padrão, o Windows Server 2016 e versões posteriores têm a versão necessária do PowerShell instalada. No entanto, Windows Server 2012 e 2012 R2 não têm a versão necessária do PowerShell instalada por padrão. Se você ainda não tiver o PowerShell atualizado em seus nós gerenciados do Windows Server 2012 ou 2012 R2, poderá fazer isso usando o Run Command. Para obter informações sobre como atualizar o PowerShell usando o Run Command, consulte Atualização da PowerShell por meio de Run Command.

> **Important**
>
> Se você tiver a configuração da política PowerShell Transcription definida em seus nós gerenciados do Windows Server, não será possível transmitir os dados da sessão.

Para transmitir dados da sessão usando o Amazon CloudWatch Logs (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Session Manager.
3. Escolha a guia Preferences (Preferências) e, em seguida, escolha Edit (Editar).
4. Marque a caixa de seleção ao lado de Enable (Habilitar) em CloudWatch logging (Registro em log do CloudWatch).

5. Selecione a opção Stream session logs (Transmitir logs da sessão).
6. (Recomendado) Marque a caixa de seleção ao lado de Allow only encrypted CloudWatch log groups (Permitir apenas grupos de logs criptografados do CloudWatch). Com essa opção ativada, os dados de log serão criptografados usando a chave de criptografia no lado do servidor especificada para o grupo de logs. Se não quiser criptografar os dados de log enviados ao CloudWatch Logs, desmarque a caixa de seleção. Você também deverá desmarcar a caixa de seleção se a criptografia não for permitida no grupo de logs.
7. Em CloudWatch logs (Logs do CloudWatch), para especificar o grupo de logs do CloudWatch Logs na Conta da AWS no qual carregar logs de sessão, selecione uma das seguintes opções:
   - Insira um nome do grupo de logs na caixa de texto que já tiver sido criada na conta para armazenar dados de log da sessão.
   - Escolha um grupo de logs: selecione um grupo de logs que já tenha sido criado na sua conta para armazenar dados de log da sessão.
8. Escolha Salvar.

Registrar dados da sessão em log usando o Amazon S3 (console)

Você pode optar por armazenar os dados de log da sessão em um bucket do Amazon Simple Storage Service (Amazon S3) de sua escolha para fins de depuração e solução de problemas. A opção padrão é que os logs sejam enviados para um bucket do Amazon S3 criptografado. A criptografia é feita usando a chave especificada para o bucket, uma chave do AWS KMS key ou uma chave de criptografia no lado do servidor (SSE) (AES-256) do Amazon S3.

> **Important**
>
> Ao usar buckets hospedados virtualmente com Secure Sockets Layer (SSL), o certificado SSL curinga corresponde somente a buckets que não contenham pontos. Para contornar isso, use HTTP ou escreva a sua própria lógica de verificação do certificado. Recomendamos não usar pontos (".") em nomes de buckets ao usar buckets hospedados virtualmente.

Criptografia de bucket do Amazon S3

Para enviar logs ao seu bucket do Amazon S3 com criptografia, a mesma deve ser ativada no bucket. Para obter informações sobre a criptografia de buckets do Amazon S3, consulte Definir o comportamento padrão da criptografia para os buckets do Amazon S3.

Chave gerenciada pelo cliente

Se você estiver usando uma chave do KMS que você mesmo gerencia para criptografar o bucket, o perfil da instância do IAM anexado às suas instâncias deve ter permissões explícitas para ler a chave. Se você usar uma Chave gerenciada pela AWS, a instância não exigirá essa permissão explícita. Para obter mais informações sobre como fornecer o perfil da instância com acesso para usar a chave, consulte Permitir que os usuários de chaves usem a chave do KMS no Guia do desenvolvedor do AWS Key Management Service.

Siga estas etapas para configurar o Session Manager para armazenar logs de sessão em um bucket do Amazon S3.

> **Note**
>
> Você também pode usar o AWS CLI para especificar ou alterar o bucket do Amazon S3 para os quais os dados serão enviados. Para mais informações, consulte Atualizar preferências do Session Manager (linha de comando).

Para registrar dados da sessão em log usando o Amazon S3 (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Session Manager.
3. Escolha a guia Preferences (Preferências) e, em seguida, escolha Edit (Editar).
4. Marque a caixa de seleção ao lado de Enable (Habilitar) em S3 logging (Registro em log do S3).
5. (Recomendado) Marque a caixa de seleção ao lado de Allow only encrypted S3 buckets (Permitir apenas buckets do S3 criptografados). Com essa opção ativada, os dados de log serão criptografados usando a chave de criptografia no lado do servidor especificada para o bucket. Se não quiser criptografar os dados de log enviados ao Amazon S3, desmarque a caixa de seleção. Você também deverá desmarcar a caixa de seleção se a criptografia não for permitida no bucket do S3.
6. Para S3 bucket name (Nome do bucket do S3), selecione uma das seguintes opções:

   > **Note**
   >
   > Recomendamos não usar pontos (".") em nomes de buckets ao usar buckets hospedados virtualmente. Para obter informações sobre as convenções para nomear

buckets do Amazon S3, consulte Restrições e limitações de buckets no guia do usuário do Amazon Simple Storage Service.

- Escolha um nome de bucket na lista: selecione um bucket do Amazon S3 que já tenha sido criado na sua conta para armazenar dados de log de sessão.
- Insira um nome de bucket na caixa de texto: insira o nome de um bucket do Amazon S3 que já tenha sido criado na conta para armazenar os dados de log da sessão.

7. (Opcional) Para prefixo de chaves do S3, insira o nome de uma pasta existente ou uma nova pasta para armazenar logs no bucket selecionado.
8. Escolha Salvar.

Para obter mais informações sobre como trabalhar com o Amazon S3 e com buckets do Amazon S3, consulte o Guia do usuário do Amazon Simple Storage Service e o Guia do usuário do Amazon Simple Storage Service.

Registrar dados da sessão em log usando o Amazon CloudWatch Logs (console)

Com o Amazon CloudWatch Logs, é possível monitorar, armazenar e acessar os arquivos de log de vários Serviços da AWS. Você pode enviar dados de log da sessão a um grupo de logs do CloudWatch Logs para fins de depuração e solução de problemas. A opção padrão é que os dados de log sejam enviados com criptografia usando sua chave do KMS, mas é possível enviar esses dados ao grupo de logs com ou sem criptografia.

Siga estas etapas para configurar AWS Systems Manager Session Manager para enviar dados de log da sessão a um grupo de logs do CloudWatch Logs no final das sessões.

> **Note**
>
> Você também pode usar a AWS CLI para especificar ou alterar o grupo de logs do CloudWatch Logs ao qual os dados de sessão serão enviados. Para mais informações, consulte Atualizar preferências do Session Manager (linha de comando).

Para registrar dados da sessão em log usando o Amazon CloudWatch Logs (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Session Manager.

3. Escolha a guia Preferences (Preferências) e, em seguida, escolha Edit (Editar).
4. Marque a caixa de seleção ao lado de Enable (Habilitar) em CloudWatch logging (Registro em log do CloudWatch).
5. Selecione a opção Upload session logs (Carregar os logs da sessão).
6. (Recomendado) Marque a caixa de seleção ao lado de Allow only encrypted CloudWatch log groups (Permitir apenas grupos de logs criptografados do CloudWatch). Com essa opção ativada, os dados de log serão criptografados usando a chave de criptografia no lado do servidor especificada para o grupo de logs. Se não quiser criptografar os dados de log enviados ao CloudWatch Logs, desmarque a caixa de seleção. Você também deverá desmarcar a caixa de seleção se a criptografia não for permitida no grupo de logs.
7. Em CloudWatch logs (Logs do CloudWatch), para especificar o grupo de logs do CloudWatch Logs na Conta da AWS no qual carregar logs de sessão, selecione uma das seguintes opções:
   - Choose a log group from the list (Escolha um grupo de logs na lista): selecione um grupo de logs que já tenha sido criado na sua conta para armazenar dados de log de sessão.
   - Enter a log group name in the text box (Insira um nome de grupo de logs na caixa de texto): insira o nome de um grupo de logs que já tenha sido criado na sua conta para armazenar dados de log de sessão.
8. Escolha Salvar.

Para obter mais informações sobre o CloudWatch Logs, consulte Guia do usuário do Amazon CloudWatch Logs.

Configurar o registro em log da sessão em disco

Após o registro em log do Session Manager no CloudWatch ou no Amazon S3 ser habilitado, todos os comandos executados durante uma sessão (e a saída resultante desses comandos) são registrados em log em um arquivo temporário no disco da instância de destino. O arquivo temporário é chamado `ipcTempFile.log`.

O `ipcTempFile.log` é controlado pelo parâmetro `SessionLogsDestination` no arquivo de configuração do SSM Agent. O parâmetro aceita os seguintes valores:

- disk: se você especificar esse parâmetro e o registro da sessão em log no CloudWatch ou no Amazon S3 estiver habilitado, o SSM Agent criará o arquivo de log temporário `ipcTempFile.log` e registrará os comandos da sessão e a saída em log no disco. O Session Manager carrega esse log no CloudWatch ou no S3 durante ou após a sessão, dependendo da configuração do

registro em log. O log é então excluído de acordo com a duração especificada para o parâmetro de configuração `SessionLogsRetentionDurationHours` do SSM Agent.

Se você especificar esse parâmetro e o registro da sessão em log no CloudWatch e no Amazon S3 estiver desabilitado, o SSM Agent ainda registrará o histórico e a saída de comandos no arquivo `ipcTempFile.log`. O log será então excluído de acordo com a duração especificada para o parâmetro de configuração `SessionLogsRetentionDurationHours` do SSM Agent.

- none: se você especificar esse parâmetro e o registro da sessão em log no CloudWatch ou no Amazon S3 estiver habilitado, o registro em log no disco funcionará exatamente como se você tivesse especificado o parâmetro `disk`. O SSM Agent requer o arquivo temporário quando o registro de sessão no CloudWatch ou no Amazon S3 está habilitado.

  Se você especificar esse parâmetro e o registro da sessão em log no CloudWatch e no Amazon S3 estiver desabilitado, o SSM Agent não criará o arquivo `ipcTempFile.log`.

Use o procedimento a seguir para habilitar ou desabilitar a criação do arquivo de log temporário `ipcTempFile.log` no disco quando uma sessão for iniciada.

Para habilitar ou desabilitar a criação do arquivo de log temporário Session Manager no disco

1. Instale SSM Agent na sua instância ou atualize para a versão 3.2.2086 ou superior. Para obter informações sobre como verificar o número da versão do agente, consulte Verificar o número de versão do SSM Agent. Para obter informações sobre como instalar manualmente o agente, localize o procedimento para seu sistema operacional nas seguintes seções:

   - Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux
   - Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para macOS
   - Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Windows Server
2. Conecte-se à sua instância e localize o `amazon-ssm-agent.json` arquivo no local a seguir.

   - Linux: /etc/amazon/ssm/
   - macOS: /opt/aws/ssm/
   - Windows Server: C:\Program Files\Amazon\SSM

   Se o arquivo `amazon-ssm-agent.json` não existir, copie o conteúdo de `amazon-ssm-agent.json.template` para um novo arquivo no mesmo diretório. Nomeie o novo arquivo como `amazon-ssm-agent.json`.

3. Especifique `none` ou `disk` para o parâmetro `SessionLogsDestination`. Salve as alterações.
4. Reinicie o SSM Agent.

Se você especificou `disk` para o parâmetro `SessionLogsDestination`, poderá verificar se o SSM Agent cria o arquivo de log temporário iniciando uma nova sessão e, em seguida, localizando o arquivo `ipcTempFile.log` no seguinte local:

- Linux: /var/lib/amazon/ssm/*target ID*/session/orchestration/*session ID*/Standard_Stream/ipcTempFile.log
- macOS: /opt/aws/ssm/data/*target ID*/session/orchestration/*session ID*/Standard_Stream/ipcTempFile.log
- Windows Server: C:\ProgramData\Amazon\SSM\InstanceData*target ID*\session\orchestration*session ID*\Standard_Stream\ipcTempFile.log

> **Note**
>
> Por padrão, o arquivo de log temporário é salvo na instância por 14 dias.

Caso deseje atualizar o parâmetro `SessionLogsDestination` em várias instâncias, recomendamos criar um documento do SSM que especifique a nova configuração. Em seguida, você poderá usar o Systems Manager Run Command para implementar a alteração em suas instâncias. Para obter mais informações, consulte Como escrever seus próprios documentos do AWS Systems Manager (blog) e Execução de comandos em nós gerenciados.

Ajustar por quanto tempo o arquivo de log temporário Session Manager é armazenado no disco

Após o registro em log do Session Manager no CloudWatch ou no Amazon S3 ser habilitado, todos os comandos executados durante uma sessão (e a saída resultante desses comandos) são registrados em log em um arquivo temporário no disco da instância de destino. O arquivo temporário é chamado `ipcTempFile.log`. Durante uma sessão, ou após sua conclusão, o Session Manager carrega esse log temporário no CloudWatch ou no S3. O log temporário é então excluído de acordo com a duração especificada para o parâmetro de configuração `SessionLogsRetentionDurationHours` do SSM Agent. Por padrão, o arquivo de log temporário é salvo na instância por 14 dias no seguinte local:

- Linux: /var/lib/amazon/ssm/*target ID*/session/orchestration/*session ID*/Standard_Stream/ipcTempFile.log
- macOS: /opt/aws/ssm/data/*target ID*/session/orchestration/*session ID*/Standard_Stream/ipcTempFile.log
- Windows Server: C:\ProgramData\Amazon\SSM\InstanceData*target ID*\session\orchestration*session ID*\Standard_Stream\ipcTempFile.log

Use o procedimento a seguir para ajustar por quanto tempo o arquivo de log temporário Session Manager é armazenado no disco.

Para ajustar por quanto tempo o arquivo **`ipcTempFile.log`** é armazenado no disco

1. Conecte-se à sua instância e localize o `amazon-ssm-agent.json` arquivo no local a seguir.
   - Linux: /etc/amazon/ssm/
   - macOS: /opt/aws/ssm/
   - Windows Server: C:\Program Files\Amazon\SSM

   Se o arquivo `amazon-ssm-agent.json` não existir, copie o conteúdo de `amazon-ssm-agent.json.template` para um novo arquivo no mesmo diretório. Nomeie o novo arquivo como `amazon-ssm-agent.json`.
2. Altere o valor de `SessionLogsRetentionDurationHours` para o número de horas desejado. Se `SessionLogsRetentionDurationHours` estiver definido como 0, o arquivo de log temporário será criado durante a sessão e excluído quando a sessão for concluída. Essa configuração deve garantir que o arquivo de log não persista após o término da sessão.
3. Salve as alterações.
4. Reinicie o SSM Agent.

Desabilitar o registro em log do Session Manager no CloudWatch Logs e no Amazon S3

Você pode usar o console do Systems Manager ou a AWS CLI para desabilitar o registro em log de sessão em sua conta.

Para desabilitar o registro em log de sessão (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.

2. No painel de navegação, escolha Session Manager.
3. Escolha a guia Preferences (Preferências) e, em seguida, escolha Edit (Editar).
4. Para desabilitar o registro em log do CloudWatch, na seção Registro em log do CloudWatch, desmarque a caixa de seleção Habilitar.
5. Para desabilitar o registro em log do S3, na seção Registro em log do S3, desmarque a caixa de seleção Habilitar.
6. Escolha Salvar.

Para desabilitar o registro em log de sessão (AWS CLI)

Para desabilitar o registro em log de sessão usando a AWS CLI, siga as instruções em Atualizar preferências do Session Manager (linha de comando).

No arquivo JSON, certifique-se de que as entradas `s3BucketName` e `cloudWatchLogGroupName` não contenham valores. Por exemplo:

```
"inputs": {
        "s3BucketName": "",
        ...
        "cloudWatchLogGroupName": "",
        ...
    }
```

Se preferir, para desabilitar o registro em log, você poderá remover todas as entradas `S3*` e `cloudWatch*` do arquivo JSON.

> **Note**
>
> Dependendo da sua configuração, após a desabilitação do CloudWatch ou S3, um arquivo de log temporário ainda poderá ser gerado em disco pelo SSM Agent. Para obter informações sobre como desabilitar o log em disco, consulte Configurar o registro em log da sessão em disco.

Esquema do documento de sessão

As informações a seguir descrevem os elementos do esquema de um documento de sessão. O Session Manager do AWS Systems Manager usa documentos de sessão para determinar qual tipo

de sessão iniciar, como uma sessão padrão, uma sessão de encaminhamento de portas ou uma sessão para executar um comando interativo.

schemaVersion

A versão do esquema do documento de sessão. Documentos de sessão são compatíveis apenas com a versão 1.0.

Tipo: string

Obrigatório: Sim

description

Uma descrição que você especifica para o documento da Session (Seção). Por exemplo, “Documento para iniciar sessão de encaminhamento de porta com o Session Manager“.

Tipo: String

Obrigatório: Não

sessionType

O tipo de sessão que o documento de sessão é usado para estabelecer.

Tipo: string

Obrigatório: Sim

Valores válidos: `InteractiveCommands` | `NonInteractiveCommands` | `Port` | `Standard_Stream`

inputs

As preferências de sessão a serem usadas para sessões estabelecidas usando este documento de sessão. Este elemento é necessário para documentos de sessão que são usados para criar sessões `Standard_Stream`.

Tipo: StringMap

Obrigatório: Não

s3BucketName

O bucket do Amazon Simple Storage Service (Amazon S3) para o qual você deseja enviar logs de sessão no final das sessões.

Tipo: String

Obrigatório: Não

s3KeyPrefix

O prefixo a ser usado ao enviar logs para o bucket do Amazon S3 que você especificou na entrada `s3BucketName`. Para obter mais informações sobre como usar um prefixo compartilhado com objetos armazenados no Amazon S3, consulte Como usar pastas em um bucket do S3? no Guia do usuário do Amazon Simple Storage Service.

Tipo: String

Obrigatório: Não

s3EncryptionEnabled

Se definido como `true`, o bucket do Amazon S3 que você especificou na entrada `s3BucketName` deve ser criptografado.

Tipo: booliano

Obrigatório: Sim

cloudWatchLogGroupName

O nome do grupo de Amazon CloudWatch Logs (CloudWatch Logs) para o qual você deseja enviar logs de sessão no final das sessões.

Tipo: String

Obrigatório: Não

cloudWatchEncryptionEnabled

Se definido como `true`, o grupo de logs que você especificou na entrada `cloudWatchLogGroupName` deve ser criptografado.

Tipo: booliano

Obrigatório: Sim

cloudWatchStreamingEnabled

Se definido como `true`, uma transmissão contínua de logs de dados de sessão será enviada ao grupo de logs que você especificou na entrada `cloudWatchLogGroupName`. Se definido

como `false`, os logs da sessão serão enviados ao grupo de logs que você especificou na entrada `cloudWatchLogGroupName` no final de suas sessões.

Tipo: booliano

Obrigatório: Sim

kmsKeyId

O ID da AWS KMS key que você quer usar para reforçar a criptografia de dados entre suas máquinas cliente locais e os nós gerenciados do Amazon Elastic Compute Cloud (Amazon EC2) aos quais você se conecta.

Tipo: String

Obrigatório: Não

runAsEnabled

Se definido como `true`, você deverá especificar uma conta de usuário existente em nós gerenciados aos quais você se conectará na entrada `runAsDefaultUser`. Caso contrário, as sessões não serão iniciadas. Por padrão, as sessões são iniciadas usando a conta do `ssm-user` criada pelo AWS Systems Manager SSM Agent. O atributo Run As é compatível apenas com a conexão com nós gerenciados para Linux e macOS.

Tipo: booliano

Obrigatório: Sim

runAsDefaultUser

O nome da conta de usuário com a qual iniciar as sessões em nós gerenciados do Linux e do macOS quando a entrada `runAsEnabled` estiver definida como `true`. A conta de usuário especificada para essa entrada deve existir em nós gerenciados aos quais você se conectará. Caso contrário, as sessões não serão iniciadas.

Tipo: String

Obrigatório: Não

idleSessionTimeout

A quantidade de tempo de inatividade que você deseja permitir antes do término de uma sessão. Essa entrada é medida em minutos.

Tipo: String

Valores válidos: 1 a 60

Obrigatório: Não

maxSessionDuration

O tempo máximo que você deseja permitir antes do término de uma sessão. Essa entrada é medida em minutos.

Tipo: String

Valores válidos: 1 a 1440

Obrigatório: Não

shellProfile

As preferências especificadas por sistema operacional para aplicar dentro das sessões, como preferências de shell, variáveis de ambiente, diretórios de trabalho e execução de vários comandos quando uma sessão é iniciada.

Tipo: StringMap

Obrigatório: Não

windows

As preferências do shell, variáveis de ambiente, diretórios de trabalho e comandos especificados para sessões em nós gerenciados do Windows Server.

Tipo: String

Obrigatório: Não

linux

As preferências do shell, variáveis de ambiente, diretórios de trabalho e comandos especificados para sessões em nós gerenciados do Linux e do macOS .

Tipo: String

Obrigatório: Não

parameters

Um objeto que define os parâmetros que o documento aceita. Para obter mais informações sobre como especificar os parâmetros do documento, consulte parâmetros no Elementos de

dados de nível superior. Com relação aos parâmetros aos quais você se refere com frequência, recomendamos armazená-los no Parameter Store do Systems Manager para isso. Você pode fazer referência aos parâmetros `String` e `StringList` do Parameter Store nesta seção de um documento. Você pode fazer referência aos parâmetros `SecureString` e Parameter Store nesta seção de um documento. Você pode fazer referência a um parâmetro do Parameter Store usando o seguinte formato:

```
{{ssm:parameter-name}}
```

Para obter mais informações sobre o Parameter Store, consulte AWS Systems Manager Parameter Store.

Tipo: StringMap

Obrigatório: Não

properties

Um objeto cujos valores que você especificar que são usados na operação da API do `StartSession`.

Para documentos de sessão que são usados para as sessões do `InteractiveCommands`, o objeto de propriedades inclui os comandos a serem executados nos sistemas operacionais especificados. Também é possível determinar se os comandos são executados como `root` usando a propriedade booleana `runAsElevated`. Para obter mais informações, consulte Restringir acesso a comandos em uma sessão.

Para documentos de sessão que são usados para sessões do `Port`, o objeto de propriedades contém o número da porta para a qual o tráfego deve ser redirecionado. Para ver um exemplo, consulte o exemplo de documento de sessão posteriormente `Port` neste tópico.

Tipo: StringMap

Obrigatório: Não

`Standard_Stream` Digite exemplo de documento de sessão

YAML

```
---
```

```
schemaVersion: '1.0'
description: Document to hold regional settings for Session Manager
sessionType: Standard_Stream
inputs:
  s3BucketName: ''
  s3KeyPrefix: ''
  s3EncryptionEnabled: true
  cloudWatchLogGroupName: ''
  cloudWatchEncryptionEnabled: true
  cloudWatchStreamingEnabled: true
  kmsKeyId: ''
  runAsEnabled: true
  runAsDefaultUser: ''
  idleSessionTimeout: '20'
  maxSessionDuration: '60'
  shellProfile:
    windows: ''
    linux: ''
```

JSON

```
{
    "schemaVersion": "1.0",
    "description": "Document to hold regional settings for Session Manager",
    "sessionType": "Standard_Stream",
    "inputs": {
        "s3BucketName": "",
        "s3KeyPrefix": "",
        "s3EncryptionEnabled": true,
        "cloudWatchLogGroupName": "",
        "cloudWatchEncryptionEnabled": true,
        "cloudWatchStreamingEnabled": true,
        "kmsKeyId": "",
        "runAsEnabled": true,
        "runAsDefaultUser": "",
        "idleSessionTimeout": "20",
        "maxSessionDuration": "60",
        "shellProfile": {
            "windows": "date",
            "linux": "pwd;ls"
        }
    }
}
```

`InteractiveCommands` Digite exemplo de documento de sessão

YAML

```
---
schemaVersion: '1.0'
description: Document to view a log file on a Linux instance
sessionType: InteractiveCommands
parameters:
  logpath:
    type: String
    description: The log file path to read.
    default: "/var/log/amazon/ssm/amazon-ssm-agent.log"
    allowedPattern: "^[a-zA-Z0-9-_/]+(.log)$"
properties:
  linux:
    commands: "tail -f {{ logpath }}"
    runAsElevated: true
```

JSON

```
{
    "schemaVersion": "1.0",
    "description": "Document to view a log file on a Linux instance",
    "sessionType": "InteractiveCommands",
    "parameters": {
        "logpath": {
            "type": "String",
            "description": "The log file path to read.",
            "default": "/var/log/amazon/ssm/amazon-ssm-agent.log",
            "allowedPattern": "^[a-zA-Z0-9-_/]+(.log)$"
        }
    },
    "properties": {
        "linux": {
            "commands": "tail -f {{ logpath }}",
            "runAsElevated": true
        }
    }
}
```

`Port` Digite exemplo de documento de sessão

YAML

```
---
schemaVersion: '1.0'
description: Document to open given port connection over Session Manager
sessionType: Port
parameters:
  paramExample:
    type: string
    description: document parameter
properties:
  portNumber: anyPortNumber
```

JSON

```
{
    "schemaVersion": "1.0",
    "description": "Document to open given port connection over Session Manager",
    "sessionType": "Port",
    "parameters": {
        "paramExample": {
            "type": "string",
            "description": "document parameter"
        }
    },
    "properties": {
        "portNumber": "anyPortNumber"
    }
}
```

Exemplo de documento de sessão com caracteres especiais

YAML

```
---
schemaVersion: '1.0'
description: Example document with quotation marks
sessionType: InteractiveCommands
parameters:
  Test:
    type: String
    description: Test Input
```

```
    maxChars: 32
properties:
  windows:
    commands: |
        $Test = '{{ Test }}'
        $myVariable = \"Computer name is $env:COMPUTERNAME\"
        Write-Host "Test variable: $myVariable`.`nInput parameter: $Test"
    runAsElevated: false
```

JSON

```
{
   "schemaVersion":"1.0",
   "description":"Test document with quotation marks",
   "sessionType":"InteractiveCommands",
   "parameters":{
      "Test":{
         "type":"String",
         "description":"Test Input",
         "maxChars":32
      }
   },
   "properties":{
      "windows":{
         "commands":[
            "$Test = '{{ Test }}'",
            "$myVariable = \\\"Computer name is $env:COMPUTERNAME\\\"",
            "Write-Host \"Test variable: $myVariable`.`nInput parameter: $Test\""
         ],
         "runAsElevated":false
      }
   }
}
```

Solução de problemas do Session Manager

Use as informações a seguir para ajudar a solucionar problemas com o Session Manager do AWS Systems Manager.

Tópicos

- AccessDeniedException ao chamar a operação TerminateSession

- O processo do documento falhou inesperadamente: o operador do documento excedeu o tempo limite
- O Session Manager não consegue se conectar pelo console do Amazon EC2
- Sem permissão para iniciar uma sessão
- SSM Agent não está online
- Sem permissão para alterar as preferências da sessão
- Nó gerenciado não disponível ou não está configurado para o Session Manager
- O plugin Session Manager não foi encontrado
- O plug-in Session Manager não é adicionado automaticamente ao caminho de linha de comando (Windows)
- O plugin Session Manager não responde
- TargetNotConnected
- Tela em branco exibida após iniciar uma sessão
- O nó gerenciado deixa de responder durante sessões de execução longa
- Ocorreu um erro (InvalidDocument) ao chamar a operação StartSession

AccessDeniedException ao chamar a operação TerminateSession

Problema: ao tentar encerrar uma sessão, o Systems Manager retorna o seguinte erro:

```
An error occurred (AccessDeniedException) when calling the TerminateSession operation:
User: <user_arn> is not authorized to perform: ssm:TerminateSession on resource:
<ssm_session_arn> because no identity-based policy allows the ssm:TerminateSession
 action.
```

Solução A: confirme se a versão mais recente do plug-in Session Manager está instalada no nó

Insira o comando a seguir no terminal e pressione Enter.

```
session-manager-plugin --version
```

Solução B: instale ou reinstale a versão mais recente do plug-in

Para obter mais informações, consulte Instalar o plug-in do Session Manager para a AWS CLI.

Solução C: tente restabelecer uma conexão com o nó

Verifique se o nó está respondendo às solicitações. Tente restabelecer a sessão. Ou, se necessário, abra o console do Amazon EC2 e verifique se o status da instância está em execução.

O processo do documento falhou inesperadamente: o operador do documento excedeu o tempo limite

Problema: ao iniciar uma sessão em um host Linux, o Systems Manager retorna o seguinte erro:

```
document process failed unexpectedly: document worker timed out,
check [ssm-document-worker]/[ssm-session-worker] log for crash reason
```

Se você configurou o log do SSM Agent conforme descrito em Visualizar logs do SSM Agent, você poderá visualizar mais detalhes no registro de depuração. Para esse problema, o Session Manager mostra a seguinte entrada de log:

```
failed to create channel: too many open files
```

Esse erro normalmente indica que há muitos processos de operador do Session Manager em execução e que o sistema operacional subjacente atingiu um limite. Há duas opções para resolver o problema.

Solução A: aumentar o limite de notificação de arquivos do sistema operacional

É possível aumentar o limite executando o comando a seguir em um host Linux separado. Esse comando usa o Run Command do Systems Manager. O valor especificado aumenta `max_user_instances` para 8192. Esse valor é consideravelmente maior do que o valor padrão de 128, mas não sobrecarregará os recursos do host:

```
aws ssm send-command --document-name AWS-RunShellScript \
--instance-id i-02573cafcfEXAMPLE  --parameters \
"commands=sudo sysctl fs.inotify.max_user_instances=8192"
```

Solução B: diminua as notificações de arquivo usadas pelo Session Manager no host de destino

Execute o comando a seguir em um host Linux separado para listar as sessões em execução no host de destino:

```
aws ssm describe-sessions --state Active --filters key=Target,value=i-02573cafcfEXAMPLE
```

Revise a saída do comando para identificar as sessões que não são mais necessárias. É possível encerrar essas sessões executando o seguinte comando em um host Linux separado:

```
aws ssm terminate-session –session-id session ID
```

Opcionalmente, quando não houver mais sessões em execução no servidor remoto, você poderá liberar recursos adicionais executando o comando a seguir em um host Linux separado. Este comando encerra todos os processos do Session Manager em execução no host remoto e, consequentemente, todas as sessões no host remoto. Antes de executá-lo, verifique se não há sessões em andamento que você gostaria de manter:

```
aws ssm send-command --document-name AWS-RunShellScript \
            --instance-id i-02573cafcfEXAMPLE --parameters \
'{"commands":["sudo kill $(ps aux | grep ssm-session-worker | grep -v grep | awk
 '"'"'{print $2}'"'"')"]}'
```

O Session Manager não consegue se conectar pelo console do Amazon EC2

Problema: após a criação de uma nova instância, o botão Conectar > guia Gerenciador de Sessões no console do Amazon Elastic Compute Cloud (Amazon EC2) não oferece a opção de conexão.

Solução A: crie um perfil de instância: se ainda não tiver feito isso (conforme instruído pelas informações na guia Gerenciador de sessões no console do EC2), crie um perfil de instância do AWS Identity and Access Management (IAM) usando a Quick Setup. A Quick Setup é uma ferramenta do AWS Systems Manager.

O Session Manager requer um perfil de instância do IAM para se conectar à instância. É possível criar um perfil de instância e atribuí-lo à instância criando uma configuração de gerenciamento de host com a Quick Setup. Uma configuração de gerenciamento de host cria um perfil de instância com as permissões necessárias e o atribui à instância. Uma configuração de gerenciamento de host também habilita outras ferramentas do Systems Manager e cria perfis do IAM para executar essas ferramentas. Não há cobrança pelo uso da Quick Setup ou pelas ferramentas habilitadas pela configuração de gerenciamento do host. Abra a Quick Setup e crie uma configuração de gerenciamento de host.

> **Important**
>
> Depois de criar a configuração de gerenciamento do host, o Amazon EC2 pode levar vários minutos para registrar a alteração e atualizar a guia Gerenciador de Sessões. Se a guia

não exibir o botão Conectar após dois ou três minutos, reinicialize a instância. Após a reinicialização, caso ainda não veja a opção de se conectar, abra a Configuração Rápida e verifique se você tem apenas uma configuração de gerenciamento de host. Se houver duas, exclua a configuração mais antiga e espere alguns minutos.

Se mesmo assim você não conseguir se conectar depois de criar uma configuração de gerenciamento de host ou se receber um erro, incluindo um erro sobre o SSM Agent, consulte uma destas soluções:

- Solução B: não há erro, mas ainda não consigo conectar
- Solução C: erro de SSM Agent ausente

Solução B: não há erro, mas ainda não consigo conectar

Se você criou a configuração de gerenciamento de host, esperou alguns minutos antes de tentar se conectar e ainda não consegue se conectar, talvez seja necessário aplicar a configuração de gerenciamento de host à instância manualmente. Use o procedimento a seguir para atualizar uma configuração de gerenciamento de host da Quick Setup e aplicar alterações em uma instância.

Para atualizar uma configuração de gerenciamento de host usando a Quick Setup

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Quick Setup.
3. Na lista Configurações, escolha a configuração do Gerenciamento de host que você criou.
4. Escolha Ações e depois Editar configuração.
5. Na parte inferior da seção Targets, em Choose how you want to target instances, escolha Manual.
6. Na seção Instâncias, escolha a instância que você criou.
7. Selecione Atualizar.

Aguarde alguns minutos para que o EC2 atualize a guia Gerenciador de Sessões. Se você ainda não conseguir se conectar ou receber um erro, revise as soluções restantes para o problema.

Solução C: erro de SSM Agent ausente

Se você não conseguiu criar uma configuração de gerenciamento de host usando a Quick Setup, ou se recebeu um erro sobre o SSM Agent não estar instalado, talvez seja necessário instalar o SSM Agent manualmente em sua instância. O SSM Agent é um software da Amazon que possibilita que o Systems Manager se conecte à sua instância usando o Session Manager. O SSM Agent é instalado por padrão na maioria das imagens de máquina da Amazon (AMI) (AMIs). Se sua instância foi criada com base em uma AMI não padrão ou em uma AMI mais antiga, talvez seja necessário instalar o agente manualmente. Para o procedimento de instalação do SSM Agent, consulte o tópico a seguir que corresponde ao sistema operacional da instância.

- Windows Server
- macOS
- AlmaLinux
- Amazon Linux 1
- Amazon Linux 2 e AL2023
- CentOS
- CentOS Stream
- Debian Server
- Oracle Linux
- Red Hat Enterprise Linux
- Rocky Linux
- SUSE Linux Enterprise Server
- Ubuntu Server

Para problemas com o SSM Agent, consulte Solução de problemas do SSM Agent.

Sem permissão para iniciar uma sessão

Problema: você tenta iniciar uma sessão, mas o sistema informa que você não tem as permissões necessárias.

- Solução: um administrador do sistema não concedeu permissões de políticas do AWS Identity and Access Management (IAM) para iniciar sessões do Session Manager. Para obter mais informações, consulte Controlar o acesso de sessão do usuário às instâncias.

SSM Agent não está online

Problema: você vê uma mensagem na guia Session Manager da instância do Amazon EC2 que informa: "SSM Agent não está online. O SSM Agent não conseguiu se conectar a um endpoint do Systems Manager para se registrar no serviço".

Solução: SSM Agent é um software da Amazon executado em instâncias do Amazon EC2 para que o Session Manager possa se conectar a elas. Se você vir esse erro, SSM Agent não consegue estabelecer uma conexão com o endpoint do Systems Manager. As causas do problema podem ser restrições de firewall, problemas de roteamento ou falta de conectividade com a Internet. Para solucionar esse problema, investigue os problemas de conectividade de rede. Para obter mais informações, consulte Solução de problemas do SSM Agent e Solução de problemas de disponibilidade do nó gerenciado. Para obter informações sobre endpoints do Systems Manager, consulte Endpoints e cotas do AWS Systems Manager na Referência Geral da AWS.

Sem permissão para alterar as preferências da sessão

Problema: você tenta atualizar as preferências de sessões globais para sua organização, mas o sistema informa que você não tem as permissões necessárias.

- Solução: um administrador do sistema não concedeu permissões de políticas do IAM para definir preferências do Session Manager. Para mais informações, consulte Conceder ou negar permissões a um usuário para atualizar as preferências do Session Manager.

Nó gerenciado não disponível ou não está configurado para o Session Manager

Problema 1: você quer iniciar uma sessão na página do console Start a session (Iniciar uma sessão), mas um nó gerenciado não está na lista.

- Solução A: o nó gerenciado ao qual você quer se conectar pode não ter sido configurado para o AWS Systems Manager. Para obter mais informações, consulte Configurar o console unificado do Systems Manager para uma organização.

> **Note**
>
> Se o SSM Agent do AWS Systems Manager já estiver em execução em um nó gerenciado ao associar o perfil da instância do IAM, talvez seja necessário reiniciar o agente antes que a instância seja listada na página do console Start a session (Iniciar uma sessão).

- Solução B: a configuração de proxy que você aplicou ao SSM Agent em seu nó gerenciado pode estar incorreta. Se a configuração do proxy estiver incorreta, o nó gerenciado não conseguirá alcançar os endpoints de serviço necessários ou o poderá ser relatado como um sistema operacional diferente para o Systems Manager. Para obter mais informações, consulte Configurar o SSM Agent para usar um proxy em nós do Linux e Configurar o SSM Agent para usar um proxy para instâncias do Windows Server.

Problema 2: um nó gerenciado ao qual você quer se conectar está na lista da página do console Start a session (Iniciar uma sessão), mas a página informa que "A instância que você selecionou não está configurada para usar o Session Manager".

- Solução A: o nó gerenciado foi configurado para uso com o serviço do Systems Manager, mas o perfil da instância do IAM anexado ao nó pode não incluir permissões para a ferramenta Session Manager. Para obter informações, consulte Verificar ou criar um perfil da instância do IAM com permissões do Session Manager.
- Solução B: o nó gerenciado não está executando uma versão do SSM Agent que oferece suporte ao Session Manager. Atualize o SSM Agent do nó para a versão 2.3.68.0 ou posterior.

  Atualize o SSM Agent manualmente em um nó gerenciado, seguindo as etapas em Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Windows Server, Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux ou Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para macOS, dependendo do sistema operacional.

  Como alternativa, use o documento do Run Command, `AWS-UpdateSSMAgent`, para atualizar a versão do agente em um ou mais nós gerenciados de uma vez. Para mais informações, consulte Atualização do SSM Agent por meio de Run Command.

  > **Tip**
  >
  > Para manter o agente sempre atualizado, recomendamos atualizar SSM Agent para a versão mais recente através de um agendamento automatizado que você define usando um dos seguintes métodos:
  >
  > - Executar `AWS-UpdateSSMAgent` como parte de uma associação do State Manager. Para mais informações, consulte Walkthrough: Automatically update SSM Agent with the AWS CLI.

- Execute `AWS-UpdateSSMAgent` como parte de uma janela de manutenção. Para obter informações sobre como trabalhar com janelas de manutenção, consulte Crie e gerencie janelas de manutenção usando o console e Tutorial: Criar e configurar uma janela de manutenção usando a AWS CLI.

- Solução C: o nó gerenciado não consegue alcançar os endpoints de serviço necessários. Você pode melhorar o procedimento de segurança dos nós gerenciados usando endpoints de interface habilitados pelo AWS PrivateLink para se conectar aos endpoints do Systems Manager. A alternativa ao uso de endpoints da interface é permitir o acesso à Internet de saída em seus nós gerenciados. Para obter mais informações, consulte Usar PrivateLink para configurar um endpoint da VPC para o Session Manager.
- Solução D: o nó gerenciado tem recursos limitados de CPU ou memória disponíveis. Embora o nó gerenciado possa ser funcional, se ele não tiver recursos disponíveis suficientes, você não poderá estabelecer uma sessão. Para obter mais informações, consulte Solucionar problemas de uma instância inacessível.

O plugin Session Manager não foi encontrado

Para usar a AWS CLI para executar comandos de sessão, o plugin do Session Manager também deve ser instalado em sua máquina local. Para mais informações, consulte Instalar o plug-in do Session Manager para a AWS CLI.

O plug-in Session Manager não é adicionado automaticamente ao caminho de linha de comando (Windows)

Quando você instalar o plug-in do Session Manager no Windows, o `session-manager-plugin` executável deverá ser adicionado automaticamente à variável de ambiente PATH do sistema operacional. Se o comando falhou depois que você o executou para verificar se o plugin do Session Manager foi instalado corretamente (`aws ssm start-session --target instance-id`), pode ser necessário configurá-lo manualmente usando o procedimento a seguir.

Para modificar sua variável PATH (Windows)

1. Pressione a tecla Windows e digite **`environment variables`**.
2. Escolha Edit environment variables for your account (Editar variáveis de ambiente para sua conta).
3. Selecione PATH e, em seguida, Editar.

4. Adicione caminhos ao campo Variable value (Valor da variável) separados por ponto e vírgula, conforme este exemplo: *C:\existing\path*;***C:\new\path***

   C:\existing\path representa o valor já no campo. ***C:\new\path*** representa o caminho que você deseja adicionar, como nestes exemplos.

   - Máquinas 64-bit: `C:\Program Files\Amazon\SessionManagerPlugin\bin\`
   - Máquinas 32-bit: `C:\Program Files (x86)\Amazon\SessionManagerPlugin\bin\`
5. Escolha OK duas vezes para aplicar as novas configurações.
6. Feche todas as solicitações de comando em execução e abra novamente.

O plugin Session Manager não responde

Durante uma sessão de encaminhamento de porta, o tráfego poderá interromper o encaminhamento se você tiver um software antivírus instalado em sua máquina local. Em alguns casos, o software antivírus interfere com o plugin Session Manager causando bloqueios de processo. Para resolver esse problema, permita ou exclua o plugin do Session Manager do software antivírus. Para obter informações sobre o caminho de instalação padrão para o plugin Session Manager, consulte Instalar o plug-in do Session Manager para a AWS CLI.

TargetNotConnected

Problema: você tenta iniciar uma sessão, mas o sistema retorna a mensagem de erro "Ocorreu um erro (TargetNotConnected) ao chamar a operação StartSession: *InstanceID* não está conectado".

- Solução A: esse erro é retornado quando o nó gerenciado de destino especificado para a sessão não estiver totalmente configurado para uso com o Session Manager. Para mais informações, consulte Configurar o Session Manager.
- Solução B: esse erro também ocorre se você tentar iniciar uma sessão em um nó gerenciado localizado em uma Conta da AWS ou Região da AWS diferente.

Tela em branco exibida após iniciar uma sessão

Problema: você inicia uma sessão e o Session Manager exibe uma tela em branco.

- Solução A: esse problema pode ocorrer quando o volume raiz em seu nó gerenciado estiver cheio. Devido à falta de espaço em disco, o SSM Agent do nó para de funcionar. Para resolver esse problema, use o Amazon CloudWatch para coletar métricas e logs dos sistemas operacionais.

Para obter informações, consulte Coletar métricas, logs e rastreamentos com o agente do CloudWatch no Guia do usuário do Amazon CloudWatch.

- Solução B: uma tela em branco pode ser exibida se você acessou o console usando um link que inclui um par de endpoint e região incompatíveis. Por exemplo, no URL do console a seguir, `us-west-2` é o endpoint especificado, mas `us-west-1` é a região da Região da AWS especificada.

```
https://us-west-2.console.aws.amazon.com/systems-manager/session-manager/sessions?
region=us-west-1
```

- Solução C: o nó gerenciado está se conectando ao Systems Manager usando endpoints da VPC, e suas preferências do Session Manager gravam a saída da sessão em um bucket do Amazon S3 ou grupo de logs do Amazon CloudWatch Logs, mas um endpoint de gateway do `s3` ou endpoint de interface do `logs` não existe na VPC. Um endpoint do `s3` no formato **`com.amazonaws.`*`region`*`.s3`** será necessário se os nós gerenciados estiverem conectados ao Systems Manager usando endpoints da VPC e suas preferências do Session Manager gravarem o resultado da sessão em um bucket do Amazon S3. Como alternativa, um endpoint do `logs` no formato **`com.amazonaws.`*`region`*`.logs`** será necessário, se os nós gerenciados estiverem se conectando ao Systems Manager usando endpoints da VPC, e as preferências do Session Manager gravarem a saída da sessão em um grupo de logs do CloudWatch Logs. Para obter mais informações, consulte Criar endpoints da VPC para o Systems Manager.
- Solução D: o grupo de logs ou bucket do Amazon S3 que você especificou nas preferências de sessão foi excluído. Para resolver esse problema, atualize suas preferências de sessão com um grupo de logs válido ou um bucket S3.
- Solução E: o grupo de logs ou bucket do Amazon S3 que você especificou em suas preferências de sessão não está criptografado, mas você definiu a entrada `cloudWatchEncryptionEnabled` ou `s3EncryptionEnabled` como `true`. Para resolver esse problema, atualize suas preferências de sessão com um grupo de logs ou bucket do Amazon S3 criptografado ou defina a entrada `cloudWatchEncryptionEnabled` ou `s3EncryptionEnabled` como `false`. Esse cenário só é aplicável aos clientes que criam preferências de sessão usando ferramentas da linha de comando.

O nó gerenciado deixa de responder durante sessões de execução longa

Problema: seu nó gerenciado deixa de responder ou falha durante uma sessão de execução longa.

Solução: diminuir a duração da retenção de logs do SSM Agent para Session Manager.

Para diminuir a duração da retenção de logs do SSM Agent para sessões

1. Localize `amazon-ssm-agent.json.template` no diretório `/etc/amazon/ssm/` no Linux, ou em `C:\Program Files\Amazon\SSM` no Windows.
2. Copie o conteúdo do `amazon-ssm-agent.json.template` em um novo arquivo no mesmo diretório chamado `amazon-ssm-agent.json`.
3. Reduza o valor padrão do valor `SessionLogsRetentionDurationHours` na propriedade `SSM` e salve o arquivo.
4. Reinicie o SSM Agent.

Ocorreu um erro (InvalidDocument) ao chamar a operação StartSession

Problema: você recebe o seguinte erro ao iniciar uma sessão usando o AWS CLI.

```
An error occurred (InvalidDocument) when calling the StartSession operation: Document
 type: 'Command' is not supported. Only type: 'Session' is supported for Session
 Manager.
```

Solução: o documento SSM que você especificou para o `--document-name` parâmetro não é um documento de sessão. Use o procedimento a seguir para visualizar uma lista de documentos da sessão no AWS Management Console.

Para visualizar uma lista de documentos da sessão

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Documents.
3. Na lista Categorias, escolha Documentos da sessão.

AWS Systems Manager State Manager

O State Manager, uma ferramenta do AWS Systems Manager, é um serviço seguro e escalável de gerenciamento de configuração que automatiza o processo de manter seus nós gerenciados e outros recursos da AWS em um estado definido por você. Para começar a usar o State Manager, abra o Systems Manager console (Console do gerenciador de sistemas). No painel de navegação, escolha State Manager.

> **Note**
>
> O State Manager e Maintenance Windows podem executar alguns tipos semelhantes de atualizações nos seus nós gerenciados. Qual deles escolher depende se você precisa automatizar a conformidade do sistema ou executar tarefas de alta prioridade e sensíveis ao tempo durante os períodos especificados.
> Para obter mais informações, consulte Selecionar entre State Manager e Maintenance Windows.

Como o State Manager beneficia minha organização?

Ao usar documentos pré-configurados do Systems Manager (documentos SSM), o State Manager oferece os seguintes benefícios para gerenciar seus nós:

- Fazer o bootstrap de nós com softwares específicos na inicialização
- Baixar e atualizar agentes em uma programação definida, incluindo o SSM Agent.
- Definir configurações de rede.
- Integrar nós a um domínio do Microsoft Active Directory.
- Executar scripts no Linux, macOS e nós gerenciados Windows ao longo do ciclo de vida.

Para gerenciar o desvio de configuração em outros recursos da AWS, você pode usar o Automation, uma ferramenta do Systems Manager, com o State Manager para realizar os seguintes tipos de tarefas:

- Anexar uma função do Systems Manager às instâncias do Amazon Elastic Compute Cloud (Amazon EC2) para torná-las instâncias gerenciadas.
- Aplicar regras de entrada e saída desejadas a um grupo de segurança.
- Crie ou exclua backups do Amazon DynamoDB.
- Crie ou exclua snapshots do Amazon Elastic Block Store (Amazon EBS).
- Desative as permissões de leitura e gravação em buckets do Amazon Simple Storage Service (Amazon S3).
- Iniciar, reiniciar ou interromper nós gerenciados e instâncias do Amazon Relational Database Service (Amazon RDS).
- Aplique patches às macOS do Linux, AMIs e Windows.

Para obter informações sobre como o uso do State Manager com runbooks do Automation, consulte Programação de automações com associações do State Manager.

Quem deve usar o State Manager?

O State Manager é apropriado para qualquer cliente da AWS que deseja melhorar a gestão e a governança de seus recursos da AWS e reduzir o desvio de configuração.

Quais são os recursos do State Manager?

Os principais recursos do State Manager incluem o seguinte:

- Associações do State Manager

  Uma associação do State Manager é uma configuração que você atribui aos seus recursos da AWS. A configuração define o estado que você deseja manter em seus recursos. Por exemplo, uma associação pode especificar que o software antivírus deve estar instalado e em execução nas instâncias ou que determinadas portas devem ser fechadas.

  Uma associação especifica uma programação para quando aplicar a configuração e destinos para a associação. Por exemplo, uma associação a um software antivírus pode ser executada uma vez por dia em uma Conta da AWS. Se o software não estiver instalado em um nó, a associação poderá instruir o State Manager para instalá-lo. Se o software estiver instalado, mas o serviço não estiver em execução, a associação poderá instruir o State Manager a iniciar o serviço.
- Opções flexíveis de agendamento

  O State Manager oferece as seguintes opções para agendamento quando uma associação é executada:

  - Processamento imediato ou atrasado

    Quando você cria uma associação, por padrão, o sistema a executa imediatamente nos recursos especificados. Após a execução inicial, a associação é executada em intervalos de acordo com a programação que você definiu.

    Você pode instruir State Manager para não executar uma associação imediatamente usando a opção Apply association only at the next specified Cron interval (Aplicar a associação do somente no próximo intervalo de Cron especificado) no console ou o parâmetro `ApplyOnlyAtCronInterval` a partir da linha de comando.
  - Expressões cron e rate

Ao criar uma associação, você especifica uma programação para quando o State Manager aplicar a configuração. O State Manager oferece suporte a maioria das expressões padrão cron e rate para agendamento quando uma associação é executada. O State Manager também oferece suporte a expressões cron que incluem um dia da semana e o sinal numérico (#) para designar o nº dia de um mês para executar uma associação e o sinal (L) para indicar o último dia X do mês.

> **Note**
>
> Atualmente, o State Manager não oferece suporte à especificação de meses em expressões cron para associações.

Para controlar ainda mais quando uma associação é executada, por exemplo, se você quiser executar uma associação dois dias após o patch de terça-feira, você pode especificar um deslocamento. Um deslocamento define quantos dias esperar após o dia programado para executar uma associação.

Para obter mais informações sobre criar expressões cron e rate, consulte Referência: Expressões cron e rate para o Systems Manager.

- Várias opções de direcionamento

  Uma associação também especifica os destinos da associação. O State Manager oferece suporte à definição de destinos de recursos do AWS usando tags, AWS Resource Groups, IDs de nó individuais ou todos os nós gerenciados no atual Região da AWS e Conta da AWS.

- Suporte ao Amazon S3

  Armazene a saída do comando de execuções de associação em um bucket do Amazon S3 de sua escolha. Para obter mais informações, consulte Para obter informações, consulte Trabalhar com associações no Systems Manager..

- Suporte ao EventBridge

  Essa ferramenta do Systems Manager é compatível com um tipo evento e um tipo destino nas regras do Amazon EventBridge. Para obter informações, consulte Monitorar eventos do Systems Manager com o Amazon EventBridge e Referência: padrões e tipos de eventos do Amazon EventBridge para o Systems Manager.

Há cobrança pelo uso do State Manager?

O State Manager está disponível sem custo adicional.

Tópicos

- Como funciona o State Manager
- Para obter informações, consulte Trabalhar com associações no Systems Manager.
- Criar associações que executam arquivos MOF
- Criar associações que executam manuais do Ansible
- Criar associações que executam receitas do Chef
- Walkthrough: Automatically update SSM Agent with the AWS CLI
- Walkthrough: Automatically update PV drivers on EC2 instances for Windows Server

Mais informações

- Combater o deslocamento usando o Amazon EC2 Systems Manager e Windows PowerShell DSC
- Configure as instâncias do Amazon EC2 em um grupo de Auto Scaling usando o State Manager

Como funciona o State Manager

O State Manager, uma ferramenta do AWS Systems Manager, é um serviço seguro e escalável que automatiza o processo de manter nós gerenciados em uma infraestrutura híbrida e multinuvem em um estado definido por você.

Como o State Manager funciona:

1. Determine o estado que deseja aplicar a seus recursos da AWS.

   Deseja garantir que seus nós gerenciados sejam configurados com aplicações específicas, como antivírus ou aplicações de malware? Você deseja automatizar o processo de atualização do SSM Agent ou outros pacotes da AWS, como `AWSPVDriver`? Você precisa garantir que determinadas portas sejam fechadas ou abertas? Para começar a usar o State Manager, determine o estado que deseja aplicar a seus recursos da AWS. O estado que você deseja aplicar determina qual documento SSM é usado para criar uma associação do State Manager.

Uma associação do State Manager é uma configuração que você atribui aos seus recursos da AWS. A configuração define o estado que você deseja manter em seus recursos. Por exemplo, uma associação pode especificar que o software antivírus deve estar instalado e em execução nas instâncias ou que determinadas portas devem ser fechadas.

Uma associação especifica uma programação para quando aplicar a configuração e destinos para a associação. Por exemplo, uma associação a um software antivírus pode ser executada uma vez por dia em uma Conta da AWS. Se o software não estiver instalado em um nó, a associação poderá instruir o State Manager para instalá-lo. Se o software estiver instalado, mas o serviço não estiver em execução, a associação poderá instruir o State Manager a iniciar o serviço.

2. Determine se um documento SSM pré-configurado pode ajudar você a criar o estado desejado nos seus recursos da AWS.

O Systems Manager inclui dezenas de documentos SSM pré-configurados que você pode usar para criar uma associação. Os documentos pré-configurados estão prontos para executar tarefas comuns, como instalação de aplicações, configuração do Amazon CloudWatch, execução de automações do AWS Systems Manager, execução de scripts PowerShell e Shell e ingresso de nós gerenciados um domínio do de serviço de diretórios para Active Directory.

Você pode visualizar todos os documentos do console do Systems Manager. Selecione o nome de um documento para saber mais sobre cada um. Veja estes dois exemplos: `AWS-ConfigureAWSPackage` e `AWS-InstallApplication`.

3. Criar uma associação.

Você pode criar uma associação usando o console do Systems Manager, a AWS Command Line Interface (AWS CLI), o AWS Tools for Windows PowerShell (Tools for Windows PowerShell) ou a API do Systems Manager. Ao criar uma associação, você especifica as seguintes informações:

- Um nome para a associação.
- Os parâmetros do documento do SSM (por exemplo, o caminho para a aplicação a ser instalada ou o script a ser executado nos nós).
- Destinos para a associação. Você pode direcionar nós gerenciados especificando tags, escolhendo IDs de nós individuais ou escolhendo um grupo no AWS Resource Groups. Você também pode direcionar todos os nós gerenciados na Região da AWS e na Conta da AWS atuais. Se seus destinos incluírem mais de 1.000 nós, o sistema usa um mecanismo de controle de utilização por hora. Isso significa que você pode ver imprecisões na contagem de

agregação de status, já que o processo de agregação é executado a cada hora e somente quando o status de execução de um nó muda.

- Uma programação para quando ou com que frequência aplicar o estado. Você pode especificar uma expressão cron ou rate. Para obter mais informações sobre como criar programações usando expressões cron e rate, consulte Expressões cron e rate para associações.

> **Note**
>
> Atualmente, o State Manager não oferece suporte à especificação de meses em expressões cron para associações.

Quando você executa o comando para criar a associação, o Systems Manager vincula as informações especificadas (programação, destinos, documento do SSM e parâmetros) aos recursos direcionados. O status da associação mostra "Pending" inicialmente, pois o sistema tenta acessar todos os destinos e aplicar imediatamente o estado especificado na associação.

> **Note**
>
> Se criar uma nova associação programada para execução enquanto uma associação anterior ainda estiver em execução, a associação anterior atingirá o tempo limite e a nova será executada.

O Systems Manager informa o status da solicitação para criar associações nos recursos. Você pode visualizar os detalhes do status no console ou (para nós gerenciados) usando a operação da API DescribeInstanceAssociationsStatus. Se você escolher gravar a saída do comando no Amazon Simple Storage Service (Amazon S3) ao criar uma associação, também poderá visualizar a saída no bucket do Amazon S3 que você especificar.

Para obter mais informações, consulte Para obter informações, consulte Trabalhar com associações no Systems Manager..

> **Note**
>
> As operações de API iniciadas pelo documento SSM durante uma execução de associação não são registradas em log no AWS CloudTrail.

4. Monitore e atualize.

   Depois de criar a associação, o State Manager reaplica a configuração de acordo com o agendamento definido na associação. Você pode visualizar o status das suas associações na página do State Manager no console ou chamando diretamente o ID da associação gerado pelo Systems Manager quando você criou a associação. Para obter mais informações, consulte Visualizar históricos de associação. Você pode atualizar seus documentos de associação e reaplicá-los conforme necessário. Você pode criar também várias versões de uma associação. Para obter mais informações, consulte Edita e criar uma nova versão de uma associação.

Noções básicas sobre quando associações são aplicadas a recursos

Ao criar uma associação, você especifica um documento do SSM que define a configuração, uma lista de recursos de destino e uma programação para a aplicação da configuração. Por padrão, o State Manager executa a associação quando você a cria e, a partir daí, de acordo com a programação especificada. O State Manager também tenta executar a associação nas seguintes situações:

- Edição da associação: o State Manager executa a associação depois que um usuário faz uma edição e salva as alterações em qualquer um dos seguintes campos de associação: `DOCUMENT_VERSION`, `PARAMETERS`, `SCHEDULE_EXPRESSION`, `OUTPUT_S3_LOCATION`.
- Edição de documentos: o State Manager executa a associação depois que um usuário faz uma edição e salva as alterações no documento do SSM que define o estado da configuração da associação. Especificamente, a associação é executada após as seguintes edições no documento:
  - Um usuário especifica uma nova versão do documento `$DEFAULT` e a associação foi criada usando a versão `$DEFAULT`.
  - Um usuário atualiza um documento e a associação foi criada usando a versão `$LATEST`.
  - Um usuário exclui o documento que foi especificado quando a associação foi criada.
- Início manual: o State Manager executa a associação quando ela é iniciada pelo usuário no console do Systems Manager ou programaticamente.
- Alterações no destino: o State Manager executa a associação após qualquer uma das seguintes atividades ocorrer em um ó de destino:
  - Um nó gerenciado entra online pela primeira vez.
  - Um nó gerenciado entra online após perder a execução de uma associação programada.
  - Um nó gerenciado entra online após permanecer parado por mais de 30 dias.

> **Note**
>
> O State Manager não monitora documentos ou pacotes usados em associações entre Contas da AWS. Se você atualizar um documento ou pacote em uma conta, a atualização não fará com que a associação seja executada na segunda conta. Você deve executar manualmente a associação na segunda conta.

Como impedir que associações sejam executadas quando um destino muda

Em alguns casos, talvez você não queira que uma associação seja executada quando um destino que consiste em nós gerenciados muda, mas somente de acordo com a programação especificada.

> **Note**
>
> A execução de um runbook do Automation gera custos. Se uma associação com um runbook do Automation tem como alvo todas as instâncias em sua conta e você inicia regularmente um grande número de instâncias, o runbook é executado em cada uma das instâncias quando ela é iniciada. Isso pode levar a cobranças elevadas de automação.

Para impedir que uma associação seja executada quando os destinos dessa associação mudarem, selecione a caixa de seleção Aplicar associação somente no próximo intervalo cron especificado. Essa caixa de seleção está localizada na área Especificar programação das páginas Criar associação e Editar associação.

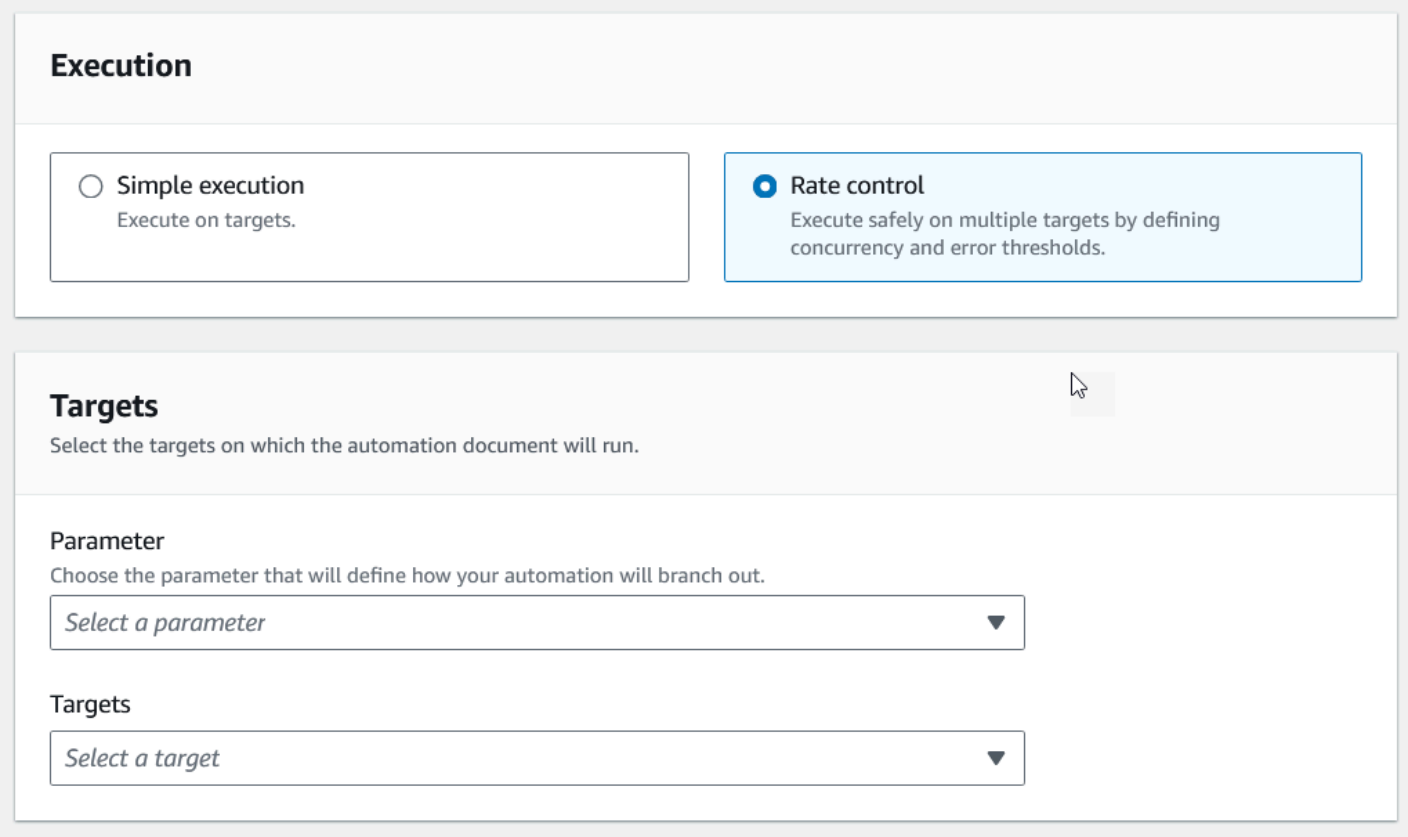
Essa opção se aplica a associações que incorporam um runbook do Automation ou um documento SSM.

Sobre as atualizações de destino com runbooks do Automation

Para que as associações criadas com os runbooks do Automation sejam aplicadas quando novos nós de destino forem detectados, as seguintes condições devem ser verdadeiras:

- A associação deve ter sido criada por uma configuração da Quick Setup. A Quick Setup é uma ferramenta do AWS Systems Manager. No momento, não há suporte a associações criadas por outros processos.
- O runbook do Automation deve ter como alvo explícito o tipo de recurso `AWS::EC2::Instance` ou `AWS::SSM::ManagedInstance`.
- A associação deve especificar os parâmetros e os destinos.

  No console, os campos Parâmetro e Destinos são exibidos quando você escolhe uma execução de controle de taxa.



  Ao usar as ações da API CreateAssociation, CreateAssociationBatch ou UpdateAssociation, é possível especificar esses valores usando as entradas `AutomationTargetParameterName` e `Targets`. Em cada uma dessas ações de API, você também pode impedir que a associação seja executada sempre que um destino for modificado definindo o parâmetro `ApplyOnlyAtCronInterval` como `true`.

  Para obter informações sobre como usar o console para controlar quando as associações são executadas, incluindo detalhes para evitar custos inesperadamente altos para execuções do Automation, consulte Noções básicas sobre quando associações são aplicadas a recursos.

Para obter informações, consulte Trabalhar com associações no Systems Manager.

Esta seção descreve como criar e gerenciar associações do State Manager usando o console do AWS Systems Manager, a AWS Command Line Interface (AWS CLI) e o Ferramentas da AWS para PowerShell.

Tópicos

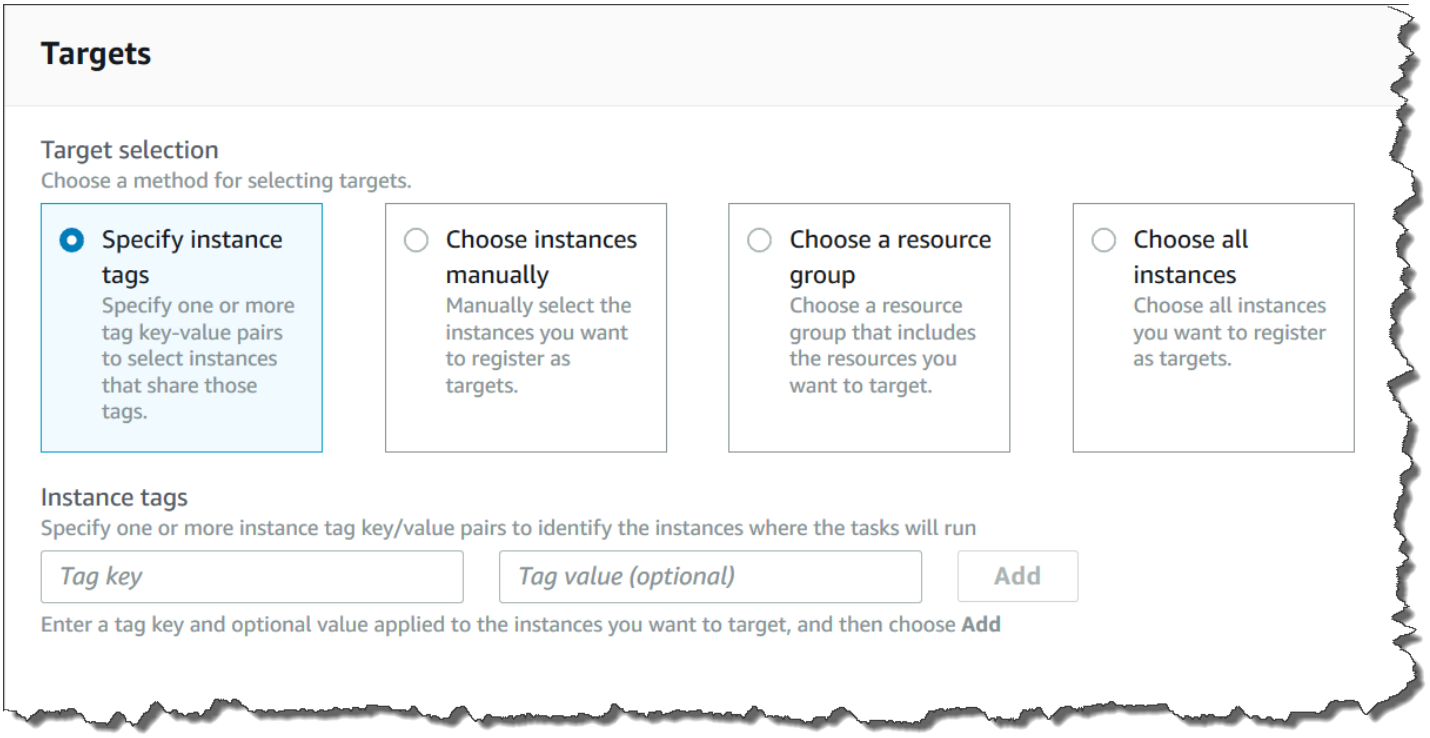
- Sobre destinos e controles de taxa em associações do State Manager
- Criar associações
- Edita e criar uma nova versão de uma associação
- Excluir associações
- Executar grupos do Auto Scaling com associações
- Visualizar históricos de associação
- Trabalhar com associações usando o IAM

Sobre destinos e controles de taxa em associações do State Manager

Este tópico descreve recursos do State Manager que ajudam você a implantar uma associação em dezenas ou centenas de instâncias enquanto controla quantos nós executam a associação no horário programado. O State Manager é uma ferramenta do AWS Systems Manager.

Uso de destinos

Ao criar uma associação do State Manager, você escolhe quais nós serão configurados com a associação na seção Targets (Destinos) do console do Systems Manager, conforme mostrado aqui.



Se você criar uma associação usando uma ferramenta da linha de comando, como a AWS Command Line Interface (AWS CLI), especifique o parâmetro `targets`. A segmentação de nós permite configurar dezenas, centenas ou milhares de nós com uma associação sem precisar especificar ou escolher IDs de nós individuais.

Cada nó gerenciado pode ser destino de, no máximo, 20 associações.

O State Manager inclui as seguintes opções de destino ao criar uma associação.

Especificar tags

Use esta opção para especificar uma chave de tag e (opcionalmente) um valor de tag que está atualmente atribuído aos seus nós. Quando você executa a solicitação, o sistema localiza e tenta criar a associação em todos os nós que correspondem à chave e ao valor de tag especificados. Se você especificou vários valores de tag, a associação se destina a qualquer nó com pelo menos um desses valores de tag. Quando o sistema cria a associação inicialmente, ele executa a associação. Após essa execução inicial, o sistema executa a associação de acordo com a programação especificada.

Se você criar novos nós e atribuir a chave e o valor de tag especificados a esses nós, o sistema aplicará automaticamente a associação, a executará imediatamente e, depois, a executará de acordo com a programação. Isso se aplica quando a associação usa um documento de comando

ou política e não se aplica se a associação usa um runbook de automação. Se você excluir as tags especificadas de um nó, o sistema não executará mais a associação nesses nós.

> **Note**
>
> Se você usar runbooks de automação com o State Manager e a limitação de marcação impedir que você atinja uma meta específica, considere o uso de runbooks de automação com o Amazon EventBridge. Para obter mais informações, consulte Executar automações com base em eventos do EventBridge. Para obter informações sobre como usar runbooks com o State Manager, consulte Programação de automações com associações do State Manager.

Como prática recomendada, é aconselhável o uso de etiquetas ao criar associações que usam um documento do Command ou Policy. Também é recomendável usar etiquetas ao criar associações para executar grupos do Auto Scaling. Para obter mais informações, consulte Executar grupos do Auto Scaling com associações.

> **Note**
>
> Observe as seguintes informações:
>
> - Ao criar uma associação no AWS Management Console que direciona nós usando tags, você pode especificar apenas uma chave de tag para uma associação de automação e cinco chaves de tags para uma associação de comando. Todas as chaves de tags especificadas na associação deverão estar atualmente atribuídas ao nó. Se não estiverem, o State Manager falhará em direcionar o nó para uma associação.
> - Se quiser usar o console e direcionar os nós usando mais de uma chave de tag para uma associação de automação e cinco chaves de tags para uma associação de comando, atribua as chaves de tags a um grupo do AWS Resource Groups e adicione os nós a ele. Em seguida, você pode escolher a opção Grupo de recursos na lista de Destinos ao criar a associação do State Manager.
> - É possível especificar no máximo cinco chaves de etiqueta usando a AWS CLI. Se você usar a AWS CLI, todas as chaves de etiqueta especificadas no comando `create-association` deverão estar atualmente atribuídas ao nó. Se não estiverem, o State Manager falhará em direcionar o nó para uma associação.

Selecione os nós manualmente

Use esta opção para selecionar manualmente os nós em que deseja criar a associação. O painel Instances (Instâncias) exibe todos os nós gerenciados do Systems Manager na Conta da AWS e Região da AWS atuais. Você pode selecionar manualmente quantos nós desejar. Quando o sistema cria a associação inicialmente, ele executa a associação. Após essa execução inicial, o sistema executa a associação de acordo com a programação especificada.

> **Note**
>
> Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

Escolher um grupo de recursos

Use esta opção para criar uma associação em todos os nós retornados por uma consulta baseada em tag do AWS Resource Groups ou em pilha do AWS CloudFormation.

Observe os detalhes abaixo sobre como segmentar grupos de recursos para uma associação.

- Se você adicionar novos nós a um grupo, o sistema mapeará automaticamente os nós para a associação direcionada ao grupo de recursos. O sistema aplica a associação aos nós quando descobre a alteração. Após essa execução inicial, o sistema executa a associação de acordo com a programação especificada.
- Se você criar uma associação cujo destino seja um grupo de recursos e o tipo de recurso `AWS::SSM::ManagedInstance` tiver sido especificado para esse grupo, intencionalmente, a associação será executada em instâncias do Amazon Elastic Compute Cloud (Amazon EC2) e em nós que não são do EC2 em um ambiente híbrido e multinuvem.

  A recíproca também é verdadeira. Se você criar uma associação cujo destino seja um grupo de recursos e o tipo de recurso `AWS::EC2::Instance` tiver sido especificado para esse grupo, intencionalmente, a associação será executada em nós que não são do EC2 em um ambiente híbrido e multinuvem e em instâncias (Amazon EC2).
- Se você criar uma associação destinada a um grupo de recursos, o grupo de recursos não deverá ter mais de cinco chaves de tag atribuídas a ele ou mais de cinco valores especificados para qualquer chave de tag. Se uma dessas condições se aplicar às tags e chaves atribuídas ao seu grupo de recursos, a associação falhará ao ser executada e retornará um erro `InvalidTarget`.

- Se você criar uma associação que tenha como destino um grupo de recursos usando tags, não poderá escolher a opção (empty value) para o valor da tag.
- Se você excluir um grupo de recursos, todas as instâncias desse grupo não executarão mais a associação. Como prática recomendada, exclua as associações direcionadas ao grupo.
- Você só pode direcionar no máximo um único grupo de recursos para uma associação. Vários grupos ou grupos aninhados não são permitidos.
- Depois de criar uma associação, o State Manager atualiza periodicamente a associação com informações sobre recursos no grupo de recursos. Se você adicionar novos recursos a um grupo de recursos, a programação para quando o sistema aplica a associação aos novos recursos dependerá de vários fatores. Você pode determinar o status da associação na página State Manager do console do Systems Manager.

> **Warning**
>
> Um usuário do AWS Identity and Access Management (IAM), grupo ou função com permissão para criar uma associação direcionada a um grupo de recursos de instâncias do Amazon EC2 automaticamente tem controle no nível raiz de todas as instâncias do grupo. Somente administradores confiáveis devem ter permissão para criar associações.

Para obter mais informações sobre os grupos de recursos, consulte O que é o AWS Resource Groups? no Guia do usuário do AWS Resource Groups.

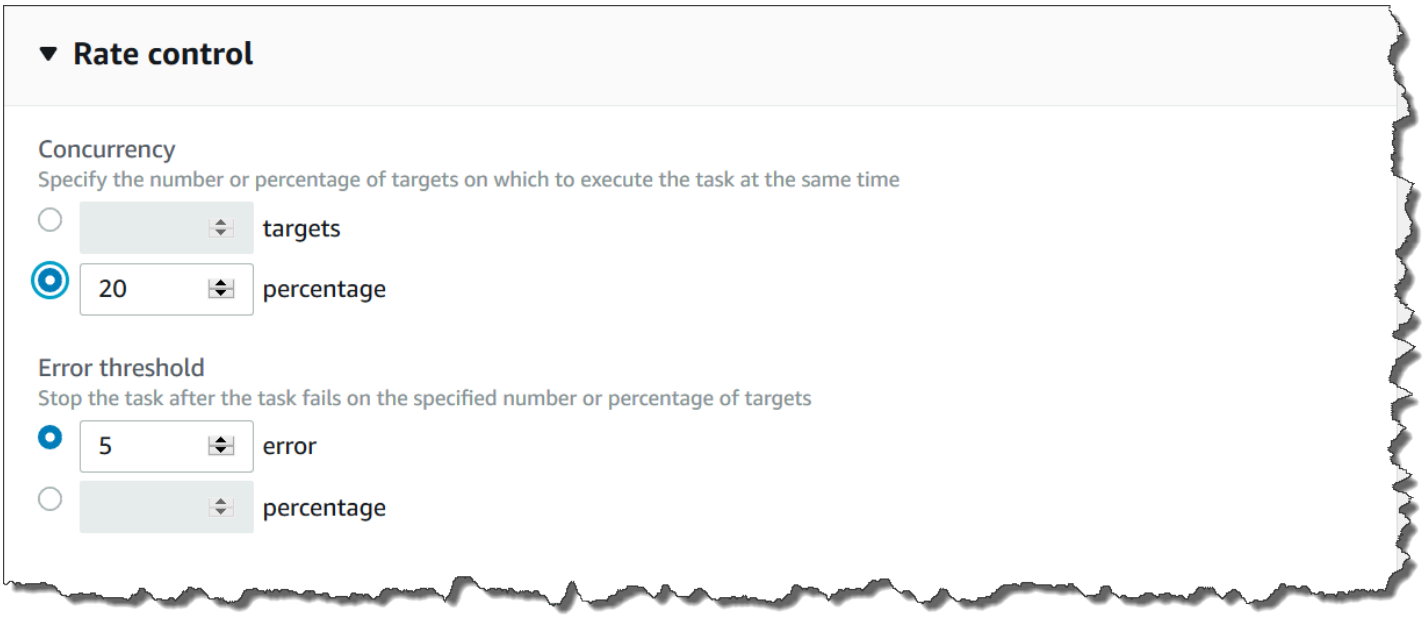
Escolher todos os nós

Use esta opção para direcionar todos os nós na Conta da AWS e Região da AWS atuais. Quando você executa a solicitação, o sistema localiza e tenta criar a associação em todos os nós na Conta da AWS e Região da AWS atuais. Quando o sistema cria a associação inicialmente, ele executa a associação. Após essa execução inicial, o sistema executa a associação de acordo com a programação especificada. Se você criar novos nós, o sistema aplicará automaticamente a associação, a executará imediatamente e, depois, a executará de acordo com a programação.

Usar controles de taxa

Você pode controlar a execução de uma associação em seus nós especificando um valor de simultaneidade e um limite de erro. O valor de simultaneidade especifica quantos nós podem executar a associação simultaneamente. Um limite de erro especifica quantas execuções

de associação podem falhar antes de o Systems Manager enviar um comando para cada nó configurado com essa associação para interromper a associação. O comando interrompe a execução da associação até a próxima execução programada. Os recursos de simultaneidade e limite de erros são coletivamente chamados de controles de taxa.



Simultaneidade

A simultaneidade ajuda a limitar o impacto nos seus nós, permitindo que você especifique que apenas um determinado número de nós pode processar uma associação de uma só vez. Você pode especificar um número absoluto de nós, por exemplo, 20, ou um percentual do conjunto de nós destino, por exemplo, 10%.

A simultaneidade do State Manager tem as seguintes restrições e limitações:

- Se você optar por criar uma associação usando destinos, mas não especificar um valor de simultaneidade, o State Manager aplicará automaticamente um máximo de 50 nós de simultaneidade.
- Se novos nós que correspondem aos critérios de destino ficarem online enquanto uma associação que usa simultaneidade estiver em execução, os novos nós executarão a associação se o valor de simultaneidade não for excedido. Se o valor de simultaneidade for excedido, então os nós serão ignorados durante o intervalo de execução da associação atual. Os nós executarão a associação durante o próximo intervalo programado, cumprindo os requisitos de simultaneidade.
- Se você atualizar uma associação que usa simultaneidade, e um ou mais nós estiverem processando essa associação quando for atualizada, qualquer nó que estiver executando

a associação terá permissão para ser concluído. As associações que não foram iniciadas serão interrompidas. Após a conclusão da execução das associações, todos os nós de destino imediatamente executam a associação novamente porque ela foi atualizada. Quando a associação é executada novamente, o valor de simultaneidade é aplicado.

Limites de erro

Um limite de erro especifica quantas execuções de associação podem falhar antes de o Systems Manager enviar um comando para cada nó configurado com essa associação. O comando interrompe a execução da associação até a próxima execução programada. Você pode especificar um número absoluto de erros, como 10, ou uma porcentagem do conjunto de destino, como 10%.

Se você especificar um número absoluto de três erros, por exemplo, o State Manager enviará o comando de parada quando o quarto erro for retornado. Se você especificar 0, o State Manager enviará o comando de parada depois que o primeiro resultado do erro for retornado.

Se você especificar um limite de erro de 10% para 50 associações, o State Manager enviará o comando de parada quando o sexto erro for retornado. As associações que já estiverem em execução quando um limite de erro for atingido poderão ser concluídas, mas algumas dessas associações também poderão falhar. Para garantir que não haverá mais erros do que o número especificado para o limite de erros, defina o valor de Concurrency (Simultaneidade) como 1 para que as associações prossigam uma por vez.

Os limites de erro do State Manager têm as seguintes restrições e limitações:

- Os limites de erro são aplicados para o intervalo atual.
- As informações sobre cada erro, incluindo detalhes no nível da etapa, são registradas no histórico da associação.
- Se você optar por criar uma associação usando destinos, mas não especificar um limite de erro, o State Manager aplicará automaticamente um limite de 100% de falhas.

Criar associações

O State Manager, uma ferramenta do AWS Systems Manager, ajuda você a manter seu recursos da AWS em um estado definido por você e a reduzir os desvios de configuração. Para fazer isso, o State Manager usa associações. Uma associação é uma configuração que você atribui aos seus recursos da AWS. A configuração define o estado que você deseja manter em seus recursos. Por

exemplo, uma associação pode especificar que o software antivírus deve estar instalado e em execução nas instâncias ou que determinadas portas devem ser fechadas.

Uma associação especifica uma programação para quando aplicar a configuração e destinos para a associação. Por exemplo, uma associação a um software antivírus pode ser executada uma vez por dia em uma Conta da AWS. Se o software não estiver instalado em um nó, a associação poderá instruir o State Manager para instalá-lo. Se o software estiver instalado, mas o serviço não estiver em execução, a associação poderá instruir o State Manager a iniciar o serviço.

> **Warning**
>
> Ao criar uma associação, você pode escolher um grupo de recursos da AWS de nós gerenciados como o destino da associação. Se um usuário do AWS Identity and Access Management (IAM), grupo ou função tem permissão para criar uma associação direcionada a um grupo de recursos de nós gerenciados, então esse usuário, grupo ou função automaticamente tem controle em nível raiz de todos os nós do grupo. Permitir somente administradores confiáveis criarem associações.

Destino de associação e controles de taxa

Uma associação especifica quais nós gerenciados, ou destinos, devem receber a associação. O State Manager inclui diversos recursos para ajudar você a direcionar os nós gerenciados e controlar como a associação é implantada para esses destinos. Para obter mais informações sobre destinos e controles de taxa, consulte Sobre destinos e controles de taxa em associações do State Manager.

Como marcar associações

É possível atribuir tags a uma associação ao criá-la usando uma ferramenta de linha de comando, como a AWS CLI ou Ferramentas da AWS para PowerShell. Não há suporte à adição de tags a uma associação usando o console do Systems Manager.

Como executar associações

Por padrão, o State Manager executa uma associação imediatamente após a criação e, em seguida, de acordo com a programação que você definiu.

O sistema também executa associações de acordo com as regras a seguir:

- O State Manager tenta executar a associação em todos os nós especificados ou direcionados durante um intervalo.

- Se uma associação não for executada durante um intervalo (porque, por exemplo, um valor de simultaneidade limita o número de nós que podem processar a associação de uma só vez), o State Manager tentará executar a associação durante o próximo intervalo.
- O State Manager executa a associação após alterações na configuração, nos nós de destino, nos documentos ou nos parâmetros da associação. Para obter mais informações, consulte Noções básicas sobre quando associações são aplicadas a recursos.
- O State Manager registra o histórico de todos os intervalos ignorados. Você pode visualizar o histórico na guia Execution History.

Como programar associações

É possível programar associações para serem executadas em intervalos básicos, como a cada dez horas, ou criar programações mais avançadas usando expressões “cron” e “rate” personalizadas. Também é possível impedir que as associações sejam executadas ao criá-las pela primeira vez.

Como usar expressões “cron” e “rate” para programar execuções de associação

Além das expressões “cron” e “rate” padrão, o State Manager também oferece suporte para expressões “cron” que incluem um dia da semana e o sinal numérico (#) para designar o enésimo dia de um mês para a execução de uma associação. Aqui está um exemplo que executa uma programação do cron na terceira terça-feira de cada mês às 23h30 UTC:

```
cron(30 23 ? * TUE#3 *)
```

Aqui está um exemplo que acontece na segunda quinta-feira de cada mês à meia-noite UTC:

```
cron(0 0 ? * THU#2 *)
```

O State Manager também oferece suporte ao sinal (L) para indicar o último dia X do mês. Aqui está um exemplo que executa uma programação do cron na última terça-feira de cada mês à meia-noite UTC:

```
cron(0 0 ? * 3L *)
```

Para controlar ainda mais quando uma associação é executada, por exemplo, se você quiser executar uma associação dois dias após o patch de terça-feira, você pode especificar um deslocamento. Um deslocamento define quantos dias esperar após o dia programado para executar uma associação. Por exemplo, se você especificou uma programação do cron de `cron(0 0 ?`

`* THU#2 *)`, você pode especificar o número 3 no campo Schedule offset (Deslocamento da programação) para executar a associação todos os domingos após a segunda quinta-feira do mês.

> **Note**
>
> Para usar deslocamentos, é necessário selecionar Aplicar associação somente no próximo intervalo Cron especificado no console ou especificar o parâmetro `ApplyOnlyAtCronInterval` na linha de comando. Quando uma dessas opções está ativada, o State Manager não executa a associação imediatamente após a criação.

Para obter mais informações sobre expressões cron e rate, consulte Referência: Expressões cron e rate para o Systems Manager.

Criar uma associação (console)

O procedimento a seguir descreve como usar o console do Systems Manager para criar uma associação do State Manager.

> **Note**
>
> Observe as seguintes informações:
>
> - Este procedimento descreve como criar uma associação que usa um documento `Command` ou `Policy` para direcionar nós gerenciados. Para obter informações sobre como criar uma associação que usa um runbook de automação para direcionar nós ou outros tipos de recursos da AWS, consulteProgramação de automações com associações do State Manager.
> - Ao criar uma associação, é possível especificar no máximo cinco chaves de tags usando o AWS Management Console. Todas as chaves de tags especificadas para a associação deverão estar atualmente atribuídas ao nó. Se não estiverem, o State Manager falhará em direcionar o nó para a associação.

Como criar uma associação do State Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha State Manager.

3. Selecione Criar associação.
4. No campo Name, especifique um nome.
5. Na lista Document (Documento), escolha a opção ao lado do nome de um documento. Observe o tipo de documento. Este procedimento aplica-se apenas a documentos `Command` e `Policy`. Para obter informações sobre como criar uma associação que usa um documento do runbook de Automação, consulte Programação de automações com associações do State Manager.

> **Important**
>
> O State Manager não é compatível com a execução de associações que usam uma nova versão de um documento se esse documento for compartilhado de outra conta. O State Manager sempre executa a versão `default` de um documento se compartilhada de outra conta, mesmo que o console do Systems Manager mostre que uma nova versão foi processada. Se você quiser executar uma associação usando uma nova versão de um documento compartilhado de outra conta, deverá definir a versão do documento como `default`.

6. Em Parameters (Parâmetros), especifique os parâmetros de entrada obrigatórios.
7. (Opcional) Escolha um alarme do CloudWatch para aplicar à sua associação para monitoramento.

> **Note**
>
> Observe as informações a seguir sobre esta etapa.
>
> - A lista de alarmes exibe um máximo de 100 alarmes. Se você não vir o alarme na lista, use a AWS Command Line Interface para criar a associação. Para obter mais informações, consulte Criar uma associação (linha de comando).
> - Para anexar um alarme do CloudWatch ao seu comando, a entidade principal do IAM que cria a associação deve ter permissão para a ação `iam:createServiceLinkedRole`. Para obter mais informações sobre alarmes do CloudWatch, consulte Usar alarmes do Amazon CloudWatch.
> - Se o alarme for ativado, quaisquer invocações ou automações de comando pendentes não serão executadas.

8. Em Targets (Destinos), escolha uma opção. Para obter informações sobre como usar destinos, consulte Sobre destinos e controles de taxa em associações do State Manager.

> **Note**
>
> Para que as associações criadas com os runbooks do Automation sejam aplicadas quando novos nós de destino são detectados, algumas condições específicas devem ser atendidas. Para mais informações, consulte Sobre as atualizações de destino com runbooks do Automation.

9. Na seção Specify schedule (Especificar programação), escolha On Schedule (Na programação) ou No schedule (Sem programação). Se você escolher On Schedule (Na programação), use os botões fornecidos para criar uma programação cron ou rate para a associação.

   Se você não quiser que a associação seja executada imediatamente após ser criada, selecione Aplicar associação somente no próximo intervalo Cron especificado.

10. (Opcional) No campo Schedule offset (Deslocamento da programação), especifique um número entre 1 e 6.

11. Na seção Advanced options (Opções avançadas), use Compliance severity (Gravidade de conformidade) para escolher um nível de gravidade para a associação e use Change Calendars (Alterar calendários) para escolher um calendário de alteração para a associação.

    Relatórios de conformidade indicam se o estado é compatível ou não, juntamente com o nível de gravidade que você indicar aqui. Para obter mais informações, consulte Sobre a conformidade de associações do State Manager.

    O calendário de alteração determina quando a associação é executada. Se o calendário estiver fechado, a associação não será aplicada. Se o calendário estiver aberto, a associação será executada apropriadamente. Para obter mais informações, consulte AWS Systems Manager Change Calendar.

12. Na seção Rate control (Controle de taxa), escolha opções para controlar como a associação é executada em vários nós. Para obter mais informações sobre como usar controles de taxa, consulte Sobre destinos e controles de taxa em associações do State Manager.

    Na seção Concurrency (Simultaneidade), escolha uma opção:

    - Escolha Targets (Destinos) para inserir um número absoluto de destinos que podem executar a associação simultaneamente.
    - Escolha Percentage (Porcentagem) para inserir uma porcentagem do conjunto de destino que pode executar a associação simultaneamente.

Na seção Error threshold (Limite de erro), escolha uma opção:

- Escolha errors (erros) para inserir um número absoluto de erros permitidos antes de o State Manager parar de executar associações em destinos adicionais.
- Escolha percentage (porcentagem) para inserir uma porcentagem de erros permitidos antes de o State Manager parar de executar associações em destinos adicionais.

13. (Opcional) Em Opções de saída, para salvar a saída de comando em um arquivo, selecione a caixa Habilitar a gravação da saída no S3. Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

> **Note**
>
> As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil da instância atribuído ao nó gerenciado, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço IAM associada ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

A seguir estão as permissões mínimas necessárias para ativar a saída do Amazon S3 para uma associação. É possível restringir ainda mais o acesso ao anexar políticas do IAM a usuários ou perfis em uma conta. No mínimo, um perfil de instância do Amazon EC2 deve ter uma função do IAM com a política gerenciada do `AmazonSSMManagedInstanceCore` e a política inline a seguir.

JSON

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

```
                "s3:PutObject",
                "s3:GetObject",
                "s3:PutObjectAcl"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

Para permissões mínimas, o bucket do Amazon S3 para o qual você exporta deve ter as configurações padrão definidas pelo console do Amazon S3. Para obter mais informações sobre como criar buckets do Amazon S3, consulte Criar um bucket, no Guia do usuário do console do Amazon S3.

> **Note**
>
> As operações de API iniciadas pelo documento SSM durante uma execução de associação não são registradas em log no AWS CloudTrail.

14. Escolha Create Association (Criar associação).

> **Note**
>
> Se você excluir a associação criada, a associação não será mais executada em nenhum destino dessa associação.

Criar uma associação (linha de comando)

O seguinte procedimento descreve como usar a AWS CLI (no Linux ou no Windows) ou o Tools for PowerShell para criar uma associação do State Manager. Esta seção inclui vários exemplos que mostram como usar destinos e controles de taxa. Destinos e controles de taxa permitem atribuir uma associação a dezenas ou centenas de nós enquanto controla a execução dessas associações. Para obter mais informações sobre destinos e controles de taxa, consulte Sobre destinos e controles de taxa em associações do State Manager.

> **Important**
>
> Este procedimento descreve como criar uma associação que usa um documento `Command` ou `Policy` para direcionar nós gerenciados. Para obter informações sobre como criar uma associação que usa um runbook de automação para direcionar nós ou outros tipos de recursos da AWS, consulte Programação de automações com associações do State Manager.

Antes de começar

O parâmetro `targets` é uma matriz de critérios de pesquisa que segmenta nós usando uma combinação `Key`,`Value` especificada por você. Se você planeja criar uma associação em dezenas ou centenas de nós usando o parâmetro `targets`, revise as seguintes opções de direcionamento antes de iniciar o procedimento.

Direcionar nós específicos especificando IDs

```
--targets Key=InstanceIds,Values=instance-id-1,instance-id-2,instance-id-3
```

```
--targets
 Key=InstanceIds,Values=i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE,i-07782c72faEXAMPLE
```

Direcionar instâncias usando tags do

```
--targets Key=tag:tag-key,Values=tag-value-1,tag-value-2,tag-value-3
```

```
--targets Key=tag:Environment,Values=Development,Test,Pre-production
```

Direcionar nós usando AWS Resource Groups

```
--targets Key=resource-groups:Name,Values=resource-group-name
```

```
--targets Key=resource-groups:Name,Values=WindowsInstancesGroup
```

Direcionar todas as instâncias na Conta da AWS e Região da AWS atuais

```
--targets Key=InstanceIds,Values=*
```

> **Note**
>
> Observe as seguintes informações:
>
> - O State Manager não é compatível com a execução de associações que usam uma nova versão de um documento se esse documento for compartilhado de outra conta. O State Manager sempre executa a versão `default` de um documento se compartilhada de outra conta, mesmo que o console do Systems Manager mostre que uma nova versão foi processada. Se você quiser executar uma associação usando uma nova versão de um documento compartilhado de outra conta, você deverá definir a versão do documento como `default`.
> - O State Manager não é compatível com o parâmetro `IncludeChildOrganizationUnits` para TargetLocation.
> - É possível especificar no máximo cinco chaves de etiqueta usando a AWS CLI. Se você usar a AWS CLI, todas as chaves de etiqueta especificadas no comando `create-association` deverão estar atualmente atribuídas ao nó. Se não estiverem, o State Manager falhará em direcionar o nó para uma associação.
> - Ao criar uma associação, você especifica quando a programação é executada. Você deve especificar a programação usando uma expressão cron ou rate. Para obter mais informações sobre expressões cron e rate, consulte Expressões cron e rate para associações.
> - Para que as associações criadas com os runbooks do Automation sejam aplicadas quando novos nós de destino são detectados, algumas condições específicas devem ser atendidas. Para mais informações, consulte Sobre as atualizações de destino com runbooks do Automation.

Como criar uma associação

1. Instale e configure a AWS CLI ou o Ferramentas da AWS para PowerShell, caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI e Instalar o Ferramentas da AWS para PowerShell.
2. Use o seguinte formato para criar um comando que cria uma associação do State Manager. Substitua cada *`espaço reservado para recurso de exemplo`* por suas próprias informações.

Linux & macOS

```
aws ssm create-association \
    --name document_name \
    --document-version version_of_document_applied \
    --instance-id instances_to_apply_association_on \
    --parameters (if any) \
    --targets target_options \
    --schedule-expression "cron_or_rate_expression" \
    --apply-only-at-cron-interval required_parameter_for_schedule_offsets \
    --schedule-offset number_between_1_and_6 \
    --output-location s3_bucket_to_store_output_details \
    --association-name association_name \
    --max-errors a_number_of_errors_or_a_percentage_of_target_set \
    --max-concurrency a_number_of_instances_or_a_percentage_of_target_set \
    --compliance-severity severity_level \
    --calendar-names change_calendar_names \
    --target-locations aws_region_or_account \
    --tags "Key=tag_key,Value=tag_value"
```

Windows

```
aws ssm create-association ^
    --name document_name ^
    --document-version version_of_document_applied ^
    --instance-id instances_to_apply_association_on ^
    --parameters (if any) ^
    --targets target_options ^
    --schedule-expression "cron_or_rate_expression" ^
    --apply-only-at-cron-interval required_parameter_for_schedule_offsets ^
    --schedule-offset number_between_1_and_6 ^
    --output-location s3_bucket_to_store_output_details ^
    --association-name association_name ^
    --max-errors a_number_of_errors_or_a_percentage_of_target_set ^
    --max-concurrency a_number_of_instances_or_a_percentage_of_target_set ^
    --compliance-severity severity_level ^
    --calendar-names change_calendar_names ^
    --target-locations aws_region_or_account ^
    --tags "Key=tag_key,Value=tag_value"
```

PowerShell

```
New-SSMAssociation `
    -Name document_name `
    -DocumentVersion version_of_document_applied `
    -InstanceId instances_to_apply_association_on `
    -Parameters (if any) `
    -Target target_options `
    -ScheduleExpression "cron_or_rate_expression" `
    -ApplyOnlyAtCronInterval required_parameter_for_schedule_offsets `
    -ScheduleOffSet number_between_1_and_6 `
    -OutputLocation s3_bucket_to_store_output_details `
    -AssociationName association_name `
    -MaxError  a_number_of_errors_or_a_percentage_of_target_set
    -MaxConcurrency a_number_of_instances_or_a_percentage_of_target_set `
    -ComplianceSeverity severity_level `
    -CalendarNames change_calendar_names `
    -TargetLocations aws_region_or_account `
    -Tags "Key=tag_key,Value=tag_value"
```

O exemplo a seguir cria uma associação em nós marcados com `"Environment,Linux"`. A associação usa o documento `AWS-UpdateSSMAgent` para atualizar o SSM Agent nos nós marcados todo domingo às 2h UTC. Essa associação é executada simultaneamente em um máximo de 10 nós a qualquer momento. Além disso, essa associação tem a execução interrompida em mais nós para um intervalo de execução específico se a contagem de erros exceder 5. Para relatórios de conformidade, essa associação recebe o nível de gravidade Médio.

Linux & macOS

```
aws ssm create-association \
  --association-name Update_SSM_Agent_Linux \
  --targets Key=tag:Environment,Values=Linux \
  --name AWS-UpdateSSMAgent  \
  --compliance-severity "MEDIUM" \
  --schedule-expression "cron(0 2 ? * SUN *)" \
  --max-errors "5" \
  --max-concurrency "10"
```

Windows

```
aws ssm create-association ^
  --association-name Update_SSM_Agent_Linux ^
  --targets Key=tag:Environment,Values=Linux ^
  --name AWS-UpdateSSMAgent  ^
  --compliance-severity "MEDIUM" ^
  --schedule-expression "cron(0 2 ? * SUN *)" ^
  --max-errors "5" ^
  --max-concurrency "10"
```

PowerShell

```
New-SSMAssociation `
  -AssociationName Update_SSM_Agent_Linux `
  -Name AWS-UpdateSSMAgent `
  -Target @{
      "Key"="tag:Environment"
      "Values"="Linux"
    } `
  -ComplianceSeverity MEDIUM `
  -ScheduleExpression "cron(0 2 ? * SUN *)" `
  -MaxConcurrency 10 `
  -MaxError 5
```

O exemplo a seguir direciona IDs de nós, especificando um valor curinga (*). Isso permite que o Systems Manager crie uma associação em todos os nós na Conta da AWS e Região da AWS atuais. Essa associação é executada simultaneamente em um máximo de 10 nós a qualquer momento. Além disso, essa associação tem a execução interrompida em mais nós para um intervalo de execução específico se a contagem de erros exceder 5. Para relatórios de conformidade, essa associação recebe o nível de gravidade Médio. Essa associação usa um deslocamento de programação, o que significa que ela é executada dois dias após a programação do cron especificada. Ela também inclui o `ApplyOnlyAtCronInterval`, que é necessário para usar o deslocamento da programação, e que indica que a associação não será executada imediatamente após a sua criação.

Linux & macOS

```
aws ssm create-association \
```

```
  --association-name Update_SSM_Agent_Linux \
  --name "AWS-UpdateSSMAgent" \
  --targets "Key=instanceids,Values=*" \
  --compliance-severity "MEDIUM" \
  --schedule-expression "cron(0 2 ? * SUN#2 *)" \
  --apply-only-at-cron-interval \
  --schedule-offset 2 \
  --max-errors "5" \
  --max-concurrency "10" \
```

Windows

```
aws ssm create-association ^
  --association-name Update_SSM_Agent_Linux ^
  --name "AWS-UpdateSSMAgent" ^
  --targets "Key=instanceids,Values=*" ^
  --compliance-severity "MEDIUM" ^
  --schedule-expression "cron(0 2 ? * SUN#2 *)" ^
  --apply-only-at-cron-interval ^
  --schedule-offset 2 ^
  --max-errors "5" ^
  --max-concurrency "10" ^
  --apply-only-at-cron-interval
```

PowerShell

```
New-SSMAssociation `
  -AssociationName Update_SSM_Agent_All `
  -Name AWS-UpdateSSMAgent `
  -Target @{
      "Key"="InstanceIds"
      "Values"="*"
    } `
  -ScheduleExpression "cron(0 2 ? * SUN#2 *)" `
  -ApplyOnlyAtCronInterval `
  -ScheduleOffset 2 `
  -MaxConcurrency 10 `
  -MaxError 5 `
  -ComplianceSeverity MEDIUM `
  -ApplyOnlyAtCronInterval
```

O exemplo a seguir cria uma associação em nós em grupos de recurso. O grupo é chamado de “Departamento de RH”. A associação usa o documento `AWS-UpdateSSMAgent` para atualizar o SSM Agent nos nós marcados todo domingo às 2h UTC. Essa associação é executada simultaneamente em um máximo de 10 nós a qualquer momento. Além disso, essa associação tem a execução interrompida em mais nós para um intervalo de execução específico se a contagem de erros exceder 5. Para relatórios de conformidade, essa associação recebe o nível de gravidade Médio. Essa associação será executada na programação de cron especificada. Ela não será executada imediatamente após criação da associação.

Linux & macOS

```
aws ssm create-association \
  --association-name Update_SSM_Agent_Linux \
  --targets Key=resource-groups:Name,Values=HR-Department \
  --name AWS-UpdateSSMAgent  \
  --compliance-severity "MEDIUM" \
  --schedule-expression "cron(0 2 ? * SUN *)" \
  --max-errors "5" \
  --max-concurrency "10" \
  --apply-only-at-cron-interval
```

Windows

```
aws ssm create-association ^
  --association-name Update_SSM_Agent_Linux ^
  --targets Key=resource-groups:Name,Values=HR-Department ^
  --name AWS-UpdateSSMAgent  ^
  --compliance-severity "MEDIUM" ^
  --schedule-expression "cron(0 2 ? * SUN *)" ^
  --max-errors "5" ^
  --max-concurrency "10" ^
  --apply-only-at-cron-interval
```

PowerShell

```
New-SSMAssociation `
  -AssociationName Update_SSM_Agent_Linux `
  -Name AWS-UpdateSSMAgent `
  -Target @{
```

```
      "Key"="resource-groups:Name"
      "Values"="HR-Department"
    } `
  -ScheduleExpression "cron(0 2 ? * SUN *)" `
  -MaxConcurrency 10 `
  -MaxError 5 `
  -ComplianceSeverity MEDIUM `
  -ApplyOnlyAtCronInterval
```

O exemplo a seguir cria uma associação que é executada em nós marcados com um ID de nó específico. A associação usa o documento SSM Agent para atualizar o SSM Agent nos nós de destino uma vez quando o calendário de alteração está aberto. A associação verifica o estado do calendário quando é executado. Se o calendário estiver fechado no momento do lançamento e a associação for executada apenas uma vez, ela não será executada novamente porque a janela de execução da associação já haverá passado. Se o calendário estiver aberto, a associação será executada apropriadamente.

> **Note**
>
> Se você adicionar novos nós às tags ou grupos de recursos nos quais uma associação atua quando o calendário de alteração for fechado, a associação será aplicada a esses nós assim que o calendário de alteração for aberto.

Linux & macOS

```
aws ssm create-association \
  --association-name CalendarAssociation \
  --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f" \
  --name AWS-UpdateSSMAgent  \
  --calendar-names "arn:aws:ssm:us-east-1:123456789012:document/testCalendar1" \
  --schedule-expression "rate(1day)"
```

Windows

```
aws ssm create-association ^
  --association-name CalendarAssociation ^
  --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f" ^
  --name AWS-UpdateSSMAgent  ^
```

```
  --calendar-names "arn:aws:ssm:us-east-1:123456789012:document/testCalendar1" ^
  --schedule-expression "rate(1day)"
```

PowerShell

```
New-SSMAssociation `
  -AssociationName CalendarAssociation `
  -Target @{
      "Key"="tag:instanceids"
      "Values"="i-0cb2b964d3e14fd9f"
    } `
  -Name AWS-UpdateSSMAgent `
  -CalendarNames "arn:aws:ssm:us-east-1:123456789012:document/testCalendar1" `
  -ScheduleExpression "rate(1day)"
```

O exemplo a seguir cria uma associação que é executada em nós marcados com um ID de nó específico. A associação usa o documento SSM Agent para atualizar o SSM Agent nos nós marcados todo domingo às 2h UTC. Essa associação será executada somente na programação do cron especificada, quando o calendário de alterações for aberto. Quando a associação é criada, ela verifica o estado do calendário. Se o calendário estiver fechado, a associação não será aplicada. Quando o intervalo para aplicar a associação começa às 2:00 da manhã de domingo, a associação verifica se o calendário está aberto. Se o calendário estiver aberto, a associação será executada apropriadamente.

> **Note**
>
> Se você adicionar novos nós às tags ou grupos de recursos nos quais uma associação atua quando o calendário de alteração for fechado, a associação será aplicada a esses nós assim que o calendário de alteração for aberto.

Linux & macOS

```
aws ssm create-association \
  --association-name MultiCalendarAssociation \
  --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f" \
  --name AWS-UpdateSSMAgent  \
  --calendar-names "arn:aws:ssm:us-east-1:123456789012:document/testCalendar1"
 "arn:aws:ssm:us-east-2:123456789012:document/testCalendar2" \
```

```
  --schedule-expression "cron(0 2 ? * SUN *)"
```

Windows

```
aws ssm create-association ^
  --association-name MultiCalendarAssociation ^
  --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f" ^
  --name AWS-UpdateSSMAgent  ^
  --calendar-names "arn:aws:ssm:us-east-1:123456789012:document/testCalendar1"
 "arn:aws:ssm:us-east-2:123456789012:document/testCalendar2" ^
  --schedule-expression "cron(0 2 ? * SUN *)"
```

PowerShell

```
New-SSMAssociation `
  -AssociationName MultiCalendarAssociation `
  -Name AWS-UpdateSSMAgent `
  -Target @{
      "Key"="tag:instanceids"
      "Values"="i-0cb2b964d3e14fd9f"
    } `
  -CalendarNames "arn:aws:ssm:us-east-1:123456789012:document/testCalendar1"
 "arn:aws:ssm:us-east-2:123456789012:document/testCalendar2" `
  -ScheduleExpression "cron(0 2 ? * SUN *)"
```

Note

Se você excluir a associação criada, a associação não será mais executada em nenhum destino dessa associação. Além disso, se você especificou o parâmetro `apply-only-at-cron-interval`, poderá redefinir essa opção. Para fazer isso, especifique o parâmetro `no-apply-only-at-cron-interval` ao atualizar a associação pela linha de comando. Este parâmetro força a execução da associação imediatamente após a atualização da associação e de acordo com o intervalo especificado.

Edita e criar uma nova versão de uma associação

Você pode editar uma associação do State Manager para especificar um novo nome, agendamento, nível de gravidade, destinos ou outros valores. Para associações baseadas em documentos do

tipo SSM Command, você pode também optar por gravar a saída do comando em um bucket do Amazon Simple Storage Service (Amazon S3). Depois de editar uma associação, o State Manager cria uma nova versão. Você pode visualizar diferentes versões após a edição, conforme descrito nos seguintes procedimentos.

> **Note**
>
> Para que as associações criadas com os runbooks do Automation sejam aplicadas quando novos nós de destino são detectados, algumas condições específicas devem ser atendidas. Para mais informações, consulte Sobre as atualizações de destino com runbooks do Automation.

Os procedimentos a seguir descrevem como editar e criar uma nova versão de uma associação usando o console do Systems Manager, a AWS Command Line Interface (AWS CLI) e o Ferramentas da AWS para PowerShell (Tools for PowerShell).

> **Important**
>
> O State Manager não é compatível com a execução de associações que usam uma nova versão de um documento se esse documento for compartilhado de outra conta. O State Manager sempre executa a versão `default` de um documento se ele for compartilhado de outra conta, mesmo que o console do Systems Manager mostre que uma nova versão foi processada. Se você quiser executar uma associação usando uma nova versão de um documento compartilhado de outra conta, você deverá definir a versão do documento como `default`.

Editar uma associação (console)

O seguinte procedimento descreve como usar o console do Systems Manager para editar e criar uma nova versão de uma associação.

> **Note**
>
> Para associações que usam documentos do SSM Command, não runbooks do Automation, esse procedimento exige que você tenha acesso de gravação a um bucket do Amazon S3 existente. Se você nunca usou o Amazon S3 antes, não será cobrado por alterações pelo

uso do Amazon S3. Para obter informações sobre como criar um bucket, consulte Criar um bucket.

Para editar uma associação do State Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha State Manager.
3. Escolha uma associação existente e, em seguida, Edit.
4. Reconfigure a associação para atender aos seus requisitos atuais.

   Para obter informações sobre opções de associação com documentos `Command` e `Policy`, consulte Criar associações. Para obter informações sobre opções de associação com runbooks do Automation, consulte Programação de automações com associações do State Manager.
5. Escolha Salvar alterações.
6. (Opcional) Para ver informações de associação, na página Associations, escolha o nome da associação que você editou e depois escolha Versions. O sistema lista cada versão da associação que você criou e editou.
7. (Opcional) Para ver a saída de associações com base em documentos do SSM `Command`, faça o seguinte:

   a. Abra o console do Amazon S3, em https://console.aws.amazon.com/s3/.

   b. Selecione o nome do bucket do Amazon S3 que você especificou para armazenamento da saída de comando e, em seguida, escolha a pasta denominada com o ID do nó que executou a associação. (Se tiver optado por armazenar a saída em uma pasta no bucket, abra-a primeiro.)

   c. Desça vários níveis na hierarquia da pasta `awsrunPowerShell`, até o arquivo `stdout`.

   d. Escolha Open (Abrir) ou Download (Fazer download) para visualizar o nome do host.

Editar uma associação (linha de comando)

O seguinte procedimento descreve como usar a AWS CLI (no Linux ou no Windows) ou o Ferramentas da AWS para PowerShell para editar e criar uma nova versão de uma associação.

Para editar uma associação do State Manager

1. Instale e configure a AWS CLI ou o Ferramentas da AWS para PowerShell, caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI e Instalar o Ferramentas da AWS para PowerShell.

2. Use o seguinte formato para criar um comando para editar e criar uma nova versão de uma associação existente do State Manager. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   > **Important**
   >
   > Quando você chama o `update-association`, o sistema descarta todos os parâmetros opcionais da solicitação e substitui a associação com valores nulos para esses parâmetros. Isso faz parte do design. Você deve especificar todos os parâmetros opcionais na chamada, mesmo se você não estiver alterando os parâmetros. Isso inclui o parâmetro `--name`. Antes de chamar essa ação, recomendamos que você chame a operação de API `describe-association` e anote todos os parâmetros opcionais necessários para a chamada `update-association`.

   Linux & macOS

   ```
   aws ssm update-association \
       --name document_name \
       --document-version version_of_document_applied \
       --instance-id instances_to_apply_association_on \
       --parameters (if any) \
       --targets target_options \
       --schedule-expression "cron_or_rate_expression" \
       --schedule-offset "number_between_1_and_6" \
       --output-location s3_bucket_to_store_output_details \
       --association-name association_name \
       --max-errors a_number_of_errors_or_a_percentage_of_target_set \
       --max-concurrency a_number_of_instances_or_a_percentage_of_target_set \
       --compliance-severity severity_level \
       --calendar-names change_calendar_names \
       --target-locations aws_region_or_account
   ```

Windows

```
aws ssm update-association ^
    --name document_name ^
    --document-version version_of_document_applied ^
    --instance-id instances_to_apply_association_on ^
    --parameters (if any) ^
    --targets target_options ^
    --schedule-expression "cron_or_rate_expression" ^
    --schedule-offset "number_between_1_and_6" ^
    --output-location s3_bucket_to_store_output_details ^
    --association-name association_name ^
    --max-errors a_number_of_errors_or_a_percentage_of_target_set ^
    --max-concurrency a_number_of_instances_or_a_percentage_of_target_set ^
    --compliance-severity severity_level ^
    --calendar-names change_calendar_names ^
    --target-locations aws_region_or_account
```

PowerShell

```
Update-SSMAssociation `
    -Name document_name `
    -DocumentVersion version_of_document_applied `
    -InstanceId instances_to_apply_association_on `
    -Parameters (if any) `
    -Target target_options `
    -ScheduleExpression "cron_or_rate_expression" `
    -ScheduleOffset "number_between_1_and_6" `
    -OutputLocation s3_bucket_to_store_output_details `
    -AssociationName association_name `
    -MaxError  a_number_of_errors_or_a_percentage_of_target_set
    -MaxConcurrency a_number_of_instances_or_a_percentage_of_target_set `
    -ComplianceSeverity severity_level `
    -CalendarNames change_calendar_names `
    -TargetLocations aws_region_or_account
```

O seguinte exemplo atualiza uma associação existente para alterar o nome para `TestHostnameAssociation2`. A nova versão de associação é executada a cada hora e grava a saída de comandos no bucket do Amazon S3 especificado.

Linux & macOS

```
aws ssm update-association \
  --association-id 8dfe3659-4309-493a-8755-01234EXAMPLE \
  --association-name TestHostnameAssociation2 \
  --parameters commands="echo Association" \
  --output-location S3Location='{OutputS3Region=us-
east-1,OutputS3BucketName=amzn-s3-demo-bucket,OutputS3KeyPrefix=logs}' \
  --schedule-expression "cron(0 */1 * * ? *)"
```

Windows

```
aws ssm update-association ^
  --association-id 8dfe3659-4309-493a-8755-01234EXAMPLE ^
  --association-name TestHostnameAssociation2 ^
  --parameters commands="echo Association" ^
  --output-location S3Location='{OutputS3Region=us-
east-1,OutputS3BucketName=amzn-s3-demo-bucket,OutputS3KeyPrefix=logs}' ^
  --schedule-expression "cron(0 */1 * * ? *)"
```

PowerShell

```
Update-SSMAssociation `
  -AssociationId b85ccafe-9f02-4812-9b81-01234EXAMPLE `
  -AssociationName TestHostnameAssociation2 `
  -Parameter @{"commands"="echo Association"} `
  -S3Location_OutputS3BucketName amzn-s3-demo-bucket `
  -S3Location_OutputS3KeyPrefix logs `
  -S3Location_OutputS3Region us-east-1 `
  -ScheduleExpression "cron(0 */1 * * ? *)"
```

O seguinte exemplo atualiza uma associação existente para alterar o nome para `CalendarAssociation`. A nova associação é executada quando o calendário está aberto e grava a saída do comando no bucket do Amazon S3 especificado.

Linux & macOS

```
aws ssm update-association \
  --association-id 8dfe3659-4309-493a-8755-01234EXAMPLE \
```

```
  --association-name CalendarAssociation \
  --parameters commands="echo Association" \
  --output-location S3Location='{OutputS3Region=us-
east-1,OutputS3BucketName=amzn-s3-demo-bucket,OutputS3KeyPrefix=logs}' \
  --calendar-names "arn:aws:ssm:us-east-1:123456789012:document/testCalendar2"
```

Windows

```
aws ssm update-association ^
  --association-id 8dfe3659-4309-493a-8755-01234EXAMPLE ^
  --association-name CalendarAssociation ^
  --parameters commands="echo Association" ^
  --output-location S3Location='{OutputS3Region=us-
east-1,OutputS3BucketName=amzn-s3-demo-bucket,OutputS3KeyPrefix=logs}' ^
  --calendar-names "arn:aws:ssm:us-east-1:123456789012:document/testCalendar2"
```

PowerShell

```
Update-SSMAssociation `
  -AssociationId b85ccafe-9f02-4812-9b81-01234EXAMPLE `
  -AssociationName CalendarAssociation `
  -AssociationName OneTimeAssociation `
  -Parameter @{"commands"="echo Association"} `
  -S3Location_OutputS3BucketName amzn-s3-demo-bucket `
  -CalendarNames "arn:aws:ssm:us-east-1:123456789012:document/testCalendar2"
```

O seguinte exemplo atualiza uma associação existente para alterar o nome para `MultiCalendarAssociation`. A nova associação é executada quando os calendários estão abertos e grava a saída do comando no bucket do Amazon S3 especificado.

Linux & macOS

```
aws ssm update-association \
  --association-id 8dfe3659-4309-493a-8755-01234EXAMPLE \
  --association-name MultiCalendarAssociation \
  --parameters commands="echo Association" \
  --output-location S3Location='{OutputS3Region=us-
east-1,OutputS3BucketName=amzn-s3-demo-bucket,OutputS3KeyPrefix=logs}' \
  --calendar-names "arn:aws:ssm:us-east-1:123456789012:document/testCalendar1"
 "arn:aws:ssm:us-east-2:123456789012:document/testCalendar2"
```

Windows

```
aws ssm update-association ^
  --association-id 8dfe3659-4309-493a-8755-01234EXAMPLE ^
  --association-name MultiCalendarAssociation ^
  --parameters commands="echo Association" ^
  --output-location S3Location='{OutputS3Region=us-
east-1,OutputS3BucketName=amzn-s3-demo-bucket,OutputS3KeyPrefix=logs}' ^
  --calendar-names "arn:aws:ssm:us-east-1:123456789012:document/testCalendar1"
 "arn:aws:ssm:us-east-2:123456789012:document/testCalendar2"
```

PowerShell

```
Update-SSMAssociation `
  -AssociationId b85ccafe-9f02-4812-9b81-01234EXAMPLE `
  -AssociationName MultiCalendarAssociation `
  -Parameter @{"commands"="echo Association"} `
  -S3Location_OutputS3BucketName amzn-s3-demo-bucket `
  -CalendarNames "arn:aws:ssm:us-east-1:123456789012:document/testCalendar1"
 "arn:aws:ssm:us-east-2:123456789012:document/testCalendar2"
```

3. Para visualizar a nova versão da associação, execute o seguinte comando.

Linux & macOS

```
aws ssm describe-association \
  --association-id b85ccafe-9f02-4812-9b81-01234EXAMPLE
```

Windows

```
aws ssm describe-association ^
  --association-id b85ccafe-9f02-4812-9b81-01234EXAMPLE
```

PowerShell

```
Get-SSMAssociation `
  -AssociationId b85ccafe-9f02-4812-9b81-01234EXAMPLE | Select-Object *
```

O sistema retorna informações como estas.

Linux & macOS

```
{
    "AssociationDescription": {
        "ScheduleExpression": "cron(0 */1 * * ? *)",
        "OutputLocation": {
            "S3Location": {
                "OutputS3KeyPrefix": "logs",
                "OutputS3BucketName": "amzn-s3-demo-bucket",
                "OutputS3Region": "us-east-1"
            }
        },
        "Name": "AWS-RunPowerShellScript",
        "Parameters": {
            "commands": [
                "echo Association"
            ]
        },
        "LastExecutionDate": 1559316400.338,
        "Overview": {
            "Status": "Success",
            "DetailedStatus": "Success",
            "AssociationStatusAggregatedCount": {}
        },
        "AssociationId": "b85ccafe-9f02-4812-9b81-01234EXAMPLE",
        "DocumentVersion": "$DEFAULT",
        "LastSuccessfulExecutionDate": 1559316400.338,
        "LastUpdateAssociationDate": 1559316389.753,
        "Date": 1559314038.532,
        "AssociationVersion": "2",
        "AssociationName": "TestHostnameAssociation2",
        "Targets": [
            {
                "Values": [
                    "Windows"
                ],
                "Key": "tag:Environment"
            }
        ]
    }
}
```

Windows


```
{
    "AssociationDescription": {
        "ScheduleExpression": "cron(0 */1 * * ? *)",
        "OutputLocation": {
            "S3Location": {
                "OutputS3KeyPrefix": "logs",
                "OutputS3BucketName": "amzn-s3-demo-bucket",
                "OutputS3Region": "us-east-1"
            }
        },
        "Name": "AWS-RunPowerShellScript",
        "Parameters": {
            "commands": [
                "echo Association"
            ]
        },
        "LastExecutionDate": 1559316400.338,
        "Overview": {
            "Status": "Success",
            "DetailedStatus": "Success",
            "AssociationStatusAggregatedCount": {}
        },
        "AssociationId": "b85ccafe-9f02-4812-9b81-01234EXAMPLE",
        "DocumentVersion": "$DEFAULT",
        "LastSuccessfulExecutionDate": 1559316400.338,
        "LastUpdateAssociationDate": 1559316389.753,
        "Date": 1559314038.532,
        "AssociationVersion": "2",
        "AssociationName": "TestHostnameAssociation2",
        "Targets": [
            {
                "Values": [
                    "Windows"
                ],
                "Key": "tag:Environment"
            }
        ]
    }
}
```

PowerShell

```
AssociationId                : b85ccafe-9f02-4812-9b81-01234EXAMPLE
AssociationName              : TestHostnameAssociation2
AssociationVersion           : 2
AutomationTargetParameterName :
ComplianceSeverity           :
Date                         : 5/31/2019 2:47:18 PM
DocumentVersion              : $DEFAULT
InstanceId                   :
LastExecutionDate            : 5/31/2019 3:26:40 PM
LastSuccessfulExecutionDate  : 5/31/2019 3:26:40 PM
LastUpdateAssociationDate    : 5/31/2019 3:26:29 PM
MaxConcurrency               :
MaxErrors                    :
Name                         : AWS-RunPowerShellScript
OutputLocation               :
 Amazon.SimpleSystemsManagement.Model.InstanceAssociationOutputLocation
Overview                     :
 Amazon.SimpleSystemsManagement.Model.AssociationOverview
Parameters                   : {[commands,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
ScheduleExpression           : cron(0 */1 * * ? *)
Status                       :
Targets                      : {tag:Environment}
```

Excluir associações

Use o procedimento a seguir para excluir uma associação usando o console do AWS Systems Manager.

Para excluir uma associação

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha State Manager.
3. Selecione uma associação e, em seguida, selecione Excluir.

Você pode excluir várias associações em uma única operação executando uma automação no console do AWS Systems Manager. Quando você seleciona várias associações para exclusão, o

State Manager inicia a página inicial do runbook de automação com os IDs de associação inseridos como valores de parâmetros de entrada.

Para excluir várias associações em uma única operação

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha State Manager.
3. Selecione cada associação que deseja excluir e, em seguida, escolha Excluir.
4. (Opcional) Na área Parâmetros de entrada adicionais, selecione o nome do recurso da Amazon (ARN) para o recurso de assumir perfil que você deseja que a automação use durante a execução. Para criar um novo recurso de assumir perfil, escolha Criar.
5. Selecione Enviar.

Executar grupos do Auto Scaling com associações

A prática recomendada ao usar associações para executar grupos do Auto Scaling é usar destinos de etiquetas. Não usar tags pode fazer com que você atinja o limite de associação.

Se todos os nós estiverem marcados com a mesma chave e valor, você precisará apenas de uma associação para executar seu grupo do Auto Scaling. O seguinte procedimento descreve como criar essa associação.

Para criar uma associação que executa grupos do Auto Scaling

1. Verifique se todos os nós no grupo do Auto Scaling estão marcados com a mesma chave e valor. Para obter mais instruções sobre como marcar nós, consulte Marcar grupos e instâncias do Auto Scaling no Manual do usuário do AWS Auto Scaling.
2. Crie uma associação usando o procedimento em Para obter informações, consulte Trabalhar com associações no Systems Manager..

   Se estiver trabalhando no console, escolha Specify instance tags (Especificar etiquetas de instância) no campo Targets (Destinos). Para Instance tags (Tags da instância), insira a chave e o valor da Tag para o grupo de Auto Scaling.

   Se você estiver usando a AWS Command Line Interface (AWS CLI), especifique `--targets Key=tag:tag-key,Values=tag-value` onde a chave e o valor corresponderem às tags que você aplicou em seus nós.

Visualizar históricos de associação

Você pode visualizar todas as execuções de determinado ID de associação usando a operação de API DescribeAssociationExecutions. Use esta operação para ver o status, o status detalhado, os resultados, o runtime mais recente e outras informações sobre uma associação do State Manager. O State Manager é uma ferramenta do AWS Systems Manager. Essa operação de API também inclui filtros para ajudá-lo a localizar as associações de acordo com os critérios que você especifica. Por exemplo, você pode especificar uma data e hora e usar o filtro GREATER_THAN para visualizar as execuções processadas após a data e a hora especificadas.

Se, por exemplo, uma execução de associação falhou, você pode analisar os detalhes de determinada execução usando a operação de API DescribeAssociationExecutionTargets. Essa operação mostra os recursos, como IDs de nós, nos quais a associação executou e os vários status de associação. Assim você pode ver qual recurso ou nó não conseguiu executar uma associação. Com o ID do recurso, você pode visualizar os detalhes da execução do comando para ver qual etapa falhou em um comando.

Os exemplos desta seção também incluem informações sobre como usar a operação de API StartAssociationsOnce para executar uma associação uma vez no momento da criação. Você pode usar essa opração de API quando investigar execuções de associação com falha. Se você vir que uma associação falhou, poderá fazer uma alteração no recurso e, em seguida, executar imediatamente a associação para ver se a alteração no recurso faz com que a associação seja executada com êxito.

> **Note**
>
> As operações de API iniciadas pelo documento SSM durante uma execução de associação não são registradas em log no AWS CloudTrail.

Visualizar históricos de associação (console)

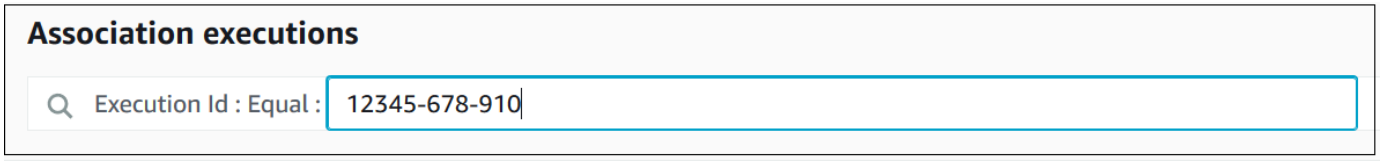
Use o seguinte procedimento para visualizar o histórico de execução de determinado ID de associação e, em seguida, visualizar os detalhes da execução de um ou mais recursos.

Para visualizar o histórico de execução de determinado ID de associação

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. Selecione State Manager.

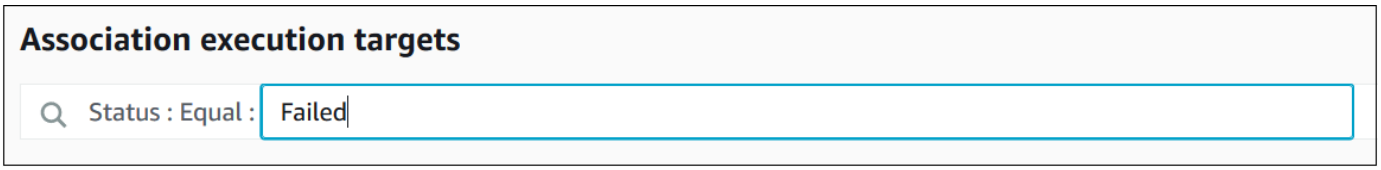
3. No campo Association id (ID da associação), escolha uma associação cujo histórico você deseja visualizar.
4. Escolha o botão View details (Visualizar detalhes).
5. Escolha a guia Execution history (Histórico de execução).
6. Escolha uma associação da qual você deseja visualizar os detalhes da execução no nível de recurso. Por exemplo, escolha uma associação que mostra um status de Failed (Com falha). Em seguida, você pode visualizar os detalhes de execução dos nós que não conseguiram executar a associação.

   Use os filtros da caixa de pesquisa para localizar a execução cujos detalhes você deseja visualizar.

7. Escolha um ID de execução. A página Association execution targets (Destinos de execução da associação) é aberta. Essa página mostra todos os recursos que executaram a associação.
8. Escolha um ID de recurso para visualizar as informações específicas sobre esse recurso.

   Use os filtros da caixa de pesquisa para localizar o recurso cujos detalhes você deseja visualizar.

9. Se você está investigando uma associação que não foi executada, pode usar o botão Apply association now (Aplicar associação agora) para executar uma associação uma vez no momento da criação. Depois que você fizer alterações no recurso que não conseguiu executar a associação, escolha o link Association ID (ID da associação) na navegação estruturada.
10. Escolha o botão Apply association now (Aplicar associação agora). Após a execução ser concluída, verifique se a execução da associação foi bem-sucedida.

Visualizar históricos de associação (linha de comando)

O procedimento a seguir descreve como usar a AWS Command Line Interface (AWS CLI) (no Linux ou no Windows) ou o Ferramentas da AWS para PowerShell para visualizar o histórico de execução

de um ID de associação específico. Depois disso, o procedimento descreve como visualizar detalhes de execução de um ou mais recursos.

Para visualizar o histórico de execução de determinado ID de associação

1. Instale e configure a AWS CLI ou o Ferramentas da AWS para PowerShell, caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI e Instalar o Ferramentas da AWS para PowerShell.

2. Execute o comando a seguir para visualizar uma lista de execuções para determinado ID de associação.

   Linux & macOS

   ```
   aws ssm describe-association-executions \
     --association-id ID \
     --filters Key=CreatedTime,Value="2018-04-10T19:15:38.372Z",Type=GREATER_THAN
   ```

   > **Note**
   >
   > Esse comando inclui um filtro para limitar os resultados somente para as execuções que ocorreram após uma data e hora específicas. Para visualizar todas as execuções de um determinado ID de associação, remova o parâmetro `--filters` e o valor `Key=CreatedTime,Value="2018-04-10T19:15:38.372Z",Type=GREATER_THAN`.

   Windows

   ```
   aws ssm describe-association-executions ^
     --association-id ID ^
     --filters Key=CreatedTime,Value="2018-04-10T19:15:38.372Z",Type=GREATER_THAN
   ```

   > **Note**
   >
   > Esse comando inclui um filtro para limitar os resultados somente para as execuções que ocorreram após uma data e hora específicas. Para visualizar todas as execuções

de um determinado ID de associação, remova o parâmetro `--filters` e o valor `Key=CreatedTime,Value="2018-04-10T19:15:38.372Z",Type=GREATER_THAN`.

PowerShell

```
Get-SSMAssociationExecution `
  -AssociationId ID `
  -Filter
 @{"Key"="CreatedTime";"Value"="2019-06-01T19:15:38.372Z";"Type"="GREATER_THAN"}
```

> **Note**
>
> Esse comando inclui um filtro para limitar os resultados somente para as execuções que ocorreram após uma data e hora específicas. Para visualizar todas as execuções de um determinado ID de associação, remova o parâmetro `-Filter` e o valor `@{"Key"="CreatedTime";"Value"="2019-06-01T19:15:38.372Z";"Type"="GREAT`

O sistema retorna informações como estas.

Linux & macOS

```
{
   "AssociationExecutions":[
      {
         "Status":"Success",
         "DetailedStatus":"Success",
         "AssociationId":"c336d2ab-09de-44ba-8f6a-6136cEXAMPLE",
         "ExecutionId":"76a5a04f-caf6-490c-b448-92c02EXAMPLE",
         "CreatedTime":1523986028.219,
         "AssociationVersion":"1"
      },
      {
         "Status":"Success",
         "DetailedStatus":"Success",
         "AssociationId":"c336d2ab-09de-44ba-8f6a-6136cEXAMPLE",
         "ExecutionId":"791b72e0-f0da-4021-8b35-f95dfEXAMPLE",
         "CreatedTime":1523984226.074,
         "AssociationVersion":"1"
```

```
      },
      {
         "Status":"Success",
         "DetailedStatus":"Success",
         "AssociationId":"c336d2ab-09de-44ba-8f6a-6136cEXAMPLE",
         "ExecutionId":"ecec60fa-6bb0-4d26-98c7-140308EXAMPLE",
         "CreatedTime":1523982404.013,
         "AssociationVersion":"1"
      }
   ]
}
```

Windows

```
{
   "AssociationExecutions":[
      {
         "Status":"Success",
         "DetailedStatus":"Success",
         "AssociationId":"c336d2ab-09de-44ba-8f6a-6136cEXAMPLE",
         "ExecutionId":"76a5a04f-caf6-490c-b448-92c02EXAMPLE",
         "CreatedTime":1523986028.219,
         "AssociationVersion":"1"
      },
      {
         "Status":"Success",
         "DetailedStatus":"Success",
         "AssociationId":"c336d2ab-09de-44ba-8f6a-6136cEXAMPLE",
         "ExecutionId":"791b72e0-f0da-4021-8b35-f95dfEXAMPLE",
         "CreatedTime":1523984226.074,
         "AssociationVersion":"1"
      },
      {
         "Status":"Success",
         "DetailedStatus":"Success",
         "AssociationId":"c336d2ab-09de-44ba-8f6a-6136cEXAMPLE",
         "ExecutionId":"ecec60fa-6bb0-4d26-98c7-140308EXAMPLE",
         "CreatedTime":1523982404.013,
         "AssociationVersion":"1"
      }
   ]
}
```

PowerShell

```
AssociationId         : c336d2ab-09de-44ba-8f6a-6136cEXAMPLE
AssociationVersion    : 1
CreatedTime           : 8/18/2019 2:00:50 AM
DetailedStatus        : Success
ExecutionId           : 76a5a04f-caf6-490c-b448-92c02EXAMPLE
LastExecutionDate     : 1/1/0001 12:00:00 AM
ResourceCountByStatus : {Success=1}
Status                : Success

AssociationId         : c336d2ab-09de-44ba-8f6a-6136cEXAMPLE
AssociationVersion    : 1
CreatedTime           : 8/11/2019 2:00:54 AM
DetailedStatus        : Success
ExecutionId           : 791b72e0-f0da-4021-8b35-f95dfEXAMPLE
LastExecutionDate     : 1/1/0001 12:00:00 AM
ResourceCountByStatus : {Success=1}
Status                : Success

AssociationId         : c336d2ab-09de-44ba-8f6a-6136cEXAMPLE
AssociationVersion    : 1
CreatedTime           : 8/4/2019 2:01:00 AM
DetailedStatus        : Success
ExecutionId           : ecec60fa-6bb0-4d26-98c7-140308EXAMPLE
LastExecutionDate     : 1/1/0001 12:00:00 AM
ResourceCountByStatus : {Success=1}
Status                : Success
```

Você pode limitar os resultados usando um ou mais filtros. O exemplo a seguir retorna todas as associações que foram executadas antes de uma data e hora específicas.

Linux & macOS

```
aws ssm describe-association-executions \
  --association-id ID \
  --filters Key=CreatedTime,Value="2018-04-10T19:15:38.372Z",Type=LESS_THAN
```

Windows

```
aws ssm describe-association-executions ^
  --association-id ID ^
  --filters Key=CreatedTime,Value="2018-04-10T19:15:38.372Z",Type=LESS_THAN
```

PowerShell

```
Get-SSMAssociationExecution `
  -AssociationId 14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE `
  -Filter
 @{"Key"="CreatedTime";"Value"="2019-06-01T19:15:38.372Z";"Type"="LESS_THAN"}
```

O exemplo a seguir retorna todas as associações que foram executadas com êxito após uma data e hora específicas.

Linux & macOS

```
aws ssm describe-association-executions \
  --association-id ID \
  --filters Key=CreatedTime,Value="2018-04-10T19:15:38.372Z",Type=GREATER_THAN
 Key=Status,Value=Success,Type=EQUAL
```

Windows

```
aws ssm describe-association-executions ^
  --association-id ID ^
  --filters Key=CreatedTime,Value="2018-04-10T19:15:38.372Z",Type=GREATER_THAN
 Key=Status,Value=Success,Type=EQUAL
```

PowerShell

```
Get-SSMAssociationExecution `
  -AssociationId 14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE `
  -Filter @{
      "Key"="CreatedTime";
      "Value"="2019-06-01T19:15:38.372Z";
      "Type"="GREATER_THAN"
    },
```

```
    @{
      "Key"="Status";
      "Value"="Success";
      "Type"="EQUAL"
    }
```

3. Execute o seguinte comando para visualizar todos os destinos em que determinada execução foi realizada.

   Linux & macOS

   ```
   aws ssm describe-association-execution-targets \
     --association-id ID \
     --execution-id ID
   ```

   Windows

   ```
   aws ssm describe-association-execution-targets ^
     --association-id ID ^
     --execution-id ID
   ```

   PowerShell

   ```
   Get-SSMAssociationExecutionTarget `
     -AssociationId 14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE `
     -ExecutionId 76a5a04f-caf6-490c-b448-92c02EXAMPLE
   ```

   Você pode limitar os resultados usando um ou mais filtros. O exemplo a seguir retorna informações sobre todos os destinos em que a associação não foi executada.

   Linux & macOS

   ```
   aws ssm describe-association-execution-targets \
     --association-id ID \
     --execution-id ID \
     --filters Key=Status,Value="Failed"
   ```

   Windows

   ```
   aws ssm describe-association-execution-targets ^
   ```

```
  --association-id ID ^
  --execution-id ID ^
  --filters Key=Status,Value="Failed"
```

PowerShell

```
Get-SSMAssociationExecutionTarget `
  -AssociationId 14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE `
  -ExecutionId 76a5a04f-caf6-490c-b448-92c02EXAMPLE `
  -Filter @{
      "Key"="Status";
      "Value"="Failed"
    }
```

O exemplo a seguir retorna informações sobre determinado nó gerenciado em que uma associação não foi executada.

Linux & macOS

```
aws ssm describe-association-execution-targets \
  --association-id ID \
  --execution-id ID \
  --filters Key=Status,Value=Failed Key=ResourceId,Value="i-02573cafcfEXAMPLE"
 Key=ResourceType,Value=ManagedInstance
```

Windows

```
aws ssm describe-association-execution-targets ^
  --association-id ID ^
  --execution-id ID ^
  --filters Key=Status,Value=Failed Key=ResourceId,Value="i-02573cafcfEXAMPLE"
 Key=ResourceType,Value=ManagedInstance
```

PowerShell

```
Get-SSMAssociationExecutionTarget `
  -AssociationId 14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE `
  -ExecutionId 76a5a04f-caf6-490c-b448-92c02EXAMPLE `
  -Filter @{
      "Key"="Status";
```

```
      "Value"="Success"
    },
    @{
      "Key"="ResourceId";
      "Value"="i-02573cafcfEXAMPLE"
    },
    @{
      "Key"="ResourceType";
      "Value"="ManagedInstance"
    }
```

4. Se você está investigando uma associação que não foi executada, pode usar a operação de API StartAssociationsOnce para executar uma associação imediatamente, e apenas uma vez. Depois de fazer alterações no recurso no qual a associação falhou, execute o seguinte comando para executar a associação imediatamente e apenas uma vez.

   Linux & macOS

   ```
   aws ssm start-associations-once \
     --association-id ID
   ```

   Windows

   ```
   aws ssm start-associations-once ^
     --association-id ID
   ```

   PowerShell

   ```
   Start-SSMAssociationsOnce `
     -AssociationId ID
   ```

Trabalhar com associações usando o IAM

O State Manager, uma ferramenta do AWS Systems Manager, usa destinos para escolher com quais instâncias você configura suas associações. Originalmente, as associações foram criadas especificando um nome de documento (`Name`) e ID da instância (`InstanceId`). Isso criou uma associação entre um documento e uma instância ou nó gerenciado. As associações eram identificadas por esses parâmetros. Esses parâmetros agora estão obsoletos, mas ainda são

suportados. Os recursos `instance` e `managed-instance` foram adicionados como recursos para ações com `Name` e `InstanceId`.

AWS Identity and Access Management O comportamento de imposição de políticas (IAM) depende do tipo de recurso especificado. Recursos para operações do State Manager são aplicadas somente com base na solicitação passada. O State Manager não executa uma verificação profunda das propriedades dos recursos na sua conta. Uma solicitação só é validada em relação a recursos de política se o parâmetro de solicitação contiver os recursos de política especificados. Por exemplo, se você especificar uma instância no bloco de recursos, a política será imposta se a solicitação usar o parâmetro `InstanceId`. O parâmetro `Targets` para cada recurso na conta não é verificado para esse `InstanceId`.

A seguir estão alguns casos com comportamento confuso:

- DescribeAssociation, DeleteAssociation e UpdateAssociation usam os recursos `instance`, `managed-instance` e `document` para especificar a forma defasada de se referir a associações. Isso inclui todas as associações criadas com o parâmetro `InstanceId` defasado.
- CreateAssociation, CreateAssociationBatch e UpdateAssociation usam recursos da `instance` e das `managed-instance` para especificar a maneira defasada de se referir a associações. Isso inclui todas as associações criadas com o parâmetro `InstanceId` defasado. O tipo de recurso `document` é parte da maneira obsoleta de se referir a associações e é uma propriedade real de uma associação. Isso significa que você pode construir políticas do IAM com permissões `Allow` ou `Deny` para as ações `Create` e `Update` com base no nome do documento.

Para obter mais informações sobre como usar políticas do IAM com o Systems Manager, consulte Gerenciamento de identidade e acesso para o AWS Systems Manager ou Ações, recursos e chaves de condição para o AWS Systems Manager na Referência de autorização do serviço.

Criar associações que executam arquivos MOF

Você pode executar arquivos Managed Object Format (MOF) para impor um estado desejado nos nós gerenciados do Windows Server com o State Manager, uma ferramenta do AWS Systems Manager, usando o documento do SSM `AWS-ApplyDSCMofs`. O documento `AWS-ApplyDSCMofs` tem dois modos de execução: Com o primeiro modo, você pode configurar a associação para verificar e relatar se os nós gerenciados estão no estado desejado definido nos arquivos MOF especificados. No segundo modo, você pode executar os arquivos MOF e alterar a configuração de seus nós com base nos recursos e seus valores definidos nos arquivos MOF. O documento `AWS-ApplyDSCMofs` permite que você faça download e execute arquivos de configuração MOF do

Amazon Simple Storage Service (Amazon S3), um compartilhamento local ou um site seguro com um domínio HTTPS.

O State Manager registra e relata o status de cada execução de arquivo MOF durante cada execução de associação. O State Manager também informa a saída de cada execução de arquivo MOF como um evento de conformidade que você pode visualizar na página Conformidade do AWS Systems Manager.

A execução do arquivo MOF é criada na configuração de estado desejado do Windows PowerShell (PowerShell DSC). O PowerShell DSC é uma plataforma declarativa usada para configuração, implantação e gerenciamento de sistemas Windows. O PowerShell DSC permite que os administradores descrevam, em documentos de texto simples chamados configurações de DSC, como desejam que um servidor seja configurado. A configuração do PowerShell DSC é um script do PowerShell especializado que informa o que fazer, mas não como fazer. Executar a configuração produz um arquivo MOF. O arquivo MOF pode ser aplicado a um ou mais servidores para atingir a configuração desejada para esses servidores. Os recursos do PowerShell DSC fazem o trabalho real de aplicação da configuração. Para obter mais informações, consulte Visão geral de configuração de estado desejado do Windows PowerShell.

Tópicos

- Usar o Amazon S3 para armazenar artefatos
- Resolver credenciais em arquivos MOF
- Usar tokens em arquivos MOF
- Pré-requisitos para criar associações que executam arquivos MOF
- Criar uma associação que executa arquivos MOF
- Solução de problemas na criação de associações que executam arquivos MOF
- Visualizar detalhes de conformidade de recurso do DSC

Usar o Amazon S3 para armazenar artefatos

Se você estiver usando o Amazon S3 para armazenar módulos do PowerShell, arquivos MOF, relatórios de conformidade ou relatórios de status, a função do AWS Identity and Access Management (IAM) usada pelo SSM Agent do AWS Systems Manager deverá ter permissões `GetObject` e `ListBucket` no bucket. Se você não fornecer essas permissões, o sistema retornará um erro de acesso negado. Abaixo estão informações importantes sobre como armazenar artefatos no Amazon S3.

- Se o bucket estiver em uma conta da Conta da AWS diferente, você deverá criar uma política de recurso de bucket que conceda as permissões `GetObject` e `ListBucket` para a conta (ou a função IAM).
- Se você quiser usar recursos do DSC personalizados, poderá fazer download desses recursos de um bucket do Amazon S3. Você pode também instalá-los automaticamente a partir da galeria do PowerShell.
- Se você estiver usando o Amazon S3 como uma origem de módulo, precisará fazer upload do módulo como um arquivo Zip com distinção entre letras maiúsculas e minúsculas no seguinte formato: *ModuleName_ModuleVersion*.zip. Por exemplo: MyModule_1.0.0.zip.
- Todos os arquivos devem estar na raiz do bucket. As estruturas da pasta não são compatíveis.

Resolver credenciais em arquivos MOF

As credenciais são resolvidas usando o AWS Secrets Manager ou AWS Systems Manager Parameter Store. Isso permite que você configure a rotação de credencial automática. Isso também permite que o DSC propague automaticamente credenciais para seus servidores sem reimplantar os MOFs.

Para usar uma chave secreta AWS Secrets Manager em uma configuração, crie um objeto PSCredential onde o nome de usuário é SecretId ou SecretARN do segredo que contém a credencial. Você pode especificar qualquer valor para a senha. O valor é ignorado. Veja um exemplo a seguir.

```
Configuration MyConfig
{
   $ss = ConvertTo-SecureString -String 'a_string' -AsPlaintext -Force
   $credential = New-Object PSCredential('a_secret_or_ARN', $ss)

    Node localhost
    {
       File file_name
       {
           DestinationPath = 'C:\MyFile.txt'
           SourcePath = '\\FileServer\Share\MyFile.txt'
           Credential = $credential
       }
    }
}
```

Compile o MOF usando a configuração PsAllowPlaintextPassword em dados de configuração. Isso é aceitável, pois a credencial contém apenas um rótulo.

No Secrets Manager, certifique-se de que o nó tenha acesso GetSecretValue em uma política gerenciada do IAM e, opcionalmente, na política de recurso secreta se houver. Para trabalhar com o DSC, o segredo deverá estar no seguinte formato.

```
{ 'Username': 'a_name', 'Password': 'a_password' }
```

O segredo pode ter outras propriedades (por exemplo, propriedades utilizadas para rotação), mas deve ter pelo menos as propriedades de nome de usuário e senha.

É recomendável que você use um método de rotação de vários usuários, onde há dois nomes de usuário e senhas diferentes, e a função de rotação AWS Lambda alterna entre eles. Esse método permite que você tenha várias contas ativas e, ao mesmo tempo, elimine o risco de bloqueio de um usuário durante a rotação.

Usar tokens em arquivos MOF

Os tokens permitem modificar valores de propriedade de recurso após o MOF ser compilado. Isso permite que você reutilize arquivos MOF comuns em vários servidores que exigem configurações muito semelhantes.

A substituição de token funciona apenas para as propriedades de recurso do tipo `String`. No entanto, se o recurso tiver uma propriedade de nó CIM aninhada, ele também resolverá tokens de propriedades `String` nesse nó CIM. Não é possível usar substituição de token para números ou matrizes.

Por exemplo, considere um cenário em que você está usando o recurso e xComputerManagement e deseja renomear o computador usando DSC. Normalmente, você precisaria de um arquivo MOF dedicado para essa máquina. No entanto, com o suporte de token, você pode criar um único arquivo MOF e aplicá-lo a todos os seus nós. Na propriedade `ComputerName`, em vez de codificar o nome do computador no MOF, você pode usar um token do tipo tag de instância. O valor é resolvido durante a análise de MOF. Veja o exemplo a seguir.

```
Configuration MyConfig
{
    xComputer Computer
    {
        ComputerName = '{tag:ComputerName}'
    }
```

```
}
```

Em seguida, defina uma tag no nó gerenciado no console do Systems Manager ou uma tag do Amazon Elastic Compute Cloud (Amazon EC2) no console do Amazon EC2. Quando você executa o documento, o script substitui o token {tag:ComputerName} para o valor da tag da instância.

Você também pode combinar várias tags em uma única propriedade, por exemplo:

```
Configuration MyConfig
{
    File MyFile
    {
        DestinationPath = '{env:TMP}\{tag:ComputerName}'
        Type = 'Directory'
    }
}
```

Há cinco tipos diferentes de tokens que você pode usar:

- tag: Amazon EC2 ou tags de nós gerenciados.
- tagb64: é igual à tag, mas o sistema usa base64 para decodificar o valor. Isso permite que você use caracteres especiais em valores de tag.
- env: resolve variáveis de ambiente.
- ssm: valores do Parameter Store. Somente os tipos string e string segura têm suporte.
- tagssm: é igual à tag, mas se a tag não estiver definida no nó, o sistema tentará resolver o valor de um parâmetro do Systems Manager com o mesmo nome. Isso é útil em situações em que você deseja um "valor global padrão", mas quer substituí-lo em um único nó (por exemplo, implantações de uma caixa).

Veja um exemplo de Parameter Store que usa o tipo de token `ssm`.

```
File MyFile
{
    DestinationPath = "C:\ProgramData\ConnectionData.txt"
    Content = "{ssm:%servicePath%/ConnectionData}"
}
```

Os tokens desempenham um papel importante para reduzir o código redundante tornando arquivos MOF genéricos e reutilizáveis. Se você pode evitar o arquivo MOF específico do servidor, não há

necessidade de um serviço de criação de MOF. Um serviço de criação de MOF aumenta os custos, reduz o tempo de provisionamento e aumenta o risco de oscilações de configuração entre nós agrupados devido a diferentes versões do módulo a ser instalado no servidor de compilação quando os MOFs foram compilados.

Pré-requisitos para criar associações que executam arquivos MOF

Antes de criar uma associação que executa arquivos MOF, verifique se os nós gerenciados têm os seguintes pré-requisitos instalados:

- Windows PowerShell versão 5.0 ou posterior. Para obter mais informações, consulte Requisitos do sistema Windows PowerShell no site Microsoft.com.
- AWS Tools for Windows PowerShell versão 3.3.261.0 ou posterior.
- SSM Agent versão 2.2 ou posterior.

Criar uma associação que executa arquivos MOF

Para criar uma associação que executa arquivos MOF

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha State Manager.
3. Escolha State Manager e, em seguida, Create Association (Criar associação).
4. No campo Name, especifique um nome. Isso é opcional, mas recomendado. Um nome pode ajudar você a compreender a finalidade da associação quando a criou. Espaços não são permitidos no nome.
5. Na lista Document (Documento), escolha **`AWS-ApplyDSCMofs`**.
6. Na seção Parameters, especifique suas opções para os parâmetros de entrada necessários e opcionais.

    a. Mofs To Apply (MOFs para aplicar): especifique um ou mais arquivos MOF a serem executados quando essa associação é executada. Use vírgulas para separar uma lista de arquivos MOF. Você pode especificar as seguintes opções para localizar o arquivo MOF.

    - Um nome de bucket do Amazon S3. Os nomes de bucket devem usar letras minúsculas. Especifique essas informações usando o formato a seguir.

    ```
    s3:amzn-s3-demo-bucket:MOF_file_name.mof
    ```

Se você quiser especificar uma Região da AWS, use o formato a seguir.

```
s3:bucket_Region:amzn-s3-demo-bucket:MOF_file_name.mof
```

- Um site seguro. Especifique essas informações usando o formato a seguir.

```
https://domain_name/MOF_file_name.mof
```

Aqui está um exemplo.

```
https://www.example.com/TestMOF.mof
```

- Um sistema de arquivos em um compartilhamento local. Especifique essas informações usando o formato a seguir.

```
\server_name\shared_folder_name\MOF_file_name.mof
```

Aqui está um exemplo.

```
\StateManagerAssociationsBox\MOFs_folder\MyMof.mof
```

b. Service Path (Caminho do serviço): (opcional) um caminho de serviço é um prefixo do bucket do Amazon S3 no qual você deseja gravar relatórios e informações de status. Ou, um caminho de serviço é um caminho para tags com base no parâmetro Parameter Store. Ao resolver tags com base em parâmetros, o sistema usa {ssm:%servicePath %/*parameter_name*} para injetar o valor servicePath no nome do parâmetro. Por exemplo, se o caminho de serviço é "WebServers/Produção", os sistemas resolve o parâmetro como: WebServers/Production/*parameter_name*. Isso é útil para quando você estiver executando vários ambientes na mesma conta.

c. Report Bucket Name (Nome do bucket de relatórios): (opcional) insira o nome de um bucket do Amazon S3 em que você deseja gravar dados de conformidade. Os relatórios são salvos nesse bucket no formato JSON.

> **Note**
>
> Você pode prefixar o nome do bucket com uma região em que o bucket está localizado. Veja um exemplo: us-west-2:MyMOFBucket. Se você estiver usando um proxy para endpoints do Amazon S3 em uma região específica que não inclua

us-east-1, prefixe o nome do bucket com uma região. Se o nome do bucket não for prefixado, ele detectará automaticamente a região do bucket usando o endpoint us-east-1.

d. Mof Operation Mode (Modo de operação do Mof): escolha o comportamento State Manager ao executar a associação **`AWS-ApplyDSCMofs`**:

- Apply (Aplicar): corrija as configurações de nó que não são compatíveis.
- ReportOnly (Somente relatar): não corrija configurações de nó, mas registre todos os dados de conformidade e relate nós que não são compatíveis.

e. Status Bucket Name (Nome do bucket de status): (opcional) insira o nome de um bucket do Amazon S3 em que você deseja gravar informações de status de execução do MOF. Esses relatórios de status são resumos da execução de conformidade mais recente de um nó. Isso significa que o relatório será substituído na próxima vez que a associação executar arquivos MOF.

> **Note**
>
> Você pode prefixar o nome do bucket com uma região em que o bucket está localizado. Veja um exemplo: `us-west-2:amzn-s3-demo-bucket` Se você estiver usando um proxy para endpoints do Amazon S3 em uma região específica que não inclua us-east-1, prefixe o nome do bucket com uma região. Se o nome do bucket não for prefixado, ele detectará automaticamente a região do bucket usando o endpoint us-east-1.

f. Module Source Bucket Name (Nome do bucket de origem do módulo): (opcional) insira o nome de um bucket do Amazon S3 que contém arquivos de módulo do PowerShell. Se você especificar None, deverá escolher True para a próxima opção, Allow PS Gallery Module Source.

> **Note**
>
> Você pode prefixar o nome do bucket com uma região em que o bucket está localizado. Veja um exemplo: `us-west-2:amzn-s3-demo-bucket` Se você estiver usando um proxy para endpoints do Amazon S3 em uma região específica que não inclua us-east-1, prefixe o nome do bucket com uma região. Se o nome do

bucket não for prefixado, ele detectará automaticamente a região do bucket usando o endpoint us-east-1.

g. Allow PS Gallery Module Source: (opcional) escolha True para fazer download de módulos do PowerShell em https://www.powershellgallery.com/. Se você escolher False, deverá especificar uma origem para a opção anterior, ModuleSourceBucketName.

h. Proxy Uri: (opcional) use essa opção para fazer download de arquivos MOF a partir de um servidor de proxy.

i. Reboot Behavior: (opcional) especifique um dos seguintes comportamentos de reinicialização se a execução do arquivo MOF precisar de reinicialização:

- AfterMof (Após MOF): reinicializa o nó depois que todas as execuções de MOF são concluídas. Mesmo se várias execuções de MOF solicitarem reinicialização, o sistema aguardará até que todas as execuções de MOF sejam concluídas para reinicializar.
- Immediately (Imediatamente): reinicia o nó sempre que uma execução de MOF solicita. Se estiver executando vários arquivos MOF que solicitam reinicialização, o nó será reinicializado várias vezes.
- Never (Nunca): os nós não serão reinicializados, mesmo se a execução de MOF explicitamente solicitar uma reinicialização.

j. Use Computer Name For Reporting: (opcional) ative essa opção para usar o nome do computador para notificar as informações de conformidade. O valor padrão é false (falso), o que significa que o sistema usa o ID do nó para notificar as informações de conformidade.

k. Enable Verbose Logging: (opcional) recomendamos que você ative o log detalhado ao implantar arquivos MOF pela primeira vez.

> **Important**
>
> Quando habilitado, o registro em log detalhado grava mais dados no bucket do Amazon S3 do que o registro de execução da associação padrão. Isso pode resultar em uma performance mais lenta e possivelmente em cobranças de armazenamento mais altas para o Amazon S3. Para reduzir problemas de tamanho do armazenamento, recomendamos ativar as políticas de ciclo de vida no bucket do Amazon S3. Para obter mais informações, consulte Como criar uma política de ciclo de vida para um bucket do S3? no Guia do usuário do console do Amazon Simple Storage Service.

l. Enable Debug Logging: (opcional) recomendamos que você ative o registro de depuração se precisar solucionar falhas de MOF. Recomendamos também que você desative essa opção para uso normal.

> **Important**
>
> Quando ativado, o registro de depuração grava mais dados no bucket do Amazon S3 do que o registro de execução da associação padrão. Isso pode resultar em uma performance mais lenta e possivelmente em cobranças de armazenamento mais altas para o Amazon S3. Para reduzir problemas de tamanho do armazenamento, recomendamos ativar as políticas de ciclo de vida no bucket do Amazon S3. Para obter mais informações, consulte Como criar uma política de ciclo de vida para um bucket do S3? no Guia do usuário do console do Amazon Simple Storage Service.

m. Compliance Type: (opcional) especifique o tipo de conformidade a ser usado para relatar informações de conformidade. O tipo de conformidade padrão é Custom:DSC. Se você criar várias associações que executam arquivos MOF, certifique-se de especificar um tipo de conformidade diferente para cada associação. Caso contrário, cada associação adicional que use Custom:DSC substituirá os dados de conformidade existentes.

n. Pre Reboot Script (Script pré-reinicialização): (opcional) especifique um script para executar se a configuração indicou que uma reinicialização é necessária. O script é executado antes da reinicialização. O script deve ser uma única linha. Separe linhas adicionais usando ponto-e-vírgula.

7. Na seção Targets (Destinos), escolha Specifying tags (Especificação de tags) ou Manually Selecting Instance (Seleção manual da instância). Se você optar por definir o destino dos recursos usando tags, insira uma chave de tag e um valor da tag nos campos fornecidos. Para obter mais informações sobre como usar destinos, consulte Sobre destinos e controles de taxa em associações do State Manager.

8. Na seção Specify schedule (Especificar programação), escolha On Schedule (Na programação) ou No schedule (Sem programação). Se você escolher On Schedule (Na programação), use os botões fornecidos para criar uma programação cron ou rate para a associação.

9. Na seção Advanced options:

   - Em Compliance severity, escolha um nível de gravidade para a associação. Relatórios de conformidade indicam se o estado é compatível ou não, juntamente com o nível de gravidade

que você indicar aqui. Para obter mais informações, consulte Sobre a conformidade de associações do State Manager.

10. Na seção Rate control (Controle de taxa), configure opções para executar associações do State Manager na frota de nós gerenciados. Para obter mais informações sobre essas opções, consulte Sobre destinos e controles de taxa em associações do State Manager.

    Na seção Concurrency (Simultaneidade), escolha uma opção:

    - Escolha Targets (Destinos) para inserir um número absoluto de destinos que podem executar a associação simultaneamente.
    - Escolha Percentage (Porcentagem) para inserir uma porcentagem do conjunto de destino que pode executar a associação simultaneamente.

    Na seção Error threshold (Limite de erro), escolha uma opção:

    - Escolha Errors (Erros) para inserir um número absoluto de erros permitidos antes de o State Manager parar de executar associações em destinos adicionais.
    - Escolha Percentage (Porcentagem) para inserir uma porcentagem de erros permitidos antes de o State Manager parar de executar associações em destinos adicionais.

11. (Opcional) Em Opções de saída, para salvar a saída de comando em um arquivo, selecione a caixa Habilitar a gravação da saída no S3. Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

    > **Note**
    >
    > As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil da instância atribuído ao nó gerenciado, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço IAM associada ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

12. Escolha Create Association (Criar associação).

O State Manager cria e executa imediatamente a associação nos nós ou destinos especificados. Após a execução inicial, a associação é executada em intervalos de acordo com o agendamento que você definiu e de acordo com as seguintes regras:

- O State Manager executa associações em nós que estiverem online quando o intervalo for iniciado e ignora os nós offline.
- O State Manager tenta executar a associação em todos os nós configurados durante um intervalo.
- Se uma associação não for executada durante um intervalo (porque, por exemplo, um valor de simultaneidade limita o número de nós que podem processar a associação de uma só vez), o State Manager tentará executar a associação durante o próximo intervalo.
- O State Manager registra o histórico de todos os intervalos ignorados. Você pode visualizar o histórico na guia Execution History.

> **Note**
>
> O `AWS-ApplyDSCMofs` é um documento do Command do Systems Manager. Isso significa que você também pode executar esse documento usando o Run Command, uma ferramenta do AWS Systems Manager. Para obter mais informações, consulte AWS Systems Manager Run Command.

Solução de problemas na criação de associações que executam arquivos MOF

Esta seção inclui informações para ajudá-lo a solucionar problemas ao criar associações que executam arquivos MOF.

Ativar registro em log aprimorado

Como primeiro passo para a solução de problemas, ative o registro aprimorado. Mais especificamente, faça o seguinte:

1. Verifique se a associação está configurada para gravar a saída do comando no Amazon S3 ou no Amazon CloudWatch Logs (CloudWatch).
2. Defina o parâmetro Enable Verbose Logging como True.
3. Defina o parâmetro Enable Debug Logging como True.

Com registro detalhado e de depuração ativado, o arquivo de saída Stdout inclui detalhes sobre a execução do script. Esse arquivo de saída pode ajudar a identificar onde o script falhou. O arquivo de saída Stderr contém erros que ocorreram durante a execução do script.

Problemas comuns ao criar associações que executam arquivos MOF

Esta seção inclui informações sobre problemas comuns que podem ocorrer ao criar associações que executam arquivos MOF e etapas para solucionar esses problemas.

Meu MOF não foi aplicado

Se State Manager não conseguiu aplicar a associação aos seus nós, comece examinando o arquivo de saída Stderr. Esse arquivo pode ajudá-lo a compreender a causa raiz do problema. Além disso, verifique o seguinte:

- O nó tem as permissões de acesso necessárias para todos os buckets do Amazon S3 relacionados ao MOF. Especificamente:
  - Permissões s3:GetObject: necessárias para arquivos MOF em buckets do Amazon S3 privados, bem como em módulos personalizados em buckets do Amazon S3.
  - Permissão s3:PutObject: necessária para gravar relatórios de conformidade e status de conformidade em buckets do Amazon S3.
- Se você estiver usando tags, certifique-se de que o nó tenha a política do IAM necessária. Usar tags exige que a função do IAM da instância tenha uma política que permita as ações as `ec2:DescribeInstances` e `ssm:ListTagsForResource`.
- Verifique se o nó tem as tags esperadas ou os parâmetros atribuídos.
- Verifique se as tags ou os parâmetros do SSM estão escritos corretamente.
- Tente aplicar o MOF localmente no nó para garantir que não haja um problema com o arquivo MOF em si.

Meu MOF parecia falhar, mas a execução do Systems Manager foi bem-sucedida

Se o documento `AWS-ApplyDSCMofs` foi executado com êxito, o status da execução do Systems Manager será Success (Êxito). Esse status não reflete o status de conformidade de seu nó com os requisitos de configuração no arquivo MOF. Para visualizar o status de conformidade de seus nós, visualize os relatórios de conformidade. Você pode visualizar um relatório JSON no bucket de relatórios do Amazon S3. Isso se aplica às execuções Run Command e State Manager. Além disso, para o State Manager, você pode visualizar detalhes de conformidade na página Compliance (Conformidade) do Systems Manager.

Estados de Stderr: falha de resolução de nome ao tentar entrar em contato com o serviço

Esse erro indica que o script não pode atingir um serviço remoto. O mais provável é que o script não consiga alcançar o Amazon S3. Esse problema geralmente ocorre quando o script tenta gravar relatórios de conformidade ou status de conformidade no bucket do Amazon S3 fornecido nos parâmetros do documento. Normalmente, esse erro ocorre quando um ambiente de computação usa um firewall ou proxy transparente que inclui uma lista de permissões. Para resolver esse problema:

- Use a sintaxe do bucket específico da região para todos os parâmetros do bucket do Amazon S3. Por exemplo, o parâmetro Mofs to Apply deve ser formatado da seguinte forma:

  s3:*bucket-region*:*amzn-s3-demo-bucket*:*mof-file-name*.mof.

  Exemplo: `s3:us-west-2:amzn-s3-demo-bucket:my-mof.mof`

  Os nomes de bucket Report, Status e Module Source devem ser formatados da seguinte forma:

  bucket-region:*amzn-s3-demo-bucket*. Aqui está um exemplo: `us-west-1:amzn-s3-demo-bucket;`
- Se a sintaxe específica da região não corrigir o problema, verifique se os nós de destino podem acessar o Amazon S3 na região desejada. Para verificar isso:
  1. Localize o nome do endpoint do Amazon S3 na região do Amazon S3 apropriada. Para obter informações, consulte Amazon S3 Service Endpoints no Referência geral da Amazon Web Services.
  2. Faça login no nó de destino e execute o comando ping a seguir:

     ```
     ping s3.s3-region.amazonaws.com
     ```

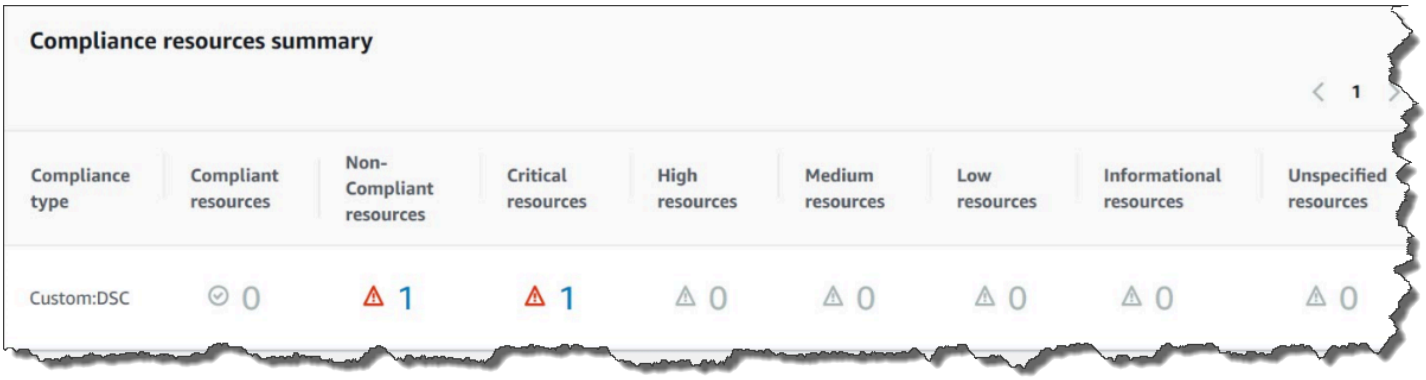
     Uma falha de ping significa que o Amazon S3 está inativo ou um firewall/proxy transparente está bloqueando o acesso à região do Amazon S3 ou o nó não pode acessar a Internet.

Visualizar detalhes de conformidade de recurso do DSC

O Systems Manager captura informações de conformidade sobre falhas de recursos do DSC no bucket de status do Amazon S3 que você especificou ao executar o documento `AWS-ApplyDSCMofs`. Pesquisar informações sobre falhas de recursos do DSC em um bucket do Amazon S3 pode ser demorado. Em vez disso, você pode exibir essas informações na página Compliance (Conformidade) do Systems Manager.

A seção Compliance resources summary (Resumo dos recursos de conformidade) exibe uma contagem do recursos com falha. No exemplo a seguir, o ComplianceType é Custom:DSC e um recurso não está em conformidade.

> **Note**
>
> Custom:DSC é o valor padrão de ComplianceType no documento `AWS-ApplyDSCMofs`. Esse valor é personalizável.

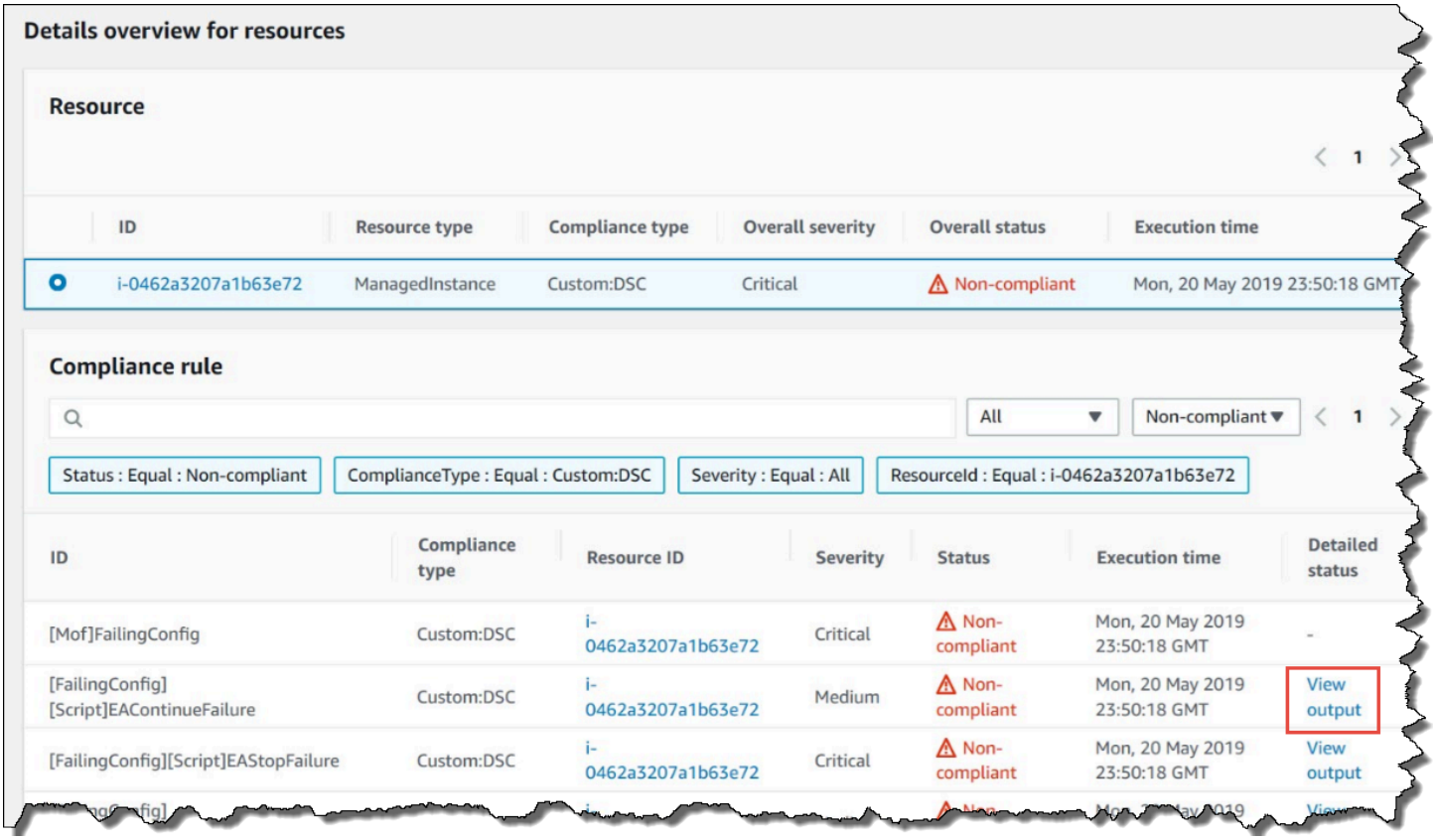


Compliance resources summary

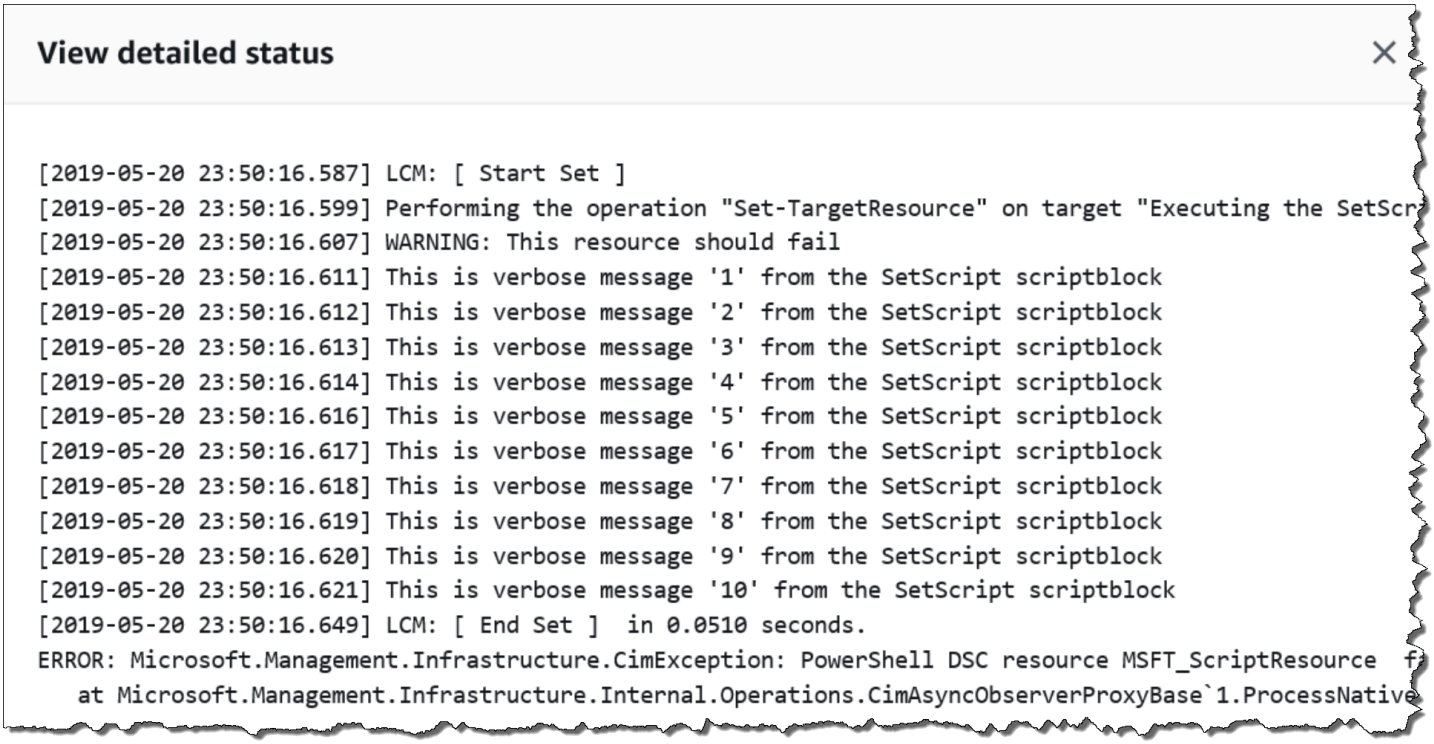
1

Compliance type	Compliant resources	Non-Compliant resources	Critical resources	High resources	Medium resources	Low resources	Informational resources	Unspecified resources
Custom:DSC	0	1	1	0	0	0	0	0

A seção Details overview for resources (Visão geral dos detalhes dos recursos) exibe informações sobre o recurso da AWS que não está em conformidade com o recurso do DSC. Esta seção também inclui o nome do MOF, as etapas de execução do script e (quando aplicável) um link View output (Visualizar saída) para visualizar informações detalhadas do status.



O link View output (Visualizar saída) mostra os últimos 4.000 caracteres do status detalhado. O Systems Manager começa com a exceção como o primeiro elemento e verifica novamente as mensagens detalhadas e adiciona todos os itens que conseguir até atingir a cota de 4.000 caracteres. Esse processo exibe as mensagens de log que foram registradas antes da exceção ser gerada, que são as mensagens mais relevantes para a solução de problemas.



Para obter informações sobre como visualizar informações de conformidade, consulte AWS Systems Manager Compliance.

Situações que afetam o relatório de conformidade

Se houver falha na associação do State Manager, os dados de conformidade não serão relatados. Mais especificamente, se houver falha no processamento de um MOF, o Systems Manager não relatará nenhum item de conformidade porque haverá falha nas associações. Por exemplo, se o Systems Manager tentar baixar um MOF de um bucket do Amazon S3 que o nó não tem permissão para acessar, haverá falha na associação e os dados sem conformidade serão relatados.

Se houver falha em um segundo recurso do MOF, o Systems Manager relatará os dados de conformidade. Por exemplo, se um MOF tentar criar um arquivo em uma unidade que não existe, o Systems Manager relatará os dados de conformidade porque o documento `AWS-ApplyDSCMofs` poderá ser processado completamente, o que significa que a associação foi executada com êxito.

Criar associações que executam manuais do Ansible

Você pode criar associações do State Manager que executam manuais do Ansible usando o documento do SSM `AWS-ApplyAnsiblePlaybooks`. O State Manager é uma ferramenta do AWS Systems Manager. Esse documento oferece os seguintes benefícios para executar manuais:

- Suporte à execução de manuais complexos
- Support ao download de playbooks do GitHub e do Amazon Simple Storage Service (Amazon S3)
- Suporte à estrutura de manual compactada
- Registro em log aprimorado
- Capacidade de especificar qual manual executar quando os manuais estiverem empacotados

> **Note**
>
> O Systems Manager inclui dois documentos do SSM que permitem criar associações do State Manager que executam manuais do Ansible: `AWS-RunAnsiblePlaybook` e `AWS-ApplyAnsiblePlaybooks`. O documento `AWS-RunAnsiblePlaybook` está obsoleto. Ele permanece disponível no Systems Manager para fins legados. Recomendamos usar o documento `AWS-ApplyAnsiblePlaybooks` devido aos aprimoramentos descritos aqui. As associações que executam manuais do Ansible não são compatíveis com o macOS.

Suporte à execução de manuais complexos

O documento `AWS-ApplyAnsiblePlaybooks` oferece suporte a manuais complexos e empacotados porque copia toda a estrutura de arquivos para um diretório local antes de executar o manual principal especificado. Você pode fornecer manuais de origem em arquivos Zip ou em uma estrutura de diretórios. O arquivo Zip ou diretório pode ser armazenado no GitHub ou no Amazon S3.

Suporte ao download de manuais do GitHub

O documento `AWS-ApplyAnsiblePlaybooks` usa o plugin `aws:downloadContent` para fazer download dos arquivos de manual. Os arquivos podem ser armazenados no GitHub em um único arquivo ou como um conjunto combinado de arquivos de manual. Para fazer download de conteúdo do GitHub, é necessário especificar informações sobre seu repositório do GitHub no formato JSON. Aqui está um exemplo.

```
{
   "owner":"TestUser",
   "repository":"GitHubTest",
   "path":"scripts/python/test-script",
   "getOptions":"branch:master",
   "tokenInfo":"{{ssm-secure:secure-string-token}}"
}
```

Suporte ao download de manuais do Amazon S3

Também é possível armazenar e baixar manuais do Ansible no Amazon S3 como um único arquivo .zip ou uma estrutura de diretórios. Para fazer download de conteúdo do Amazon S3, é necessário especificar o caminho para o arquivo. Veja dois exemplos a seguir.

Exemplo 1: fazer download de um arquivo de manual específico

```
{
   "path":"https://s3.amazonaws.com/amzn-s3-demo-bucket/playbook.yml"
}
```

Exemplo 2: fazer download do conteúdo de um diretório

```
{
   "path":"https://s3.amazonaws.com/amzn-s3-demo-bucket/ansible/webservers/"
}
```

> **Important**
>
> Se você especificar o Amazon S3, o perfil de instância do AWS Identity and Access Management (IAM) nos seus nós gerenciados deverá incluir permissões para o bucket do S3. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager.

Suporte à estrutura de manual compactada

O documento `AWS-ApplyAnsiblePlaybooks` permite executar arquivos .zip compactados no pacote obtido por download. O documento verifica se os arquivos obtidos por download contêm um arquivo compactado no formato .zip. Se um .zip for encontrado, o documento descompactará automaticamente o arquivo e executará a automação do Ansible especificada.

Registro em log aprimorado

O documento `AWS-ApplyAnsiblePlaybooks` inclui um parâmetro opcional para especificar diferentes níveis de registro em log. Especifique -v para nível de detalhe baixo, -vv ou -vvv para nível de detalhe médio e -vvvv para registro em log no nível de depuração. Essas opções são mapeadas diretamente nas opções de nível de detalhe do Ansible.

Capacidade de especificar qual manual executar quando os manuais estiverem empacotados

O documento `AWS-ApplyAnsiblePlaybooks` inclui um parâmetro necessário para especificar qual manual executar quando vários manuais estiverem empacotados. Essa opção fornece flexibilidade para executar manuais a fim de oferecer suporte a diferentes casos de uso.

Noções básicas sobre as dependências instaladas

Se você especificar True (Verdadeiro) para o parâmetro InstallDependencies, o Systems Manager verificará se as dependências a seguir estão instaladas em seus nós.

- Ubuntu Server/Debian Server: Apt-get (gerenciamento de pacotes), Python 3, Ansible, Unzip
- Amazon Linux: Ansible
- RHEL: Python 3, Ansible, Unzip

Se uma ou mais dessas dependências não forem encontradas, o Systems Manager as instalará automaticamente.

Criar uma associação que execute manuais do Ansible (console)

O procedimento a seguir descreve como usar o console do Systems Manager para criar uma associação do State Manager que execute manuais do Ansible usando o documento `AWS-ApplyAnsiblePlaybooks`.

Para criar uma associação que execute manuais do Ansible (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha State Manager.
3. Escolha State Manager e, em seguida, Create Association (Criar associação).
4. Em Name (Nome), especifique um nome que ajude você a lembrar a finalidade da associação.
5. Na lista Document (Documento), escolha **`AWS-ApplyAnsiblePlaybooks`**.
6. Na seção Parameters (Parâmetros), em Source Type (Tipo de origem), escolha GitHub ou S3.

   GitHub

   Se você escolher GitHub, insira as informações do repositório no seguinte formato:

```
{
   "owner":"user_name",
   "repository":"name",
   "path":"path_to_directory_or_playbook_to_download",
```

```
   "getOptions":"branch:branch_name",
   "tokenInfo":"{{(Optional)_token_information}}"
}
```

S3

Se você escolher S3, insira as informações do caminho no seguinte formato:

```
{
   "path":"https://s3.amazonaws.com/path_to_directory_or_playbook_to_download"
}
```

7. Em Install Dependencies (Instalar dependências), escolha uma opção.
8. (Opcional) Em Playbook File (Arquivo do manual), insira um nome de arquivo. Se o manual estiver contido em um arquivo Zip, especifique um caminho relativo para o arquivo Zip.
9. (Opcional) Em Variáveis extras, insira as variáveis que você deseja que o State Manager envie ao Ansible no runtime.
10. (Opcional) Em Check (Verificar), escolha uma opção.
11. (Opcional) Em Verbose (Detalhado), escolha uma opção.
12. Em Targets (Destinos), escolha uma opção. Para obter informações sobre como usar destinos, consulte Sobre destinos e controles de taxa em associações do State Manager.
13. Na seção Specify schedule (Especificar programação), escolha On Schedule (Na programação) ou No schedule (Sem programação). Se você escolher On Schedule (Na programação), use os botões fornecidos para criar uma programação cron ou rate para a associação.
14. Na seção Advanced options (Opções avançadas), em Compliance severity (Severidade de conformidade), escolha um nível de gravidade para a associação. Relatórios de conformidade indicam se o estado é compatível ou não, juntamente com o nível de gravidade que você indicar aqui. Para obter mais informações, consulte Sobre a conformidade de associações do State Manager.
15. Na seção Rate control (Controle de taxa), configure opções para executar associações do State Manager na frota de nós gerenciados. Para obter informações sobre como usar controles de taxa, consulte Sobre destinos e controles de taxa em associações do State Manager.

    Na seção Concurrency (Simultaneidade), escolha uma opção:

    - Escolha Targets (Destinos) para inserir um número absoluto de destinos que podem executar a associação simultaneamente.

- Escolha Percentage (Porcentagem) para inserir uma porcentagem do conjunto de destino que pode executar a associação simultaneamente.

Na seção Error threshold (Limite de erro), escolha uma opção:

- Escolha errors (erros) para inserir um número absoluto de erros permitidos antes de o State Manager parar de executar associações em destinos adicionais.
- Escolha percentage (porcentagem) para inserir uma porcentagem de erros permitidos antes de o State Manager parar de executar associações em destinos adicionais.

16. (Opcional) Em Opções de saída, para salvar a saída de comando em um arquivo, selecione a caixa Habilitar a gravação da saída no S3. Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

> **Note**
>
> As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil da instância atribuído ao nó gerenciado, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço IAM associada ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

17. Escolha Create Association (Criar associação).

> **Note**
>
> Se você usar tags para criar uma associação em um ou mais nós de destino e, em seguida, remover as tags de um nó, esse nó não executará mais a associação. O nó será dissociado do documento do State Manager.

Criar uma associação que execute manuais do Ansible (CLI)

O procedimento a seguir descreve como usar a AWS Command Line Interface (AWS CLI) para criar uma associação do State Manager que execute manuais do Ansible usando o documento `AWS-ApplyAnsiblePlaybooks`.

Para criar uma associação que execute manuais do Ansible (CLI)

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Execute um dos comandos a seguir para criar uma associação que execute manuais do Ansible definindo como destino nós que utilizam tags. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações. O comando (A) especifica GitHub como o tipo de origem. O comando (B) especifica o Amazon S3 como o tipo de origem.

   (A) Origem do GitHub

   Linux & macOS

```
aws ssm create-association --name "AWS-ApplyAnsiblePlaybooks" \
    --targets Key=tag:TagKey,Values=TagValue \
    --parameters '{"SourceType":["GitHub"],"SourceInfo":
["{\"owner\":\"owner_name\", \"repository\": \"name\",
 \"getOptions\": \"branch:master\"}"],"InstallDependencies":
["True_or_False"],"PlaybookFile":["file_name.yml"],"ExtraVariables":["key/
value_pairs_separated_by_a_space"],"Check":["True_or_False"],"Verbose":["-v,-
vv,-vvv, or -vvvv"],"TimeoutSeconds":["3600"]}' \
    --association-name "name" \
    --schedule-expression "cron_or_rate_expression"
```

   Windows

```
aws ssm create-association --name "AWS-ApplyAnsiblePlaybooks" ^
    --targets Key=tag:TagKey,Values=TagValue ^
    --parameters '{"SourceType":["GitHub"],"SourceInfo":
["{\"owner\":\"owner_name\", \"repository\": \"name\",
 \"getOptions\": \"branch:master\"}"],"InstallDependencies":
["True_or_False"],"PlaybookFile":["file_name.yml"],"ExtraVariables":["key/
value_pairs_separated_by_a_space"],"Check":["True_or_False"],"Verbose":["-v,-
vv,-vvv, or -vvvv"], "TimeoutSeconds":["3600"]}' ^
```

```
    --association-name "name" ^
    --schedule-expression "cron_or_rate_expression"
```

Aqui está um exemplo.

```
aws ssm create-association --name "AWS-ApplyAnsiblePlaybooks" \
    --targets "Key=tag:OS,Values=Linux" \
    --parameters '{"SourceType":["GitHub"],"SourceInfo":["{\"owner\":
\"ansibleDocumentTest\", \"repository\": \"Ansible\", \"getOptions\":
 \"branch:master\"}"],"InstallDependencies":["True"],"PlaybookFile":["hello-world-
playbook.yml"],"ExtraVariables":["SSM=True"],"Check":["False"],"Verbose":["-v"]}' \
    --association-name "AnsibleAssociation" \
    --schedule-expression "cron(0 2 ? * SUN *)"
```

(B) Origem do S3

Linux & macOS

```
aws ssm create-association --name "AWS-ApplyAnsiblePlaybooks" \
    --targets Key=tag:TagKey,Values=TagValue \
    --parameters '{"SourceType":["S3"],"SourceInfo":["{\"path\":\"https://
s3.amazonaws.com/
path_to_Zip_file,_directory,_or_playbook_to_download\"}"],"InstallDependencies":
["True_or_False"],"PlaybookFile":["file_name.yml"],"ExtraVariables":["key/
value_pairs_separated_by_a_space"],"Check":["True_or_False"],"Verbose":["-v,-
vv,-vvv, or -vvvv"]}' \
    --association-name "name" \
    --schedule-expression "cron_or_rate_expression"
```

Windows

```
aws ssm create-association --name "AWS-ApplyAnsiblePlaybooks" ^
    --targets Key=tag:TagKey,Values=TagValue ^
    --parameters '{"SourceType":["S3"],"SourceInfo":["{\"path\":\"https://
s3.amazonaws.com/
path_to_Zip_file,_directory,_or_playbook_to_download\"}"],"InstallDependencies":
["True_or_False"],"PlaybookFile":["file_name.yml"],"ExtraVariables":["key/
value_pairs_separated_by_a_space"],"Check":["True_or_False"],"Verbose":["-v,-
vv,-vvv, or -vvvv"]}' ^
    --association-name "name" ^
```

```
    --schedule-expression "cron_or_rate_expression"
```

Aqui está um exemplo.

```
aws ssm create-association --name "AWS-ApplyAnsiblePlaybooks" \
    --targets "Key=tag:OS,Values=Linux" \
    --parameters '{"SourceType":["S3"],"SourceInfo":["{\"path\":\"https://
s3.amazonaws.com/amzn-s3-demo-bucket/playbook.yml\"}"],"InstallDependencies":
["True"],"PlaybookFile":["playbook.yml"],"ExtraVariables":["SSM=True"],"Check":
["False"],"Verbose":["-v"]}' \
    --association-name "AnsibleAssociation" \
    --schedule-expression "cron(0 2 ? * SUN *)"
```

> **Note**
>
> As associações do State Manager, não comportam todas as expressões cron e de taxa. Para obter mais informações sobre como criar expressões cron e rate para associações, consulte Referência: Expressões cron e rate para o Systems Manager.

O sistema tenta criar a associação nos nós e aplicar imediatamente o estado.

3. Execute o comando a seguir para visualizar um status atualizado da associação que você acabou de criar.

```
aws ssm describe-association --association-id "ID"
```

Criar associações que executam receitas do Chef

É possível criar associações do State Manager que executam receitas do Chef usando o documento do SSM `AWS-ApplyChefRecipes`. O State Manager é uma ferramenta do AWS Systems Manager. Você pode segmentar nós gerenciados do Systems Manager baseados em Linux com o documento `AWS-ApplyChefRecipes`. Esse documento oferece os seguintes benefícios para a execução de receitas do Chef:

- Compatível com várias versões do Chef (Chef 11 a Chef 18).
- Instala automaticamente o software cliente do Chef nos nós de destino.

- Opcionalmente, executa as verificações de conformidade do Systems Manager nos nós de destino e armazena os resultados das verificações de conformidade em um bucket do Amazon Simple Storage Service (Amazon S3).
- Executa vários livros de receitas e receitas em uma única execução do documento.
- Opcionalmente, executa receitas no modo `why-run`, para mostrar quais receitas serão alteradas nos nós de destino sem fazer alterações.
- Opcionalmente aplica atributos JSON personalizados a execuções do `chef-client`.
- Opcionalmente, aplica atributos JSON personalizados de um arquivo de origem armazenado em um local que você especificou.

Você pode usar o Git, o GitHub, HTTP ou buckets do Amazon S3 como origens para cookbooks e fórmulas do Chef especificados em um documento `AWS-ApplyChefRecipes`.

> **Note**
>
> As associações que executam receitas do Chef não são compatíveis com o macOS.

Começar

Antes de criar um documento `AWS-ApplyChefRecipes`, prepare seus livros de receitas e o repositório de livros de receitas do Chef. Se ainda não tiver um livro de receitas do Chef que deseja usar, você pode começar usando um livro de receitas `HelloWorld` de teste que a AWS preparou para você. O documento `AWS-ApplyChefRecipes` já aponta para esse livro de receitas por padrão. Seus livros de receitas devem ser configurados de forma semelhante à seguinte estrutura de diretório. No exemplo a seguir, `jenkins` e `nginx` são exemplos de livros de receitas do Chef que estão disponíveis no Chef Supermarket no site do Chef.

Embora a AWS não possa oferecer suporte oficialmente a livros de receitas no site do Chef Supermarket, muitos deles trabalham com o documento `AWS-ApplyChefRecipes`. Veja a seguir exemplos de critérios a serem verificados ao testar um livro de receitas da comunidade:

- O livro de receitas deve oferecer suporte aos sistemas operacionais baseados em Linux dos nós gerenciados do Systems Manager que você tem como destino.
- O cookbook deve ser válido para a versão do cliente do Chef (Chef 11 a Chef 18) que você usa.
- O livro de receitas é compatível com o Chef Infra Client e não requer um servidor do Chef.

Verifique se você pode acessar o site `Chef.io` para que todos os livros de receitas que você especificar na lista de execução possam ser instalados quando o documento do Systems Manager (documento SSM) for executado. O uso de uma pasta `cookbooks` aninhada é compatível, mas não é necessário; você pode armazenar livros de receitas diretamente sob o nível raiz.

```
<Top-level directory, or the top level of the archive file (ZIP or tgz or tar.gz)>
    ### cookbooks (optional level)
        ### jenkins
        #   ### metadata.rb
        #   ### recipes
        ### nginx
            ### metadata.rb
            ### recipes
```

Important

Antes de criar uma associação do State Manager que executa receitas do Chef, lembre-se de que a execução do documento instala o software cliente do Chef em seus nós gerenciados do Systems Manager, a menos que você defina o valor de Versão do cliente do Chef como `None`. Essa ação usa um script de instalação do Chef para instalar componentes do Chef em seu nome. Antes de executar um documento `AWS-ApplyChefRecipes`, certifique-se de que sua empresa pode cumprir todos os requisitos legais aplicáveis, incluindo termos de licença aplicáveis ao uso do software Chef. Para obter mais informações, acesse o site do Chef.

O Systems Manager pode fornecer relatórios de conformidade para um bucket do S3, para o console do Systems Manager ou disponibilizar resultados da conformidade em resposta a comandos da API do Systems Manager. Para executar relatórios de conformidade do Systems Manager, o perfil da instância anexado a nós gerenciados pelo Systems Manager deve ter permissões para gravar no bucket do S3. O perfil de instância deve ter permissões para usar a API `PutComplianceItem` do Systems Manager. Para obter mais informações sobre a conformidade do Systems Manager, consulte AWS Systems Manager Compliance.

Registrar a execução do documento em log

Quando você executa um documento do Systems Manager (documento SSM) usando uma associação do State Manager você pode configurar a associação para escolher a saída da execução do documento e pode enviar a saída para o Amazon S3 ou Amazon CloudWatch Logs (CloudWatch

Logs). Para ajudar a facilitar a solução de problemas quando a execução de uma associação for concluída, verifique se a associação está configurada para gravar a saída do comando em um bucket do Amazon S3 ou do CloudWatch Logs. Para obter mais informações, consulte Para obter informações, consulte Trabalhar com associações no Systems Manager..

Aplicar atributos JSON aos destinos ao executar uma fórmula

Você pode especificar atributos JSON para seu cliente do Chef aplicar aos nós de destino durante uma execução de associação. Ao configurar a associação, é possível fornecer JSON bruto ou fornecer o caminho para um arquivo JSON armazenado no Amazon S3.

Use atributos JSON quando quiser personalizar como a fórmula é executada sem precisar modificar a fórmula em si, por exemplo:

- Substituir poucos atributos

  Use JSON personalizado para evitar a necessidade de manter múltiplas versões de uma fórmula para ajustar pequenas diferenças.
- Fornecer valores variáveis

  Use JSON personalizado para especificar valores que podem ser alterados entre execuções. Por exemplo, se seus cookbooks do Chef configurarem uma aplicação de terceiros que aceite pagamentos, você poderá usar JSON personalizado para especificar o URL do endpoint de pagamento.

Especificar atributos em JSON bruto

Veja a seguir um exemplo do formato que você pode usar para especificar atributos JSON personalizados para sua receita do Chef.

```
{"filepath":"/tmp/example.txt", "content":"Hello, World!"}
```

Especificar um caminho para um arquivo JSON

Veja a seguir um exemplo do formato que você pode usar para especificar o caminho para atributos JSON personalizados para sua receita do Chef.

```
{"sourceType":"s3", "sourceInfo":"someS3URL1"}, {"sourceType":"s3",
 "sourceInfo":"someS3URL2"}
```

Usar o Git como a origem do cookbook

O documento `AWS-ApplyChefRecipes` usa o plugin aws:downloadContent para fazer download de livros de receitas do Chef. Para baixar conteúdo do Git, especifique informações sobre seu repositório Git no formato JSON, como no exemplo a seguir. Substitua cada *example-resource-placeholder* por suas próprias informações.

```
{
   "repository":"GitCookbookRepository",
   "privateSSHKey":"{{ssm-secure:ssh-key-secure-string-parameter}}",
   "skipHostKeyChecking":"false",
   "getOptions":"branch:refs/head/main",
   "username":"{{ssm-secure:username-secure-string-parameter}}",
   "password":"{{ssm-secure:password-secure-string-parameter}}"
}
```

Usar o GitHub como a origem de livros de receitas

O documento `AWS-ApplyChefRecipes` usa o plugin aws:downloadContent para fazer download de livros de receitas. Para baixar conteúdo do GitHub, especifique informações sobre seu repositório GitHub no formato JSON, como no exemplo a seguir. Substitua cada *example-resource-placeholder* por suas próprias informações.

```
{
   "owner":"TestUser",
   "repository":"GitHubCookbookRepository",
   "path":"cookbooks/HelloWorld",
   "getOptions":"branch:refs/head/main",
   "tokenInfo":"{{ssm-secure:token-secure-string-parameter}}"
}
```

Usar o HTTP como a origem do cookbook

Você pode armazenar cookbooks do Chef em um local HTTP personalizado como um único arquivo `.zip` ou `tar.gz` ou uma estrutura de diretórios. Para baixar conteúdo de HTTP, é necessário especificar o caminho para o arquivo ou diretório no formato JSON, como no exemplo a seguir. Substitua cada *example-resource-placeholder* por suas próprias informações.

```
{
   "url":"https://my.website.com/chef-cookbooks/HelloWorld.zip",
```

```
    "allowInsecureDownload":"false",
    "authMethod":"Basic",
    "username":"{{ssm-secure:username-secure-string-parameter}}",
    "password":"{{ssm-secure:password-secure-string-parameter}}"
}
```

Usar o Amazon S3 como a origem de livros de receitas

Também é possível armazenar e baixar cookbooks do Chef no Amazon S3 como um único arquivo `.zip` ou `tar.gz` ou como uma estrutura de diretórios. Para baixar conteúdo do Amazon S3, é necessário especificar o caminho para o arquivo no formato JSON, como nos exemplos a seguir. Substitua cada *example-resource-placeholder* por suas próprias informações.

Exemplo 1: Fazer download de um livro de receitas específico

```
{
    "path":"https://s3.amazonaws.com/chef-cookbooks/HelloWorld.zip"
}
```

Exemplo 2: fazer download do conteúdo de um diretório

```
{
    "path":"https://s3.amazonaws.com/chef-cookbooks-test/HelloWorld"
}
```

> **Important**
>
> Se você especificar o Amazon S3, o perfil de instância do AWS Identity and Access Management (IAM) nos nós gerenciados deverá ser configurado com a política `AmazonS3ReadOnlyAccess`. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager.

Criar uma associação que executa receitas do Chef (console)

O procedimento a seguir descreve como usar o console do Systems Manager para criar uma associação do State Manager que execute livros de receitas do Chef usando o documento `AWS-ApplyChefRecipes`.

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.

2. No painel de navegação, escolha State Manager.
3. Escolha State Manager e, em seguida, Create Association (Criar associação).
4. Em Name (Nome), insira um nome que ajude você a lembrar a finalidade da associação.
5. Na lista Document (Documento), escolha **`AWS-ApplyChefRecipes`**.
6. Em Parâmetros, em Tipo de origem, selecione Git, GitHub, HTTP ou S3.
7. Em Informações da origem, insira as informações da origem do cookbook usando o formato apropriado para o Tipo de origem que você selecionou na etapa 6. Para obter mais informações, consulte os tópicos a seguir.

   - the section called “Usar o Git como a origem do cookbook”
   - the section called “Usar o GitHub como a origem de livros de receitas”
   - the section called “Usar o HTTP como a origem do cookbook”
   - the section called “Usar o Amazon S3 como a origem de livros de receitas”
8. Em Run list (Lista de execução), liste as receitas que você deseja executar no seguinte formato, separando cada receita com uma vírgula, como mostrado. Não inclua um espaço após a vírgula. Substitua cada *`example-resource-placeholder`* por suas próprias informações.

   ```
   recipe[cookbook-name1::recipe-name],recipe[cookbook-name2::recipe-name]
   ```
9. (Opcional) Especifique atributos JSON que você deseja que o cliente do Chef passe para seus nós de destino.

   a. Em Conteúdo de atributos JSON, adicione os atributos que você deseja que o cliente do Chef passe para seus nós de destino.

   b. Em Origens de atributos JSON, adicione caminhos para atributos que você deseja que o cliente do Chef passe para seus nós de destino.

   Para obter mais informações, consulte the section called “Aplicar atributos JSON aos destinos ao executar uma fórmula”.
10. Em Versão do cliente do Chef, especifique uma versão do Chef. Os valores válidos são 11 a 18 ou `None`. Se você especificar em número de 11 a 18 (inclusive), o Systems Manager instalará a versão correta do cliente do Chef em seus nós de destino. Se você especificar `None`, o Systems Manager não instalará o cliente do Chefem nós de destino antes de executar as receitas do documento.

11. (Opcional) Em Argumentos do cliente do Chef, especifique argumentos adicionais que sejam compatíveis com a versão do Chef que você está usando. Para saber mais sobre os argumentos compatíveis, execute `chef-client -h` em um nó que esteja executando o cliente do Chef.
12. (Opcional) Ative o Why-Run (Por que executar) para mostrar as alterações feitas nos nós de destino, se as receitas forem executadas, sem realmente alterar os nós de destino.
13. Em Compliance severity (Gravidade da conformidade), escolha a gravidade dos resultados da Conformidade so Systems Manager que você deseja relatar. Os relatórios de conformidade indicam se o estado é compatível ou não, juntamente com o nível de gravidade que você especificar. Os relatórios de conformidade são armazenados em um bucket do S3 especificado como o valor do parâmetro Compliance report bucket (Bucket de relatório de conformidade) (etapa 14). Para obter mais informações sobre a Conformidade da configuração, consulte Conheça mais sobre a conformidade neste guia.

    As verificações de conformidade medem a variação entre a configuração especificada nas receitas do Chef e nos recursos do nó. Os valores válidos são `Critical`, `High`, `Medium`, `Low`, `Informational`, `Unspecified` ou `None`. Para ignorar os relatórios de conformidade, escolha `None`.
14. Em Compliance type (Tipo de conformidade), especifique o tipo de conformidade para o qual você deseja que os resultados sejam relatados. Os valores válidos são `Association` para associações do State Manager ou `Custom:`*`custom_type`*. O valor padrão é `Custom:Chef`.
15. Em Bucket do relatório de conformidade, insira o nome de um bucket do S3 no qual armazenar informações sobre cada execução do Chef realizada por este documento, incluindo a configuração de recursos e os resultados da Conformidade.
16. Em Rate control (Controle de taxa), configure opções para executar associações do State Manager na frota de nós gerenciados. Para obter informações sobre como usar controles de taxa, consulte Sobre destinos e controles de taxa em associações do State Manager.

    Em Concurrency (Simultaneidade), escolha uma opção:

    - Escolha Targets (Destinos) para inserir um número absoluto de destinos que podem executar a associação simultaneamente.
    - Escolha Percentage (Porcentagem) para inserir uma porcentagem do conjunto de destino que pode executar a associação simultaneamente.

    Em Error threshold (Limitação de erros), escolha uma opção:

- Escolha errors (erros) para inserir um número absoluto de erros permitidos antes de o State Manager parar de executar associações em destinos adicionais.
- Escolha percentage (porcentagem) para inserir uma porcentagem de erros permitidos antes de o State Manager parar de executar associações em destinos adicionais.

17. (Opcional) Em Opções de saída, para salvar a saída de comando em um arquivo, selecione a caixa Habilitar a gravação da saída no S3. Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

> **Note**
>
> As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil da instância atribuído ao nó gerenciado, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço IAM associada ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

18. Escolha Create Association (Criar associação).

Criar uma associação que executa receitas do Chef (CLI)

O procedimento a seguir descreve como usar a AWS Command Line Interface (AWS CLI) para criar uma associação do State Manager que executa livros de receitas do Chef usando o documento `AWS-ApplyChefRecipes`.

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Execute um dos comandos a seguir para criar uma associação que execute cookbooks do Chef em nós de destino que tenham as etiquetas especificadas. Use o comando apropriado para o tipo de origem e sistema operacional do cookbook. Substitua cada *`example-resource-placeholder`* por suas próprias informações.

   a. Origem Git

Linux & macOS

```
aws ssm create-association --name "AWS-ApplyChefRecipes" \
    --targets Key=tag:TagKey,Values=TagValue \
    --parameters '{"SourceType":["Git"],"SourceInfo":["{\"repository\":
\"repository-name\", \"getOptions\": \"branch:branch-name\", \"username
\": \"{{ ssm-secure:username-secure-string-parameter }}\", \"password\":
 \"{{ ssm-secure:password-secure-string-parameter }}\"}"], "RunList":
["{\"recipe[cookbook-name-1::recipe-name]\", \"recipe[cookbook-
name-2::recipe-name]\"}"], "JsonAttributesContent": ["{custom-json-
content}"], "JsonAttributesSources": "{\"sourceType\":\"s3\", \"sourceInfo
\":\"s3-bucket-endpoint-1\"}, {\"sourceType\":\"s3\", \"sourceInfo\":
\"s3-bucket-endpoint-2\"}", "ChefClientVersion": ["version-number"],
 "ChefClientArguments":["{chef-client-arguments}"], "WhyRun": boolean,
 "ComplianceSeverity": ["severity-value"], "ComplianceType":
 ["Custom:Chef"], "ComplianceReportBucket": ["s3-bucket-name"]}' \
    --association-name "name" \
    --schedule-expression "cron-or-rate-expression"
```

Windows

```
aws ssm create-association --name "AWS-ApplyChefRecipes" ^
    --targets Key=tag:TagKey,Values=TagValue ^
    --parameters '{"SourceType":["Git"],"SourceInfo":["{\"repository\":
\"repository-name\", \"getOptions\": \"branch:branch-name\", \"username
\": \"{{ ssm-secure:username-secure-string-parameter }}\", \"password\":
 \"{{ ssm-secure:password-secure-string-parameter }}\"}"], "RunList":
["{\"recipe[cookbook-name-1::recipe-name]\", \"recipe[cookbook-
name-2::recipe-name]\"}"], "JsonAttributesContent": ["{custom-json}"],
 "JsonAttributesSources": "{\"sourceType\":\"s3\", \"sourceInfo\":
\"s3-bucket-endpoint-1\"}, {\"sourceType\":\"s3\", \"sourceInfo\":
\"s3-bucket-endpoint-2\"}", "ChefClientVersion": ["version-number"],
 "ChefClientArguments":["{chef-client-arguments}"], "WhyRun": boolean,
 "ComplianceSeverity": ["severity-value"], "ComplianceType":
 ["Custom:Chef"], "ComplianceReportBucket": ["s3-bucket-name"]}' ^
    --association-name "name" ^
    --schedule-expression "cron-or-rate-expression"
```

b. Origem do GitHub

Linux & macOS

```
aws ssm create-association --name "AWS-ApplyChefRecipes" \
    --targets Key=tag:TagKey,Values=TagValue \
    --parameters '{"SourceType":["GitHub"],"SourceInfo":["{\"owner\":
\"owner-name\", \"repository\": \"name\", \"path\": \"path-to-directory-
or-cookbook-to-download\", \"getOptions\": \"branch:branch-name\"}"],
 "RunList":["{\"recipe[cookbook-name-1::recipe-name]\", \"recipe[cookbook-
name-2::recipe-name]\"}"], "JsonAttributesContent": ["{custom-json}"],
 "ChefClientVersion": ["version-number"], "ChefClientArguments":["{chef-
client-arguments}"], "WhyRun": boolean, "ComplianceSeverity": ["severity-
value"], "ComplianceType": ["Custom:Chef"], "ComplianceReportBucket": ["s3-
bucket-name"]}' \
    --association-name "name" \
    --schedule-expression "cron-or-rate-expression"
```

Windows

```
aws ssm create-association --name "AWS-ApplyChefRecipes" ^
    --targets Key=tag:TagKey,Values=TagValue \
    --parameters '{"SourceType":["GitHub"],"SourceInfo":["{\"owner\":
\"owner-name\", \"repository\": \"name\", \"path\": \"path-to-directory-
or-cookbook-to-download\", \"getOptions\": \"branch:branch-name\"}"],
 "RunList":["{\"recipe[cookbook-name-1::recipe-name]\", \"recipe[cookbook-
name-2::recipe-name]\"}"], "JsonAttributesContent": ["{custom-json}"],
 "ChefClientVersion": ["version-number"], "ChefClientArguments":["{chef-
client-arguments}"], "WhyRun": boolean, "ComplianceSeverity": ["severity-
value"], "ComplianceType": ["Custom:Chef"], "ComplianceReportBucket": ["s3-
bucket-name"]}' ^
    --association-name "name" ^
    --schedule-expression "cron-or-rate-expression"
```

Aqui está um exemplo.

Linux & macOS

```
aws ssm create-association --name "AWS-ApplyChefRecipes" \
    --targets Key=tag:OS,Values=Linux \
    --parameters '{"SourceType":["GitHub"],"SourceInfo":["{\"owner
\":\"ChefRecipeTest\", \"repository\": \"ChefCookbooks\", \"path
```

```
\": \"cookbooks/HelloWorld\", \"getOptions\": \"branch:master
\"}"], "RunList":["{\"recipe[HelloWorld::HelloWorldRecipe]\",
 \"recipe[HelloWorld::InstallApp]\"}"], "JsonAttributesContent":
 ["{\"state\": \"visible\",\"colors\": {\"foreground\": \"light-blue
\",\"background\": \"dark-gray\"}}"], "ChefClientVersion": ["14"],
 "ChefClientArguments":["{--fips}"], "WhyRun": false, "ComplianceSeverity":
 ["Medium"], "ComplianceType": ["Custom:Chef"], "ComplianceReportBucket":
 ["ChefComplianceResultsBucket"]}' \
    --association-name "MyChefAssociation" \
    --schedule-expression "cron(0 2 ? * SUN *)"
```

Windows

```
aws ssm create-association --name "AWS-ApplyChefRecipes" ^
    --targets Key=tag:OS,Values=Linux ^
    --parameters '{"SourceType":["GitHub"],"SourceInfo":["{\"owner
\":\"ChefRecipeTest\", \"repository\": \"ChefCookbooks\", \"path
\": \"cookbooks/HelloWorld\", \"getOptions\": \"branch:master
\"}"], "RunList":["{\"recipe[HelloWorld::HelloWorldRecipe]\",
 \"recipe[HelloWorld::InstallApp]\"}"], "JsonAttributesContent":
 ["{\"state\": \"visible\",\"colors\": {\"foreground\": \"light-blue
\",\"background\": \"dark-gray\"}}"], "ChefClientVersion": ["14"],
 "ChefClientArguments":["{--fips}"], "WhyRun": false, "ComplianceSeverity":
 ["Medium"], "ComplianceType": ["Custom:Chef"], "ComplianceReportBucket":
 ["ChefComplianceResultsBucket"]}' ^
    --association-name "MyChefAssociation" ^
    --schedule-expression "cron(0 2 ? * SUN *)"
```

c. Origem HTTP

Linux & macOS

```
aws ssm create-association --name "AWS-ApplyChefRecipes" \
    --targets Key=tag:TagKey,Values=TagValue \
    --parameters '{"SourceType":["HTTP"],"SourceInfo":["{\"url\":\"url-
to-zip-file|directory|cookbook\", \"authMethod\": \"auth-method\",
 \"username\": \"{{ ssm-secure:username-secure-string-parameter }}\",
 \"password\": \"{{ ssm-secure:password-secure-string-parameter }}\"}"],
 "RunList":["{\"recipe[cookbook-name-1::recipe-name]\", \"recipe[cookbook-
name-2::recipe-name]\"}"], "JsonAttributesContent": ["{custom-json-
content}"], "JsonAttributesSources": "{\"sourceType\":\"s3\", \"sourceInfo
\":\"s3-bucket-endpoint-1\"}, {\"sourceType\":\"s3\", \"sourceInfo\":
\"s3-bucket-endpoint-2\"}", "ChefClientVersion": ["version-number"],
```

```
 "ChefClientArguments":["{chef-client-arguments}"], "WhyRun": boolean,
 "ComplianceSeverity": ["severity-value"], "ComplianceType":
 ["Custom:Chef"], "ComplianceReportBucket": ["s3-bucket-name"]}' \
    --association-name "name" \
    --schedule-expression "cron-or-rate-expression"
```

Windows

```
aws ssm create-association --name "AWS-ApplyChefRecipes" ^
    --targets Key=tag:TagKey,Values=TagValue ^
    --parameters '{"SourceType":["HTTP"],"SourceInfo":["{\"url\":\"url-
to-zip-file|directory|cookbook\", \"authMethod\": \"auth-method\",
 \"username\": \"{{ ssm-secure:username-secure-string-parameter }}\",
 \"password\": \"{{ ssm-secure:password-secure-string-parameter }}\"}"],
 "RunList":["{\"recipe[cookbook-name-1::recipe-name]\", \"recipe[cookbook-
name-2::recipe-name]\"}"], "JsonAttributesContent": ["{custom-json-
content}"], "JsonAttributesSources": "{\"sourceType\":\"s3\", \"sourceInfo
\":\"s3-bucket-endpoint-1\"}, {\"sourceType\":\"s3\", \"sourceInfo\":
\"s3-bucket-endpoint-2\"}", "ChefClientVersion": ["version-number"],
 "ChefClientArguments":["{chef-client-arguments}"], "WhyRun": boolean,
 "ComplianceSeverity": ["severity-value"], "ComplianceType":
 ["Custom:Chef"], "ComplianceReportBucket": ["s3-bucket-name"]}' \
    --association-name "name" ^
    --schedule-expression "cron-or-rate-expression"
```

d. Origem Amazon S3

Linux & macOS

```
aws ssm create-association --name "AWS-ApplyChefRecipes" \
    --targets Key=tag:TagKey,Values=TagValue \
    --parameters '{"SourceType":["S3"],"SourceInfo":["{\"path\":\"https://
s3.amazonaws.com/path_to_Zip_file,_directory,_or_cookbook_to_download\"}"],
 "RunList":["{\"recipe[cookbook_name1::recipe_name]\",
 \"recipe[cookbook_name2::recipe_name]\"}"], "JsonAttributesContent":
 ["{Custom_JSON}"], "ChefClientVersion": ["version_number"],
 "ChefClientArguments":["{chef_client_arguments}"], "WhyRun": true_or_false,
 "ComplianceSeverity": ["severity_value"], "ComplianceType":
 ["Custom:Chef"], "ComplianceReportBucket": ["amzn-s3-demo-bucket"]}' \
    --association-name "name" \
    --schedule-expression "cron_or_rate_expression"
```

Windows

```
aws ssm create-association --name "AWS-ApplyChefRecipes" ^
    --targets Key=tag:TagKey,Values=TagValue ^
    --parameters '{"SourceType":["S3"],"SourceInfo":["{\"path\":\"https://
s3.amazonaws.com/path_to_Zip_file,_directory,_or_cookbook_to_download\"}"],
 "RunList":["{\"recipe[cookbook_name1::recipe_name]\",
 \"recipe[cookbook_name2::recipe_name]\"}"], "JsonAttributesContent":
 ["{Custom_JSON}"], "ChefClientVersion": ["version_number"],
 "ChefClientArguments":["{chef_client_arguments}"], "WhyRun": true_or_false,
 "ComplianceSeverity": ["severity_value"], "ComplianceType":
 ["Custom:Chef"], "ComplianceReportBucket": ["amzn-s3-demo-bucket"]}' ^
    --association-name "name" ^
    --schedule-expression "cron_or_rate_expression"
```

Aqui está um exemplo.

Linux & macOS

```
aws ssm create-association --name "AWS-ApplyChefRecipes" \
    --targets "Key=tag:OS,Values= Linux" \
    --parameters '{"SourceType":["S3"],"SourceInfo":["{\"path
\":\"https://s3.amazonaws.com/amzn-s3-demo-bucket/HelloWorld
\"}"], "RunList":["{\"recipe[HelloWorld::HelloWorldRecipe]\",
 \"recipe[HelloWorld::InstallApp]\"}"], "JsonAttributesContent":
 ["{\"state\": \"visible\",\"colors\": {\"foreground\": \"light-blue
\",\"background\": \"dark-gray\"}}"], "ChefClientVersion": ["14"],
 "ChefClientArguments":["{--fips}"], "WhyRun": false, "ComplianceSeverity":
 ["Medium"], "ComplianceType": ["Custom:Chef"], "ComplianceReportBucket":
 ["ChefComplianceResultsBucket"]}' \
    --association-name "name" \
    --schedule-expression "cron(0 2 ? * SUN *)"
```

Windows

```
aws ssm create-association --name "AWS-ApplyChefRecipes" ^
    --targets "Key=tag:OS,Values= Linux" ^
    --parameters '{"SourceType":["S3"],"SourceInfo":["{\"path
\":\"https://s3.amazonaws.com/amzn-s3-demo-bucket/HelloWorld
\"}"], "RunList":["{\"recipe[HelloWorld::HelloWorldRecipe]\",
 \"recipe[HelloWorld::InstallApp]\"}"], "JsonAttributesContent":
```

```
 ["{\"state\": \"visible\",\"colors\": {\"foreground\": \"light-blue
\",\"background\": \"dark-gray\"}}"], "ChefClientVersion": ["14"],
 "ChefClientArguments":["{--fips}"], "WhyRun": false, "ComplianceSeverity":
 ["Medium"], "ComplianceType": ["Custom:Chef"], "ComplianceReportBucket":
 ["ChefComplianceResultsBucket"]}' ^
    --association-name "name" ^
    --schedule-expression "cron(0 2 ? * SUN *)"
```

O sistema criará a associação e, a menos que sua expressão cron ou de taxa especificada a impeça, o sistema executará a associação nos nós de destino.

> **Note**
>
> As associações do State Manager, não comportam todas as expressões cron e de taxa. Para obter mais informações sobre como criar expressões cron e rate para associações, consulte Referência: Expressões cron e rate para o Systems Manager.

3. Execute o comando a seguir para visualizar um status da associação que você acabou de criar.

```
aws ssm describe-association --association-id "ID"
```

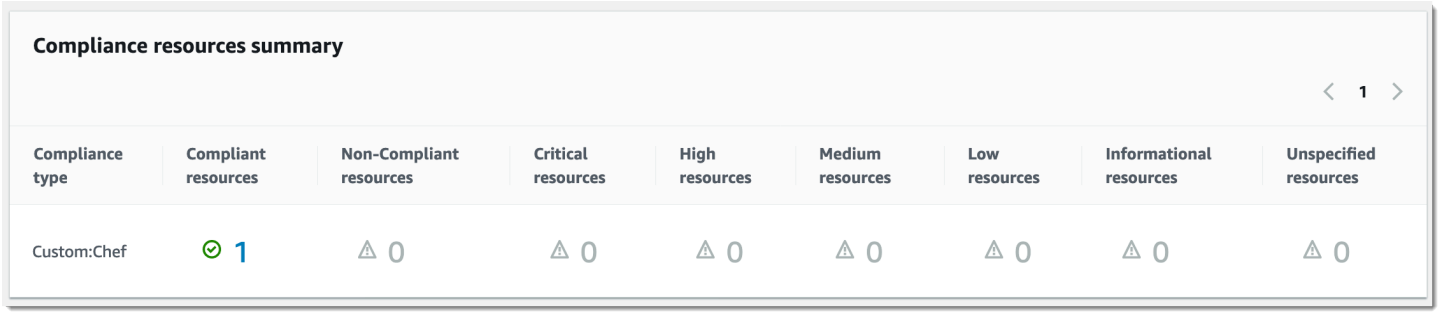
Visualizar detalhes de conformidade de recursos do Chef

O Systems Manager captura informações de conformidade sobre recursos gerenciados pelo Chef no valor de Bucket de relatórios de conformidade do Amazon S3 que você especificou ao executar o documento `AWS-ApplyChefRecipes`. Pesquisar informações sobre falhas de recursos do Chef em um bucket do S3 pode ser demorado. Em vez disso, você pode exibir essas informações na página Compliance (Conformidade) do Systems Manager.

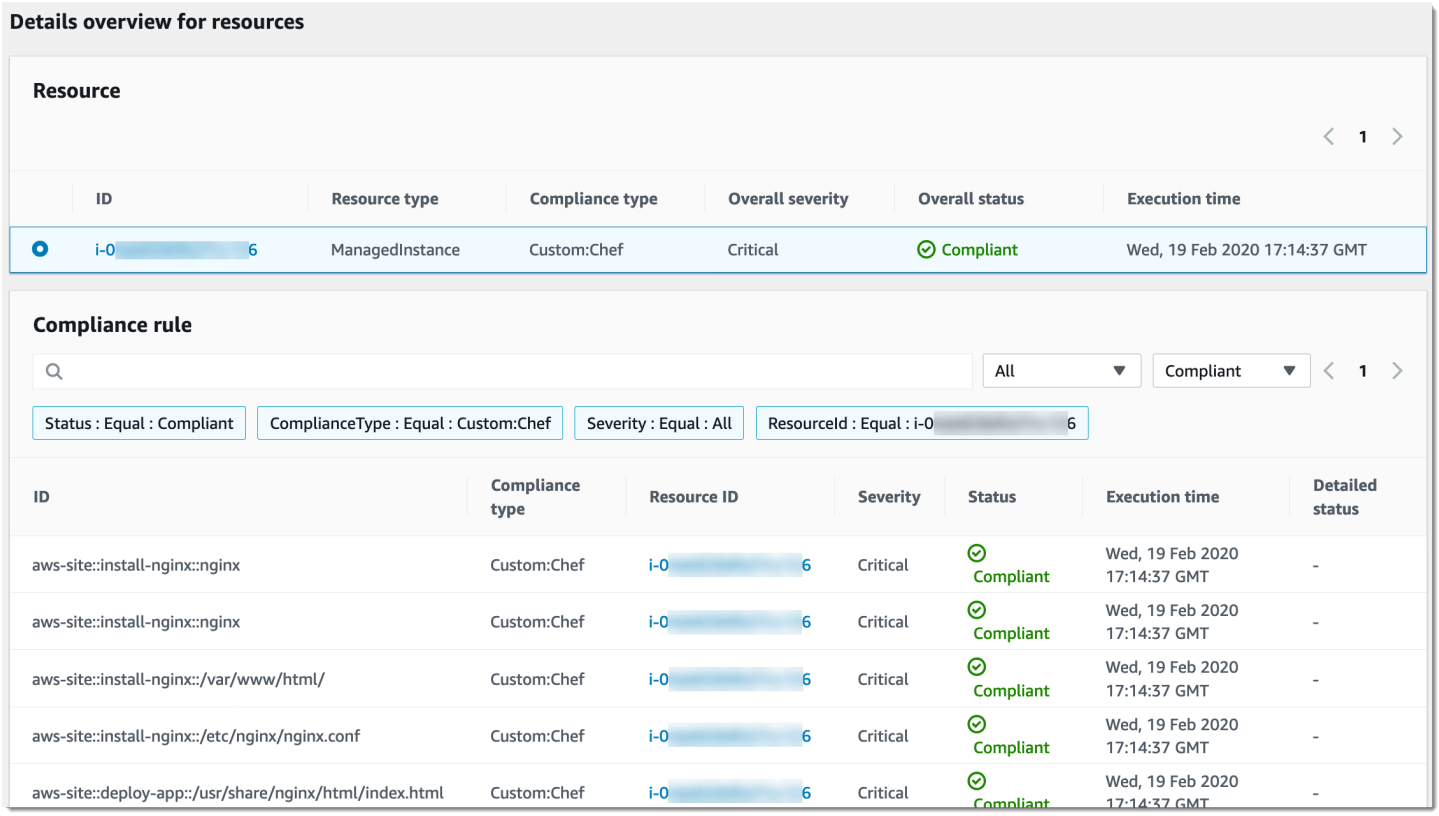
Uma verificação de conformidade do Systems Manager coleta informações sobre recursos em seus nós gerenciados que foram criados ou verificados na execução mais recente do Chef. Os recursos podem incluir arquivos, diretórios, serviços do `systemd`, pacotes do `yum`, arquivos de modelos, pacotes do `gem` e livros de receitas dependentes, entre outros.

A seção Compliance resources summary (Resumo dos recursos de conformidade) exibe uma contagem do recursos com falha. No exemplo a seguir, ComplianceType é Custom:Chef e um recurso não está em conformidade.

> **Note**
>
> `Custom:Chef` é o valor padrão de ComplianceType no documento `AWS-ApplyChefRecipes`. Esse valor é personalizável.



A seção Details overview for resources (Visão geral de detalhes dos recursos) mostra informações sobre o recurso da AWS que não está em conformidade. Essa seção também inclui o tipo de recurso do Chef em relação ao qual a conformidade foi executada, a gravidade do problema, o status da conformidade e os links para mais informações quando aplicável.



Visualizar saída mostra os últimos 4.000 caracteres do status detalhado. O Systems Manager começa com a exceção como o primeiro elemento, localiza as mensagens detalhadas e as exibe até atingir a cota de 4.000 caracteres. Esse processo exibe as mensagens de log que foram registradas antes da exceção ser gerada, que são as mensagens mais relevantes para a solução de problemas.

Para obter informações sobre como visualizar informações de conformidade, consulte AWS Systems Manager Compliance.

> **Important**
>
> Se houver falha na associação do State Manager, os dados de conformidade não serão relatados. Por exemplo, se o Systems Manager tentar baixar um livro de receitas do Chef em um bucket do S3 que o nó não tem permissão para acessar, haverá falha na associação e o Systems Manager relatará dados fora de conformidade.

Walkthrough: Automatically update SSM Agent with the AWS CLI

O procedimento a seguir demonstra o processo de criação de uma associação do State Manager usando a AWS Command Line Interface. A associação atualiza automaticamente o SSM Agent de acordo com uma programação que você especifica. Para obter mais informações sobre o SSM Agent, consulte Trabalhar com o SSM Agent. Para personalizar o agendamento de atualização do SSM Agent usando o console, consulte Atualizar automaticamente o SSM Agent.

Para receber notificações sobre atualizações do SSM Agent, inscreva-se na página Notas de versão do SSM Agent no GitHub.

Antes de começar

Antes de concluir o procedimento a seguir, verifique se você tem pelo menos uma instância do Amazon Elastic Compute Cloud (Amazon EC2) para o Linux, macOS ou Windows Server em execução configurada para o Systems Manager. Para obter mais informações, consulte Configurar nós gerenciados para o AWS Systems Manager.

Se você criar uma associação usando a AWS CLI ou o AWS Tools for Windows PowerShell, use o parâmetro `--Targets` para as instâncias de destino, conforme mostrado no exemplo a seguir. Não use o parâmetro `--InstanceID`. O parâmetro `--InstanceID` é um parâmetro legado.

Para criar uma associação para atualizar automaticamente o SSM Agent

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Execute o comando a seguir para criar uma associação com instâncias como destinos com etiquetas do Amazon Elastic Compute Cloud (Amazon EC2). Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações. O parâmetro `Schedule` define um agendamento para executar a associação todos os domingos de manhã às 2h. (UTC).

   As associações do State Manager, não comportam todas as expressões cron e de taxa. Para obter mais informações sobre como criar expressões cron e rate para associações, consulte Referência: Expressões cron e rate para o Systems Manager.

   Linux & macOS

   ```
   aws ssm create-association \
   --targets Key=tag:tag_key,Values=tag_value \
   --name AWS-UpdateSSMAgent \
   --schedule-expression "cron(0 2 ? * SUN *)"
   ```

   Windows

   ```
   aws ssm create-association ^
   --targets Key=tag:tag_key,Values=tag_value ^
   --name AWS-UpdateSSMAgent ^
   --schedule-expression "cron(0 2 ? * SUN *)"
   ```

   Se você quiser, também poderá especificar várias instâncias de destino indicando os IDs de instâncias em uma lista separada por vírgulas.

   Linux & macOS

   ```
   aws ssm create-association \
   --targets Key=instanceids,Values=instance_ID,instance_ID,instance_ID \
   --name AWS-UpdateSSMAgent \
   --schedule-expression "cron(0 2 ? * SUN *)"
   ```

Windows

```
aws ssm create-association ^
--targets Key=instanceids,Values=instance_ID,instance_ID,instance_ID ^
--name AWS-UpdateSSMAgent ^
--schedule-expression "cron(0 2 ? * SUN *)"
```

Você pode especificar a versão do SSM Agent para a qual deseja atualizar.

Linux & macOS

```
aws ssm create-association \
--targets Key=instanceids,Values=instance_ID,instance_ID,instance_ID \
--name AWS-UpdateSSMAgent \
--schedule-expression "cron(0 2 ? * SUN *)" \
--parameters version=ssm_agent_version_number
```

Windows

```
aws ssm create-association ^
--targets Key=instanceids,Values=instance_ID,instance_ID,instance_ID ^
--name AWS-UpdateSSMAgent ^
--schedule-expression "cron(0 2 ? * SUN *)" ^
--parameters version=ssm_agent_version_number
```

O sistema retorna informações como estas.

```
{
    "AssociationDescription": {
        "ScheduleExpression": "cron(0 2 ? * SUN *)",
        "Name": "AWS-UpdateSSMAgent",
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "AssociationId": "123.............",
        "DocumentVersion": "$DEFAULT",
        "LastUpdateAssociationDate": 1504034257.98,
        "Date": 1504034257.98,
```

```
        "AssociationVersion": "1",
        "Targets": [
            {
                "Values": [
                    "TagValue"
                ],
                "Key": "tag:TagKey"
            }
        ]
    }
}
```

O sistema tenta criar a associação na(s) instância(s) e aplicar imediatamente o estado. O status da associação mostra `Pending`.

3. Execute o comando a seguir para visualizar um status atualizado da associação que você acabou de criar.

```
aws ssm list-associations
```

Se suas instâncias não estiverem executando a versão mais recente do SSM Agent, o status mostrará `Failed`. Quando uma nova versão do SSM Agent é publicada, a associação instala automaticamente o novo agente e o status mostra `Success`.

Walkthrough: Automatically update PV drivers on EC2 instances for Windows Server

As Amazon Machine Images (AMIs) do Windows da Amazon contêm um conjunto de drivers para permitir acesso ao hardware virtualizado. Esses drivers são usados pelo Amazon Elastic Compute Cloud (Amazon EC2) para mapear armazenamento de instâncias e volumes do Amazon Elastic Block Store (Amazon EBS) para seus dispositivos. É recomendável instalar os drivers mais recentes para melhorar a estabilidade e a performance de suas instâncias do EC2 para Windows Server. Para obter mais informações sobre drivers do PV, consulte AWS PV Drivers (Drivers do PV).

A demonstração a seguir mostra como configurar uma associação do State Manager para baixar e instalar automaticamente novos drivers do AWS PV quando eles estiverem disponíveis. O State Manager é uma ferramenta do AWS Systems Manager.

Antes de começar

Antes de concluir o procedimento a seguir, verifique se você tem pelo menos uma instância do Amazon EC2 para o Windows Server em execução configurada para o Systems Manager. Para obter mais informações, consulte Configurar nós gerenciados para o AWS Systems Manager.

Para criar uma associação do State Manager que atualiza automaticamente os drivers do PV

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha State Manager.
3. Selecione Criar associação.
4. No campo Nome, insira um nome descritivo para a associação.
5. Na lista Document (Documento), escolha `AWS-ConfigureAWSPackage`.
6. Na área Parâmetros, faça o seguinte:
   - Em Action (Ação), selecione Install (Instalar).
   - Para Installation type (Tipo de instalação), escolha Uninstall and reinstall (Desinstalar e reinstalar).

     > **Note**
     >
     > Não há suporte a atualizações no local para este pacote. Ele deve ser desinstalado e reinstalado.

   - Em Nome, digite **`AWSPVDriver`**.

     Não é necessário inserir nada em Versão e Argumentos adicionais.
7. Na seção Targets (Destinos), escolha os nós gerenciados nos quais você quer executar essa operação, especificando as tags, selecionando as instâncias ou dispositivos de borda manualmente ou especificando um grupo de recursos.

   > **Tip**
   >
   > Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

> **Note**
>
> Se você optar por especificar instâncias de destino usando tags e especificar tags que são mapeadas para instâncias do Linux, a associação será bem-sucedida na instância do Windows, mas falhará nas instâncias do Linux. O status geral da associação mostra Failed.

8. Na área Especificar agenda, escolha se deseja executar a associação em uma programação configurada ou apenas uma vez. Os drivers do PV atualizados só são lançados algumas vezes por ano. Por isso, você pode agendar a associação para ser executada uma vez por mês, se desejar.
9. Na área Opções avançadas, em Gravidade da conformidade, escolha um nível de gravidade para a associação. Relatórios de conformidade indicam se o estado é compatível ou não, juntamente com o nível de gravidade que você indicar aqui. Para obter mais informações, consulte Sobre a conformidade de associações do State Manager.
10. Para Rate control (Controle de taxa):
    - Em Concurrency (Concorrência), especifique um número ou uma porcentagem de nós gerenciados nos quais executar o comando ao mesmo tempo.

      > **Note**
      >
      > Se você selecionou destinos especificando tags aplicadas a instâncias a nós gerenciados ou especificando grupos de recursos da AWS, e não tiver certeza de quantas instâncias são direcionadas, restrinja o número de instâncias que poderão executar o documento ao mesmo tempo, especificando uma porcentagem.

    - Em Error threshold (Limite de erro), especifique quando parar de executar o comando em outros nós depois de falhar em alguns ou em uma porcentagem de nós. Por exemplo, se você especificar três erros, o Systems Manager deixará de enviar o comando quando o 4° erro for recebido. Os nós gerenciados que continuam processando o comando também podem enviar erros.
11. (Opcional) Em Opções de saída, para salvar a saída de comando em um arquivo, selecione a caixa Habilitar a gravação da saída no S3. Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

> **Note**
>
> As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil da instância atribuído ao nó gerenciado, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço IAM associada ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

12. (Opcional) Na seção Alarme do CloudWatch, em Nome do alarme, escolha um alarme do CloudWatch existente para aplicar à associação para monitoramento.

> **Note**
>
> Observe as informações a seguir sobre esta etapa.
>
> - A lista de alarmes exibe um máximo de 100 alarmes. Se você não vir o alarme na lista, use a AWS Command Line Interface para criar a associação. Para obter mais informações, consulte Criar uma associação (linha de comando).
> - Para anexar um alarme do CloudWatch ao seu comando, a entidade principal do IAM que cria a associação deve ter permissão para a ação `iam:createServiceLinkedRole`. Para obter mais informações sobre alarmes do CloudWatch, consulte Usar alarmes do Amazon CloudWatch.
> - Se o alarme for ativado, quaisquer invocações ou automações de comando pendentes não serão executadas.

13. Escolha Create Association (Criar associação) e, em seguida, Close (Fechar). O sistema tenta criar a associação nas instâncias e aplicar imediatamente o estado.

    Se você criou a associação em uma ou mais instâncias do Amazon EC2 para Windows Server, o status mudará para Success (Êxito). Se as suas instâncias não estiverem configuradas corretamente para o Systems Manager ou se você tiver inadvertidamente especificado instâncias de destino do Linux, o status mostrará Failed.

Se o status for Failed (Com falha), escolha o ID da associação, escolha a guia Resources (Recursos) e verifique se a associação foi criada com êxito em suas instâncias do EC2 para o Windows Server. Se as instâncias do EC2 para o Windows Server mostrarem o status Failed (Com falha), verifique se o SSM Agent está sendo executado na instância e se a instância está configurada com uma função do AWS Identity and Access Management (IAM) para o Systems Manager. Para obter mais informações, consulte Configurar o console unificado do Systems Manager para uma organização.

Ferramentas de gerenciamento de alterações do AWS Systems Manager

O AWS Systems Manager fornece as ferramentas a seguir para fazer alterações em seus recursos da AWS.

Tópicos

- AWS Systems Manager Automation
- AWS Systems Manager Change Calendar
- AWS Systems Manager Change Manager
- Documentos do AWS Systems Manager
- AWS Systems Manager Maintenance Windows
- AWS Systems Manager Quick Setup

AWS Systems Manager Automation

O Automation, uma ferramenta do AWS Systems Manager, simplifica as tarefas comuns de manutenção, implantação e correção para Serviços da AWS como o Amazon Elastic Compute Cloud (Amazon EC2), o Amazon Relational Database Service (Amazon RDS), o Amazon Redshift, o Amazon Simple Storage Service (Amazon S3) e muitos outros. Para começar a usar o Automation, abra o Systems Manager console (Console do gerenciador de sistemas). No painel de navegação à esquerda, escolha Automation.

O Automation ajuda você a criar soluções automatizadas para implantar, configurar e gerenciar recursos da AWS em escala. Com o Automation, você tem controle granular sobre a simultaneidade

de suas automações. Isso significa que você pode especificar quantos recursos a destinar simultaneamente e quantos erros podem ocorrer antes que uma automação seja interrompida.

Para ajudar você a começar a usar o Automation, a AWS desenvolve e mantém vários runbooks predefinidos. Dependendo do seu caso de uso, você pode usar esses runbooks predefinidos que executam uma variedade de tarefas ou criar seus próprios runbooks personalizados que possam atender melhor às suas necessidades. Para monitorar o progresso e o status de suas automações, você pode usar o console do Automation do Systems Manager ou sua ferramenta da linha de comando preferida. O Automation também se integra ao Amazon EventBridge para ajudar você a criar arquitetura orientada a eventos em escala.

Como o Automation beneficia minha organização?

O Automation oferece os seguintes benefícios:

- Suporte a scripts no conteúdo do runbook

  Com o uso da ação `aws:executeScript`, você pode executar funções de Python e PowerShell personalizadas diretamente de seus runbooks. Isso oferece maior flexibilidade na criação de runbooks personalizados, pois você pode concluir várias tarefas que outras ações do Automation não suportam. Você também tem maior controle sobre a lógica do runbook. Para obter um exemplo de como essa ação pode ser usada e como ela pode ajudar a melhorar uma solução automatizada existente, consulte Criar runbooks do Automation.
- Execute automações em várias Contas da AWS e Regiões da AWS a partir de um local centralizado

  Os administradores podem executar automações em recursos em várias contas e regiões a partir do console do Systems Manager.
- Melhoria das operações de segurança

  Os administradores têm um lugar centralizado para conceder e revogar o acesso a runbooks. Usando somente políticas do AWS Identity and Access Management (IAM), você pode controlar quais usuários individuais ou grupos na sua organização podem usar o Automation e quais nós gerenciados eles podem acessar.
- Automatize tarefas comuns de TI

  A automatização de tarefas comuns pode ajudar a melhorar a eficiência operacional, impor padrões organizacionais e reduzir erros do operador. Por exemplo, você pode usar o runbook `AWS-UpdateCloudFormationStackWithApproval` para atualizar os recursos que foram

implantados usando um modelo AWS CloudFormation. A atualização aplica-se a um novo modelo. Você pode configurar a Automação para solicitar aprovação de um ou mais usuários do antes de a atualização começar.

- Execução segura e em massa de tarefas problemáticas

  O Automation inclui recursos, como controles de taxa, que permitem controlar a implantação de uma automação em toda a frota, especificando um valor de simultaneidade e um limiar de erro. Para obter mais informações sobre como trabalhar com controles de taxas, consulte Executar operações automatizadas em escala.

- Simplificação de tarefas complexas

  O Automation fornece runbooks predefinidos que simplificam tarefas complexas e demoradas, como a criação de Amazon Machine Images de ouro (AMIs). Use os runbooks `AWS-UpdateLinuxAmi` e `AWS-UpdateWindowsAmi` para criar AMIs de ouro a partir de uma AMI de origem. Com o us desses runbooks, você pode executar scripts personalizados antes e depois que as atualizações são aplicadas. Você pode também incluir ou excluir pacotes de software específicos com relação à instalação. Para obter exemplos de como usar esses runbooks, consulte Tutoriais.

- Definição das limitações para entradas

  Você pode definir restrições em runbooks personalizados para limitar os valores que o Automation aceitará para um parâmetro de entrada específico. Por exemplo, `allowedPattern` só aceitará valores para um parâmetro de entrada que corresponda à expressão regular que você definir. Se você especificar `allowedValues` para um parâmetro de entrada, somente os valores especificados no runbook são aceitos.

- Registre a saída das ações de automação no Amazon CloudWatch Logs

  Para satisfazer os requisitos de segurança ou operacionais em sua organização, pode ser necessário fornecer um registro dos scripts executados durante um runbook. Com o CloudWatch Logs, é possível monitorar, armazenar e acessar os arquivos de log de vários Serviços da AWS. Você pode enviar a saída da ação `aws:executeScript` para um grupo de logs do CloudWatch Logs para fins de depuração e solução de problemas. Os dados de log podem ser enviados ao seu grupo de logs com ou sem criptografia do AWS KMS usando sua chave do KMS. Para obter mais informações, consulte Registro de saída de ações do Automation em log com o CloudWatch Logs.

- Integração com o Amazon EventBridge

Há suporte para o Automation como um tipo destino nas regras do Amazon EventBridge. Isso significa que você pode acionar runbooks usando eventos. Para obter mais informações, consulte Monitorar eventos do Systems Manager com o Amazon EventBridge e Referência: padrões e tipos de eventos do Amazon EventBridge para o Systems Manager.

- Compartilhe práticas recomendadas organizacionais

  Você pode definir práticas recomendadas para gerenciamento de recursos, tarefas de operações e muito mais em runbooks compartilhados entre contas e regiões.

Quem deve usar o Automation?

- Qualquer cliente da AWS que deseje melhorar sua eficiência operacional em escala, reduzir erros associados à intervenção manual e reduzir o tempo de resolução de problemas comuns.
- Especialistas em infraestrutura que desejem automatizar tarefas de implantação e configuração.
- Administradores que desejem resolver problemas comuns de forma confiável, melhorar a eficiência da solução de problemas e reduzir operações repetitivas.
- Usuários que desejarem automatizar uma tarefa que normalmente executam manualmente.

O que é uma automação?

Uma automação consiste em todas as tarefas que são definidas em um runbook e são executadas pelo serviço Automation. O Automation usa os seguintes componentes para executar automações.

Conceito	Detalhes
Runbook do Automation	Um runbook do Systems Manager Automation define a automação (as ações que o Systems Manager realiza em nós gerenciados e nos recursos da AWS). O Automation inclui vários runbooks predefinidos que você pode usar para executar tarefas comuns, como reiniciar uma ou mais instâncias do Amazon EC2 ou criar uma Amazon Machine Image (AMI). Você também pode criar seus próprios runbooks. Os runbooks usam YAML ou JSON

Conceito	Detalhes
	e incluem etapas e parâmetros especificados por você. As etapas são executadas em ordem sequencial. Para obter mais informações, consulte Criação dos seus próprios runbooks. Runbooks são documentos do Systems Manager do tipo `Automation` , ao contrário dos documentos `Command`, `Policy` e `Session`. Runbooks oferecem suporte para o esquema versão 0.3. Documentos de comando usam a versão de esquema 1.2, 2.0 ou 2.2. Documentos de política usam a versão de esquema 2.0 ou posterior.
Ação de automação	A automação definida em um runbook inclui uma ou mais etapas. Cada etapa está associada a uma ação específica. A ação determina as entradas, o comportamento e as saídas da etapa. As etapas são definidas na seção `mainSteps` do seu runbook. A automação oferece suporte a 20 tipos de ação distintos. Para mais informações, consulte Referência de ações do Systems Manager Automation.

Conceito	Detalhes
Cota de automação	Cada Conta da AWS pode executar 100 automações simultaneamente. Isso inclui automações filho (automações que são iniciadas por outra automação) e automaçõe s de controle de taxa. Se você tentar executar mais do que isso, o Systems Manager adicionará as outras automações em uma fila e exibirá o status Pending (Pendente). Essa cota pode ser ajustada usando a simultane idade adaptativa. Para obter mais informações, consulte Permitir que o Automation se adapte às suas necessidades de simultaneidade. Para obter mais informações sobre como executar automações, consulte Executar uma operação automatizada com tecnologia do Systems Manager Automation.
Cota de filas da automação	Se você tentar executar mais automações do que o limite de automação simultânea, as automações subsequentes serão adicionad as a uma fila. Cada Conta da AWS pode colocar 5.000 automações na fila. Quando uma automação for concluída (ou atingir um estado terminal), a primeira automação na fila será iniciada.

Conceito	Detalhes
Cota da automação do controle de taxas	Cada Conta da AWS pode executar 25 automações de controle de taxas simultane amente. Se você tentar executar mais automações de controle de taxas do que o limite de automação de controle de taxa simultânea, o Systems Manager adiciona as automações de controle subsequentes a uma fila e exibe o status Pending (Pendente). Para obter mais informações sobre como executar automações de controle de taxas, consulte Executar operações automatizadas em escala.
Cota da fila de automação do controle de taxas	Se você tentar executar mais automações do que o limite de automação de controle de taxa simultânea, as automações subsequen tes serão adicionadas a uma fila. Cada Conta da AWS pode colocar 1.000 automações na fila. Quando uma automação for concluída (ou atingir um estado terminal), a primeira automação na fila será iniciada.

Tópicos

- Configurar a automação
- Executar uma operação automatizada com tecnologia do Systems Manager Automation
- Executar uma automação que exija aprovações
- Executar operações automatizadas em escala
- Executar automações em várias regiões e contas da Regiões da AWS
- Executar automações com base em eventos do EventBridge
- Executar uma automação passo a passo
- Programação de automações com associações do State Manager
- Agendar automações com janelas de manutenção
- Referência de ações do Systems Manager Automation

- Criação dos seus próprios runbooks
- Referência de runbooks do Systems Manager Automation
- Tutoriais
- Saiba mais sobre os status retornados pelo Systems Manager Automation
- Solução de problemas do Systems Manager Automation

Configurar a automação

Para configurar o Automation, uma ferramenta do AWS Systems Manager, é necessário verificar o acesso do usuário ao serviço do Automation e configurar os perfis de forma que o serviço possa executar ações nos recursos. Também recomendamos que você opte pelo modo de simultaneidade adaptativo em suas preferências do Automation. A simultaneidade adaptativa dimensiona automaticamente sua cota de automação para atender às suas necessidades. Para ter mais informações, consulte Permitir que o Automation se adapte às suas necessidades de simultaneidade.

Para garantir o acesso adequado ao Automation AWS Systems Manager, revise os seguintes requisitos de função de serviço e usuário.

Verificar o acesso do usuário aos runbooks

Verifique se você tem permissão para usar runbooks. Se seu usuário, grupo ou perfil tiver permissões de administrador atribuídas, você terá acesso ao Systems Manager Automation. Se você não tiver permissões de administrador, um administrador deverá conceder as permissões ao atribuir a política gerenciada `AmazonSSMFullAccess` ou uma política que forneça permissões comparáveis ao seu usuário, grupo ou perfil.

> **Important**
>
> A política do IAM `AmazonSSMFullAccess` concede permissões para ações do Systems Manager. No entanto, alguns runbooks exigem permissões para outros serviços, como o documento `AWS-ReleaseElasticIP`, que requer permissões do IAM para o `ec2:ReleaseAddress`. Portanto, é necessário analisar as ações executadas em um runbook para garantir que seu usuário, grupo ou perfil tenha as permissões necessárias atribuídas para executar as ações inclusas no runbook.

Configurar o acesso a uma função de serviço (função assumida) para automações

As automações podem ser iniciadas no contexto de uma função de serviço (ou função assumida). Isso permite que o serviço execute ações em seu nome. Se você não especificar uma função a ser assumida, o Automation usará o contexto do usuário que invocou a execução.

No entanto, as seguintes situações ainda exigem que você especifique uma função de serviço para o Automation:

- Quando você quer restringir as permissões de um usuário em um recurso, mas deseja que esse usuário execute uma automação que exija privilégios elevados. Nesse cenário, você poderá criar uma função de serviço com permissões elevadas e permitir que o usuário execute a automação.
- Quando você cria uma associação do Systems Manager State Manager que executa um runbook.
- Ao ter operações que espera executar por mais de 12 horas.
- Quando você estiver executando um runbook que não pertence à Amazon, e que usa a ação `aws:executeScript` para chamar uma operação de API da AWS ou para agir em um recurso da AWS. Para ter mais informações, consulte Permissões para usar runbooks.

Se você precisar criar uma função de serviço para o Automation, poderá usar um dos métodos a seguir.

Tópicos

- Criar perfis de serviço do Automation usando o AWS CloudFormation
- Criar perfis de serviço para o Automation usando o console
- Permitir que o Automation se adapte às suas necessidades de simultaneidade
- Implementação de controles de alteração para o Automation

Criar perfis de serviço do Automation usando o AWS CloudFormation

É possível criar um perfil de serviço para o Automation, uma ferramenta do AWS Systems Manager, com base em um modelo do AWS CloudFormation. Depois de criar a função de serviço, você poderá especificar a função do serviço nos runbooks, usando o parâmetro `AutomationAssumeRole`.

Criar a função de serviço com o AWS CloudFormation

Use o procedimento a seguir para criar as funções do AWS Identity and Access Management (IAM) necessárias para o Systems Manager Automation, usando o AWS CloudFormation.

Para criar as funções do IAM necessárias

1. Baixe e descompacte o arquivo AWS-SystemsManager-AutomationServiceRole.zip. Este arquivo inclui o arquivo de modelo do AWS-SystemsManager-AutomationServiceRole.yaml AWS CloudFormation.
2. Abra o console do AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
3. Selecione Create Stack (Criar pilha).
4. Na seção Specify template (Especificar modelo) escolha Upload a template file (Fazer upload de um arquivo de modelo).
5. Escolha Browse (Procurar) e, depois, escolha o arquivo de modelo AWS-SystemsManager-AutomationServiceRole.yaml do AWS CloudFormation.
6. Escolha Próximo.
7. Na página Specify stack details (Especificar detalhes da tarefa), no campo Stack name (Nome da pilha), insira um nome.
8. Na página Configure stack options (Configurar opções de pilha) não é necessário fazer nenhuma seleção. Escolha Próximo.
9. Na página Review, role para baixo e escolha a opção I acknowledge that AWS CloudFormation might create IAM resources.
10. Escolha Criar.

O CloudFormation mostra o status CREATE_IN_PROGRESS por cerca de três minutos. O status mudará para CREATE_COMPLETE depois que a pilha for criada e suas funções estiverem prontas para uso.

> **Important**
>
> Se você executar um fluxo de trabalho de automação que chama outros serviços usando uma função de serviço do AWS Identity and Access Management (IAM), esteja ciente de que esta função deve ser configurada com permissão para chamar esses serviços. Esse requisito aplica-se a todos os runbooks do Automation da AWS (runbooks da AWS-*), como os runbooks AWS-ConfigureS3BucketLogging, AWS-CreateDynamoDBBackup e AWS-RestartEC2Instance, entre outros. Esse requisito também se aplica a todos os runbooks personalizados do Automation criados que invoquem outros Serviços da AWS, usando ações que chamam outros serviços. Por exemplo, se você usar as ações aws:executeAwsApi, aws:createStack ou aws:copyImage, configure a função de serviço com permissão

para invocar esses serviços. É possível habilitar permissões para outros Serviços da AWS, adicionando uma política em linha do IAM à função. Para obter mais informações, consulte (Opcional) Adicione uma política em linha ou uma política gerenciada pelo cliente para invocar outros Serviços da AWS.

Copiar informações de função para a automação

Use o procedimento a seguir para copiar as informações sobre a função de serviço de Automação por meio do console do AWS CloudFormation. É necessário especificar essas funções ao usar um runbook.

> **Note**
>
> Você não precisará copiar as informações da função usando esse procedimento, se executar os runbooks `AWS-UpdateLinuxAmi` ou `AWS-UpdateWindowsAmi`. Esses runbooks já possuem as funções necessárias especificadas como valores padrão. As funções especificadas nesses runbooks usam as políticas gerenciadas do IAM.

Para copiar nomes de funções

1. Abra o console do AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
2. Selecione o Stack name (Nome da pilha) do Automation que você criou no procedimento anterior.
3. Escolha a guia Resources (Recursos).
4. Escolha o link Physical ID para AutomationServiceRole. O console do IAM é aberto em um resumo da função de serviço do Automation.
5. Copie o nome do recurso da Amazon (ARN) ao lado de Role ARN (ARN da função). O ARN é semelhante ao seguinte: `arn:aws:iam::12345678:role/AutomationServiceRole`
6. Cole o ARN em um arquivo de texto para uso posterior.

Você concluiu a configuração da função de serviço para Automação. Agora você pode usar o ARN da função de serviço do Automation em seus runbooks.

Criar perfis de serviço para o Automation usando o console

Se você precisar criar um perfil de serviço para o Automation, uma ferramenta do AWS Systems Manager, conclua as tarefas a seguir. Para obter mais informações sobre quando uma função de serviço é necessária para o Automation, consulte Configurar a automação.

Tarefas

- Tarefa 1: Criar uma função de serviço para a automação
- Tarefa 2: Anexar a política iam:PassRole à função de automação

Tarefa 1: Criar uma função de serviço para a automação

Use o procedimento a seguir para criar uma função de serviço (ou função assumida) para o Systems Manager Automation.

> **Note**
>
> Você também pode usar essa função em runbooks, como no runbook `AWS-CreateManagedLinuxInstance`. O uso dessa função ou do nome do recurso da Amazon (ARN) de uma função do AWS Identity and Access Management (IAM) em runbooks permite que o Automation realize ações no seu ambiente, como iniciar novas instâncias e executar ações em seu nome.

Para criar uma função do IAM e permitir que ela seja assumida pelo Automation

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Roles e Create role.
3. Em Select type of trusted entity (Selecionar o tipo de entidade confiável), escolha AWS service (serviço).
4. Na seção Selecionar um caso de uso, selecione Systems Manager e Próximo: permissões.
5. Na página Attached permissions policy, procure a política AmazonSSMAutomationRole, escolha essa política e, em seguida, escolha Next: Review.
6. Na página Review (Análise), digite um nome na caixa Role name (Nome da função) e, em seguida, uma descrição.
7. Selecione Create role (Criar função). O sistema faz com que você retorne para a página Roles.

8. Na página Roles (Funções), escolha a função que você acabou de criar para abrir a página Summary (Resumo). Anote os valores de Role Name (Nome da função) e Role ARN (ARN da função). Você especificará o ARN da função ao associar a política iam:PassRole à sua conta do IAM no próximo procedimento. Você também pode especificar o nome da função e o ARN nos runbooks.

> **Note**
>
> A política `AmazonSSMAutomationRole` atribui a permissão de função do Automation a um subconjunto de funções do AWS Lambda na sua conta. Essas funções começam com "Automation". Se você planeja usar o Automation com funções Lambda, o ARN Lambda deverá usar o seguinte formato: `"arn:aws:lambda:*:*:function:Automation*"`
> Se você tiver funções Lambda cujos ARNs não usam esse formato, também precisará anexar uma política Lambda adicional à sua função de automação, como a política AWSLambdaRole. A política ou função adicional deve fornecer acesso mais amplo às funções Lambda na Conta da AWS.

Depois de criar sua função de serviço, recomendamos editar a política de confiança para ajudar a evitar o problema entre serviços confused deputy. O problema confused deputy é um problema de segurança em que uma entidade que não tem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executá-la. Na AWS, a personificação entre serviços pode resultar no problema do 'confused deputy'. A personificação entre serviços pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço chamado). O serviço de chamada pode ser manipulado de modo a usar suas permissões para atuar nos recursos de outro cliente de uma forma na qual ele não deveria ter permissão para acessar. Para evitar isso, o AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com entidades principais de serviço que receberam acesso aos recursos em sua conta.

Recomendamos o uso das chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` em políticas de recursos para limitar as permissões que o Automation concede a outro serviço para o recurso. Se o valor `aws:SourceArn` não contiver o ID da conta, como um ARN de bucket do Amazon S3, você deve usar ambas as chaves de contexto de condição global para limitar as permissões. Se você utilizar ambas as chaves de contexto de condição global e o valor de `aws:SourceArn` contiver o ID da conta, o valor de `aws:SourceAccount` e a conta no valor de `aws:SourceArn` deverão utilizar o mesmo ID de conta quando utilizados na

mesma declaração da política. Use `aws:SourceArn` se quiser apenas um recurso associado a acessibilidade de serviço. Use `aws:SourceAccount` se quiser permitir que qualquer recurso nessa conta seja associado ao uso entre serviços. O valor de `aws:SourceArn` deve ser o ARN para execuções de automação. Se você não souber o ARN completo do recurso ou se estiver especificando vários recursos, use a chave da condição de contexto global `aws:SourceArn` com curingas (*) para as partes desconhecidas do ARN. Por exemplo, `arn:aws:ssm:*:123456789012:automation-execution/*`.

O exemplo a seguir mostra como é possível usar as chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` para o Automation, a fim de evitar o problema confused deputy.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "ssm.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:ssm:*:123456789012:automation-execution/*"
        }
      }
    }
  ]
}
```

Para modificar a política de confiança da função

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.

2. No painel de navegação, selecione Roles (Funções).
3. Na lista de funções em sua conta, escolha o nome da função do serviço do Automation.
4. Escolha a guia Trust relationships (Relacionamentos de confiança) e, em seguida, selecione Edit trust relationship (Editar relacionamento de confiança).
5. Edite a política de confiança usando as chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` para o Automation, a fim de evitar o problema confused deputy.
6. Para salvar a alteração, escolha Update Trust Policy (Atualizar política de confiança).

(Opcional) Adicione uma política em linha ou uma política gerenciada pelo cliente para invocar outros Serviços da AWS

Se você executar uma automação que invoca outros Serviços da AWS usando uma função de serviço do IAM, essa última deverá ser configurada com permissão para invocar esses serviços. Esse requisito aplica-se a todos os runbooks do Automation da AWS (runbooks da `AWS-*`), como os runbooks `AWS-ConfigureS3BucketLogging`, `AWS-CreateDynamoDBBackup` e `AWS-RestartEC2Instance`, entre outros. Esse requisito também se aplica a todos os runbooks personalizados criados que invoquem outros Serviços da AWS, usando ações que chamam outros serviços. Por exemplo, se você usar as ações `aws:executeAwsApi`, `aws:CreateStack` ou `aws:copyImage`, entre outras, deverá configurar a função de serviço com permissão para chamar esses serviços. É possível habilitar permissões para outros Serviços da AWS, adicionando uma política em linha do IAM ou uma política gerenciada pelo cliente ao perfil.

Para incorporar uma política em linha para uma função de serviço (console do IAM)

1. Faça login no AWS Management Console e abra o console do IAM, em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Perfis.
3. Na lista, selecione o nome da função que você deseja editar.
4. Escolha a aba Permissões.
5. Na seção Adicionar permissões, selecione Anexar políticas ou Criar política em linha.
6. Ao escolher Anexar políticas, marque a caixa de seleção ao lado da política que deseja adicionar e escolha Adicionar permissões.
7. Se você escolher Criar política em linha, escolha a guia JSON.
8. Insira um documento de política JSON para os Serviços da AWS que você deseja chamar. Veja a seguir dois exemplos de documentos da política JSON.

Exemplo do PutObject e GetObject do Amazon S3

JSON

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

Exemplos CreateSnapshot e DescribeSnapShots do Amazon EC2

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshot",
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":"ec2:DescribeSnapshots",
         "Resource":"*"
      }
   ]
}
```

Para obter detalhes sobre a linguagem da política do IAM, consulte a Referência da política JSON do IAM no Manual do usuário do IAM.

9. Ao concluir, selecione Review policy (Revisar política). O Validador de política indica se há qualquer erro de sintaxe.
10. Na página Review policy (Revisar política), insira um nome em Name para a política que você está criando. Revise o Resumo da política para ver as permissões que são concedidas pela política. Em seguida, escolha Criar política para salvar seu trabalho.
11. Após a criação de uma política em linha, ela é automaticamente incorporada à sua função.

Tarefa 2: Anexar a política iam:PassRole à função de automação

Use o procedimento a seguir para associar a política `iam:PassRole` à sua função de serviço de Automação. Isso permite que o serviço do Automation passe o perfil a outros serviços ou ferramentas do Systems Manager ao executar as automações.

Para anexar a política iam:PassRole à sua função de Automação

1. Na página Summary da função que você acabou de criar, escolha a guia Permissions (Permissões).
2. Escolha Add inline policy (Adicionar política em linha).
3. Na página Create policy (Criar política), selecione a guia Visual editor (Editor visual).
4. Selecione Service (Serviço) e, em seguida, selecione IAM.
5. Selecione Select actions (Selecionar ações).
6. Na caixa de texto Filter actions (Filtrar ações), digite **`PassRole`** e selecione a opção PassRole.
7. Escolha atributos. Verifique se Specific (Específico) está selecionado e, em seguida, selecione Add ARN (Adicionar ARN).
8. No campo Specify ARN for role (Especificar ARN para função) cole o ARN da função de Automação que você copiou no final da Tarefa 1. O sistema preenche os campos Account (Conta) e Role name with path (Nome da função com caminho).

> **Note**
>
> Se você quiser que a função de serviço do Automation associe uma função de perfil da instância do IAM a uma instância do EC2, você deverá adicionar o ARN da função de

perfil da instância do IAM. Isso permite que a função de serviço do Automation passe a função do perfil da instância do IAM para a instância de destino do EC2.

9. Escolha Add (Adicionar).
10. Escolha Review policy (Revisar política).
11. Na página Review Policy (Revisar política), digite um nome e, em seguida, selecione Create Policy (Criar política).

Permitir que o Automation se adapte às suas necessidades de simultaneidade

Por padrão, o Automation permite que você execute até 100 automações simultâneas por vez. O Automation também fornece uma configuração opcional que você pode usar para ajustar sua cota de automação de simultaneidade automaticamente. Com essa configuração, sua cota de automação de simultaneidade pode acomodar até 500 automações simultâneas, dependendo dos recursos disponíveis.

> **Note**
>
> Se sua automação chamar operações de API, o dimensionamento adaptativo para seus destinos pode resultar em exceções de limitação. Se ocorrerem exceções de limitação recorrentes ao executar automações com simultaneidade adaptativa ativada, talvez seja necessário solicitar aumentos de cotas para a operação da API, se disponível.

Para ativar a simultaneidade adaptativa usando o AWS Management Console

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação à esquerda, escolha Automation (Automação).
3. Escolha a guia Preferences (Preferências) e, em seguida, escolha Edit (Editar).
4. Marque a caixa de seleção ao lado de Enable adaptive concurrency (Ativar simultaneidade adaptativa).
5. Escolha Salvar.

Para ativar a simultaneidade adaptativa usando a linha de comando

- Abra a AWS CLI ou as ferramentas para Windows PowerShell e execute o comando a seguir para ativar a simultaneidade adaptativa para sua conta na região solicitante.

  Linux & macOS

  ```
  aws ssm update-service-setting \
      --setting-id /ssm/automation/enable-adaptive-concurrency \
      --setting-value True
  ```

  Windows

  ```
  aws ssm update-service-setting ^
      --setting-id /ssm/automation/enable-adaptive-concurrency ^
      --setting-value True
  ```

  PowerShell

  ```
  Update-SSMServiceSetting `
      -SettingId "/ssm/automation/enable-adaptive-concurrency" `
      -SettingValue "True"
  ```

Implementação de controles de alteração para o Automation

Por padrão, o Automation permite que você use runbooks sem restrições de data e hora. Ao integrar o Automation com o Change Calendar, é possível implementar controles de alteração para todas as automações em sua Conta da AWS. Com essa configuração, as entidades principais do AWS Identity and Access Management (IAM) em sua conta poderão somente executar automações durante os períodos permitidos pelo calendário de alterações. Para saber mais sobre como trabalhar com o Change Calendar, consulte Trabalhar com o Change Calendar.

Como ativar os controles de alteração (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação à esquerda, escolha Automation (Automação).
3. Escolha a guia Preferences (Preferências) e, em seguida, escolha Edit (Editar).
4. Marque a caixa de seleção ao lado de Ativar a integração do Change Calendar.

5. Na lista suspensa Escolher um calendário de alterações, escolha o calendário de alterações que você deseja que o Automation siga.
6. Escolha Salvar.

Executar uma operação automatizada com tecnologia do Systems Manager Automation

Ao executar uma automação, por padrão, a automação será executada no contexto do usuário que iniciou a automação. Isso significa, por exemplo, que se o usuário tiver permissões de administrador, a automação será executada com permissões de administrador e acesso total aos recursos que estão sendo configurados pela automação. Como uma prática recomendada de segurança, sugerimos que você execute a automação usando um perfil de serviço do IAM, nesse caso, conhecido como o perfil assumir, que é configurado com a política gerenciada AmazonSSMAutomationRole. Talvez seja necessário adicionar outras políticas do IAM ao seu perfil assumir para usar vários runbooks. O uso de uma função de serviço do IAM para executar a automação é chamado de administração delegada.

Quando você usa uma função de serviço, a automação tem permissão para ser executada nos recursos da AWS, mas o usuário que executou a automação tem acesso restrito (ou nenhum acesso) a esses recursos. Por exemplo, é possível configurar uma função de serviço e usá-la com o Automation para reiniciar uma ou mais instâncias do Amazon Elastic Compute Cloud (Amazon EC2). O Automation é uma ferramenta do AWS Systems Manager. A automação reinicia as instâncias, mas a função do serviço não concede permissão ao usuário para acessar essas instâncias.

Você pode especificar uma função de serviço em runtime ao executar uma automação ou você pode criar runbooks personalizados e especificar a função de serviço diretamente no runbook. Se você especificar uma função de serviço, seja em runtime ou em um runbook, o serviço será executado no contexto da função de serviço especificada. Se você não especificar uma função de serviço, o sistema criará uma sessão temporária no contexto do usuário e executará a automação.

> **Note**
>
> Especifique uma função de serviço para as automações que você espera que sejam executadas por mais de 12 horas. Se você iniciar uma automação de execução prolongada no contexto de um usuário, a sessão temporária do usuário expirará após 12 horas.

A administração delegada garante segurança e controle elevados de seus recursos da AWS. Isso também permite uma experiência de auditoria aprimorada, pois as ações estão sendo executadas nos recursos por uma função de serviço central em vez de várias contas do IAM.

Antes de começar

Antes de concluir o procedimento a seguir, é necessário criar o perfil de serviço do IAM e configurar uma relação de confiança para o Automation, uma ferramenta do AWS Systems Manager. Para obter mais informações, consulte Tarefa 1: Criar uma função de serviço para a automação.

Os procedimentos a seguir descrevem como usar o console do Systems Manager ou a ferramenta de linha de comando de sua preferência para executar uma automação simples.

Executar uma automação simples (console)

O procedimento a seguir descreve como usar o console do Systems Manager para executar uma automação simples.

Para executar uma automação simples

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, selecione Automation e Execute automation (Executar automação).
3. Na lista Automation document (Documento do Automation), escolha um runbook. Escolha uma ou mais opções no painel Document categories (Categorias de documentos) para filtrar documentos SSM de acordo com sua finalidade. Para visualizar um runbook que você tenha, escolha a guia Owned by me (De minha propriedade). Para visualizar um runbook compartilhado com sua conta, escolha a guia Shared with me (Compartilhado comigo). Para visualizar todos os runbooks, escolha a guia All documents (Todos os documentos).

   > **Note**
   >
   > Você pode visualizar informações sobre um runbook, selecionando o nome dele.

4. Na seção Document details (Detalhes do documento), verifique se Document version (Versão do documento) está definida como a versão que você quer executar. O sistema inclui as seguintes opções de versão:

   - Versão padrão no runtime: escolha essa opção se o runbook do Automation for atualizado periodicamente e uma nova versão padrão for atribuída.

- Versão mais recente no runtime: escolha essa opção se o runbook do Automation for atualizado periodicamente e se você quiser executar a versão mais recente.
- 1 (padrão): escolha esta opção para executar a primeira versão do documento, que é a versão padrão.

5. Escolha Próximo.
6. Na seção Execution mode (Modo de execução), escolha Simple execution (Execução simples).
7. Na seção Input parameters (Parâmetros de entrada), especifique as entradas necessárias. Opcionalmente, você pode escolher uma função de serviço do IAM na lista AutomationAssumeRole.
8. (Opcional) Escolha um alarme do CloudWatch a fim de aplicar à sua automação para monitoramento. Para anexar um alarme do CloudWatch à sua automação, a entidade principal do IAM que inicia a automação deve ter permissão para a ação `iam:createServiceLinkedRole`. Para obter mais informações sobre alarmes do CloudWatch, consulte Usar alarmes do Amazon CloudWatch. Observe que a automação será interrompida se o alarme for ativado. Se você usar o AWS CloudTrail, você verá a chamada de API em sua trilha.
9. Clique em Executar.

O console exibe o status da automação. Se houver falha na execução da automação, consulte Solução de problemas do Systems Manager Automation.

Executar uma automação simples (linha de comando)

O procedimento a seguir descreve como usar a AWS CLI (no Linux ou no Windows) ou o Ferramentas da AWS para PowerShell para executar uma automação simples.

Para executar uma automação simples

1. Instale e configure a AWS CLI ou o Ferramentas da AWS para PowerShell, caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI e Instalar o Ferramentas da AWS para PowerShell.
2. Execute o comando a seguir para iniciar uma automação simples. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

Linux & macOS

```
aws ssm start-automation-execution \
    --document-name runbook name \
    --parameters runbook parameters
```

Windows

```
aws ssm start-automation-execution ^
    --document-name runbook name ^
    --parameters runbook parameters
```

PowerShell

```
Start-SSMAutomationExecution `
  -DocumentName runbook name `
  -Parameter runbook parameters
```

Este é um exemplo que usa o runbook `AWS-RestartEC2Instance` para reiniciar a instância do EC2 especificada.

Linux & macOS

```
aws ssm start-automation-execution \
    --document-name "AWS-RestartEC2Instance" \
    --parameters "InstanceId=i-02573cafcfEXAMPLE"
```

Windows

```
aws ssm start-automation-execution ^
    --document-name "AWS-RestartEC2Instance" ^
    --parameters "InstanceId=i-02573cafcfEXAMPLE"
```

PowerShell

```
Start-SSMAutomationExecution `
  -DocumentName AWS-RestartEC2Instance `
  -Parameter @{"InstanceId"="i-02573cafcfEXAMPLE"}
```

O sistema retorna informações como estas.

Linux & macOS

```
{
    "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0123456789ab"
}
```

Windows

```
{
    "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0123456789ab"
}
```

PowerShell

```
4105a4fc-f944-11e6-9d32-0123456789ab
```

3. Execute o comando a seguir para recuperar o status da automação.

Linux & macOS

```
aws ssm describe-automation-executions \
    --filter "Key=ExecutionId,Values=4105a4fc-f944-11e6-9d32-0123456789ab"
```

Windows

```
aws ssm describe-automation-executions ^
    --filter "Key=ExecutionId,Values=4105a4fc-f944-11e6-9d32-0123456789ab"
```

PowerShell

```
Get-SSMAutomationExecutionList | `
  Where {$_.AutomationExecutionId -eq "4105a4fc-f944-11e6-9d32-0123456789ab"}
```

O sistema retorna informações como estas.

Linux & macOS

```
{
    "AutomationExecutionMetadataList": [
        {
            "AutomationExecutionStatus": "InProgress",
            "CurrentStepName": "stopInstances",
            "Outputs": {},
            "DocumentName": "AWS-RestartEC2Instance",
            "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0123456789ab",
            "DocumentVersion": "1",
            "ResolvedTargets": {
                "ParameterValues": [],
                "Truncated": false
            },
            "AutomationType": "Local",
            "Mode": "Auto",
            "ExecutionStartTime": 1564600648.159,
            "CurrentAction": "aws:changeInstanceState",
            "ExecutedBy": "arn:aws:sts::123456789012:assumed-role/Administrator/
Admin",
            "LogFile": "",
            "Targets": []
        }
    ]
}
```

Windows

```
{
    "AutomationExecutionMetadataList": [
        {
            "AutomationExecutionStatus": "InProgress",
            "CurrentStepName": "stopInstances",
            "Outputs": {},
            "DocumentName": "AWS-RestartEC2Instance",
            "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0123456789ab",
            "DocumentVersion": "1",
            "ResolvedTargets": {
                "ParameterValues": [],
                "Truncated": false
            },
```

```
            "AutomationType": "Local",
            "Mode": "Auto",
            "ExecutionStartTime": 1564600648.159,
            "CurrentAction": "aws:changeInstanceState",
            "ExecutedBy": "arn:aws:sts::123456789012:assumed-role/Administrator/
Admin",
            "LogFile": "",
            "Targets": []
        }
    ]
}
```

PowerShell

```
AutomationExecutionId       : 4105a4fc-f944-11e6-9d32-0123456789ab
AutomationExecutionStatus   : InProgress
AutomationType              : Local
CurrentAction               : aws:changeInstanceState
CurrentStepName             : startInstances
DocumentName                : AWS-RestartEC2Instance
DocumentVersion             : 1
ExecutedBy                  : arn:aws:sts::123456789012:assumed-role/
Administrator/Admin
ExecutionEndTime            : 1/1/0001 12:00:00 AM
ExecutionStartTime          : 7/31/2019 7:17:28 PM
FailureMessage              :
LogFile                     :
MaxConcurrency              :
MaxErrors                   :
Mode                        : Auto
Outputs                     : {}
ParentAutomationExecutionId :
ResolvedTargets             :
 Amazon.SimpleSystemsManagement.Model.ResolvedTargets
Target                      :
TargetMaps                  : {}
TargetParameterName         :
Targets                     : {}
```

Executar uma automação que exija aprovações

Os procedimentos a seguir descrevem como usar o console do AWS Systems Manager e a AWS Command Line Interface (AWS CLI) para executar uma automação com aprovações usando uma execução simples. A automação usa a ação `aws:approve`, que pausa temporariamente a automação até que as entidades principais designadas aprovem ou neguem a ação. A automação é executada no contexto do usuário atual. Isso significa que você não precisa configurar outras permissões do IAM, desde que tenha permissão para usar o runbook e qualquer ação chamada pelo runbook. Se você tiver permissões de administrador no IAM, isso significa que você já tem permissão para usar esse runbook.

Antes de começar

Além das entradas padrão exigidas pelo runbook, a ação `aws:approve` requer os dois parâmetros a seguir:

- Uma lista de aprovadores. A lista de aprovadores deve conter, no mínimo, um aprovador na forma de um nome de usuário ou ARN de usuário. Se vários aprovadores forem fornecidos, uma contagem de aprovações mínimas correspondente também deverá ser especificada no runbook.
- ARN do tópico de um Amazon Simple Notification Service (Amazon SNS) O nome do tópico do Amazon SNS deve começar com `Automation`.

Esse procedimento pressupõe que você já tenha criado um tópico do Amazon SNS, que é necessário para fornecer a solicitação de aprovação. Para obter informações, consulte Criar um tópico no Manual do desenvolvedor do Amazon Simple Notification Service.

Executar uma automação com aprovadores (console)

Para executar uma automação com aprovadores

O procedimento a seguir descreve como usar o console do Systems Manager para executar uma automação com aprovadores.

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, selecione Automation e Execute automation (Executar automação).
3. Na lista Automation document (Documento do Automation), escolha um runbook. Escolha uma ou mais opções no painel Document categories (Categorias de documentos) para filtrar documentos SSM de acordo com sua finalidade. Para visualizar um runbook que você tenha,

escolha a guia Owned by me (De minha propriedade). Para visualizar um runbook compartilhado com sua conta, escolha a guia Shared with me (Compartilhado comigo). Para visualizar todos os runbooks, escolha a guia All documents (Todos os documentos).

> **Note**
>
> Você pode visualizar informações sobre um runbook, selecionando o nome dele.

4. Na seção Document details (Detalhes do documento), verifique se Document version (Versão do documento) está definida como a versão que você quer executar. O sistema inclui as seguintes opções de versão:

   - Versão padrão no runtime: escolha essa opção se o runbook do Automation for atualizado periodicamente e uma nova versão padrão for atribuída.
   - Versão mais recente no runtime: escolha essa opção se o runbook do Automation for atualizado periodicamente e se você quiser executar a versão mais recente.
   - 1 (padrão): escolha esta opção para executar a primeira versão do documento, que é a versão padrão.

5. Escolha Próximo.
6. Na página Execute automation document (Executar documento de automação), escolha Simple execution (Execução simples).
7. Na seção Input parameters (Parâmetros de entrada), especifique os parâmetros de entrada necessários.

   Por exemplo, se você escolheu o runbook **`AWS-StartEC2InstanceWithApproval`**, deverá especificar ou escolher IDs de instância para o parâmetro InstanceId.
8. Na seção Aprovadores, especifique os nomes de usuário ou os ARNs de usuário dos aprovadores para a ação de automação.
9. Na seção SNSTopicARN, especifique o ARN do tópico do SNS a ser usado para enviar notificações de aprovação. O nome do tópico do SNS deve começar com Automation (Automação).
10. Opcionalmente, você pode escolher uma função de serviço do IAM na lista AutomationAssumeRole. Se seu destino tem mais de 100 contas e regiões, é necessário especificar o `AWS-SystemsManager-AutomationAdministrationRole`.
11. Escolha Execute automation.

O aprovador especificado recebe uma notificação do Amazon SNS com detalhes para aprovar ou rejeitar a automação. Essa ação de aprovação é válida por 7 dias a contar da data de emissão e pode ser emitida usando o console do Systems Manager ou a AWS Command Line Interface (AWS CLI).

Se você optou por aprovar a automação, ela continuará a executar as etapas incluídas no runbook especificado. O console exibe o status da automação. Se houver falha na execução da automação, consulte Solução de problemas do Systems Manager Automation.

Para aprovar ou negar uma automação

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Automation (Automação) e selecione a automação que foi executada no procedimento anterior.
3. Escolha Actions (Ações) e depois Approve/Deny (Aprovar/negar).
4. Escolha entre Approve (Aprovar) ou Deny (Negar) e, opcionalmente, forneça um comentário.
5. Selecione Enviar.

Executar uma automação com aprovadores (linha de comando)

O procedimento a seguir descreve como usar a AWS CLI (no Linux ou no Windows) ou o Ferramentas da AWS para PowerShell para executar uma automação com aprovadores.

Para executar uma automação com aprovadores

1. Instale e configure a AWS CLI ou o Ferramentas da AWS para PowerShell, caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI e Instalar o Ferramentas da AWS para PowerShell.
2. Execute o comando a seguir para executar uma automação com aprovadores. Substitua cada *`espaço reservado para recurso de exemplo`* por suas próprias informações. Na seção de nome do documento, especifique um runbook que inclua a ação de automação `aws:approve`.

   Em `Approvers`, especifique os nomes de usuário ou os ARNs de usuário dos aprovadores para a ação. Em `SNSTopic`, especifique o ARN do tópico do SNS a ser usado para enviar a notificação de aprovação. O nome do tópico do Amazon SNS deve começar com `Automation`.

> **Note**
>
> Os nomes específicos dos valores dos parâmetros para aprovadores e o tópico do SNS dependem dos valores especificados no runbook escolhido.

Linux & macOS

```
aws ssm start-automation-execution \
    --document-name "AWS-StartEC2InstanceWithApproval" \
    --parameters
 "InstanceId=i-02573cafcfEXAMPLE,Approvers=arn:aws:iam::123456789012:role/
Administrator,SNSTopicArn=arn:aws:sns:region:123456789012:AutomationApproval"
```

Windows

```
aws ssm start-automation-execution ^
    --document-name "AWS-StartEC2InstanceWithApproval" ^
    --parameters
 "InstanceId=i-02573cafcfEXAMPLE,Approvers=arn:aws:iam::123456789012:role/
Administrator,SNSTopicArn=arn:aws:sns:region:123456789012:AutomationApproval"
```

PowerShell

```
Start-SSMAutomationExecution `
    -DocumentName AWS-StartEC2InstanceWithApproval `
    -Parameters @{
        "InstanceId"="i-02573cafcfEXAMPLE"
        "Approvers"="arn:aws:iam::123456789012:role/Administrator"
        "SNSTopicArn"="arn:aws:sns:region:123456789012:AutomationApproval"
    }
```

O sistema retorna informações como estas.

Linux & macOS

```
{
    "AutomationExecutionId": "df325c6d-b1b1-4aa0-8003-6cb7338213c6"
```

```
}
```

Windows

```
{
    "AutomationExecutionId": "df325c6d-b1b1-4aa0-8003-6cb7338213c6"
}
```

PowerShell

```
df325c6d-b1b1-4aa0-8003-6cb7338213c6
```

Para aprovar uma automação

- Execute o comando a seguir para aprovar uma automação. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

  Linux & macOS

  ```
  aws ssm send-automation-signal \
      --automation-execution-id "df325c6d-b1b1-4aa0-8003-6cb7338213c6" \
      --signal-type "Approve" \
      --payload "Comment=your comments"
  ```

  Windows

  ```
  aws ssm send-automation-signal ^
      --automation-execution-id "df325c6d-b1b1-4aa0-8003-6cb7338213c6" ^
      --signal-type "Approve" ^
      --payload "Comment=your comments"
  ```

  PowerShell

  ```
  Send-SSMAutomationSignal `
      -AutomationExecutionId df325c6d-b1b1-4aa0-8003-6cb7338213c6 `
      -SignalType Approve `
      -Payload @{"Comment"="your comments"}
  ```

Não haverá saída se o comando for bem-sucedido.

Para negar uma automação

- Execute o comando a seguir para negar uma automação. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

  Linux & macOS

  ```
  aws ssm send-automation-signal \
      --automation-execution-id "df325c6d-b1b1-4aa0-8003-6cb7338213c6" \
      --signal-type "Deny" \
      --payload "Comment=your comments"
  ```

  Windows

  ```
  aws ssm send-automation-signal ^
      --automation-execution-id "df325c6d-b1b1-4aa0-8003-6cb7338213c6" ^
      --signal-type "Deny" ^
      --payload "Comment=your comments"
  ```

  PowerShell

  ```
  Send-SSMAutomationSignal `
      -AutomationExecutionId df325c6d-b1b1-4aa0-8003-6cb7338213c6 `
      -SignalType Deny `
      -Payload @{"Comment"="your comments"}
  ```

  Não haverá saída se o comando for bem-sucedido.

Executar operações automatizadas em escala

Com o AWS Systems Manager Automation, você pode executar automações em uma frota de recursos da AWS usando destinos. Além disso, você pode controlar a implantação da automação em toda a frota, especificando um valor de simultaneidade e um erro limite. Os recursos de simultaneidade e limite de erros são coletivamente chamados de controles de taxa. O valor de simultaneidade determina quantos recursos são permitidos para executar a automação

simultaneamente. O Automation também fornece um modo de simultaneidade adaptativa que você pode optar por participar. A simultaneidade adaptativa dimensiona automaticamente sua cota de automação de 100 automações em execução simultânea até 500. Um limite de erro determina quantas automações podem falhar antes do Systems Manager parar de enviar a automação para outros recursos.

Para obter mais informações sobre simultaneidade e limites de erro, consulte Automações de controle em grande escala. Para obter mais informações sobre destinos, consulte Mapeamento de destino para uma automação.

Os procedimentos a seguir descrevem como ativar a simultaneidade adaptativa e como executar uma automação com destinos e controles de taxa usando o console do Systems Manager e a AWS Command Line Interface (AWS CLI).

Executar uma automação com destinos e controles de taxa (console)

O procedimento a seguir descreve como usar o console do Systems Manager para executar uma automação com destinos e controles de taxa.

Para executar uma automação com destinos e controles de taxa

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, selecione Automation e Execute automation (Executar automação).
3. Na lista Automation document (Documento do Automation), escolha um runbook. Escolha uma ou mais opções no painel Document categories (Categorias de documentos) para filtrar documentos SSM de acordo com sua finalidade. Para visualizar um runbook que você tenha, escolha a guia Owned by me (De minha propriedade). Para visualizar um runbook compartilhado com sua conta, escolha a guia Shared with me (Compartilhado comigo). Para visualizar todos os runbooks, escolha a guia All documents (Todos os documentos).

   > **Note**
   >
   > Você pode visualizar informações sobre um runbook, selecionando o nome dele.

4. Na seção Document details (Detalhes do documento), verifique se Document version (Versão do documento) está definida como a versão que você quer executar. O sistema inclui as seguintes opções de versão:

- Versão padrão no runtime: escolha essa opção se o runbook do Automation for atualizado periodicamente e uma nova versão padrão for atribuída.
- Versão mais recente no runtime: escolha essa opção se o runbook do Automation for atualizado periodicamente e se você quiser executar a versão mais recente.
- 1 (padrão): escolha esta opção para executar a primeira versão do documento, que é a versão padrão.

5. Escolha Próximo.
6. Na seção Execution Mode (Modo de execução), escolha Rate Control (Controle de taxa). Você deve usar esse modo ou Multi-account and Region (Várias contas e regiões) se quiser usar destinos e controles de taxa.
7. Na seção Targets (Destinos), escolha como você quer definir os destinos dos recursos da AWS em que você deseja executar o Automation. Essas opções são obrigatórias.

   a. Use a lista Parameter (Parâmetro) para escolher um parâmetro. Os itens na lista Parameter (Parâmetro) são determinados pelos parâmetros no runbook do Automation que você selecionou no início deste procedimento. Ao escolher um parâmetro, você define o tipo de recurso em que o fluxo de trabalho da Automação é executado.

   b. Use a lista Targets (Destinos) para escolher como você deseja definir o destino dos recursos.

      i. Se você optar por definir o destino dos recursos usando valores de parâmetro, insira o valor do parâmetro para aquele escolhido na seção Input parameters (Parâmetros de entrada).

      ii. Se você escolher definir o destino dos recursos usando o AWS Resource Groups, escolha o nome do grupo na lista Resource Group (Grupo de recursos).

      iii. Se você optar por definir o destino dos recursos usando tags, insira a chave de tag e (opcionalmente) o valor da tag nos campos fornecidos. Escolha Adicionar.

      iv. Se você quiser executar um runbook do Automation em todas as instâncias nas Conta da AWS e Região da AWS atuais, escolha All instances (Todas as instâncias).

8. Na seção Input parameters (Parâmetros de entrada), especifique as entradas necessárias. Opcionalmente, você pode escolher uma função de serviço do IAM na lista AutomationAssumeRole.

> **Note**
>
> Pode ser que não seja necessário escolher algumas das opções na seção Input parameters (Parâmetros de entrada). Isso ocorre porque você definiu o destino dos recursos usando tags ou um grupo de recursos. Por exemplo, se você escolheu o runbook `AWS-RestartEC2Instance`, você não precisará especificar ou escolher IDs de instância na seção Input parameters (Parâmetros de entrada). A execução do Automation localiza as instâncias para reiniciar usando as tags ou grupos de recursos que você especificou.

9. Use as opções na seção Rate control (Controle de taxa) para restringir o número de recursos da AWS que podem executar o Automation dentro de cada par conta-região.

   Na seção Concurrency (Simultaneidade), escolha uma opção:

   - Escolha Targets (Destinos) para inserir um número absoluto de destinos que podem executar o fluxo de trabalho de Automação simultaneamente.
   - Escolha Percentage (Porcentagem) para inserir uma porcentagem do conjunto de destino que pode executar o fluxo de trabalho de Automação simultaneamente.

10. Na seção Error threshold (Limite de erro), escolha uma opção:

    - Escolha errors (erros) para inserir um número absoluto de erros permitidos antes que a Automação deixe de enviar o fluxo de trabalho para outros recursos.
    - Escolha percentage (porcentagem) para inserir uma porcentagem de erros permitidos antes que a Automação deixe de enviar o fluxo de trabalho para outros recursos.

11. (Opcional) Escolha um alarme do CloudWatch a fim de aplicar à sua automação para monitoramento. Para anexar um alarme do CloudWatch à sua automação, a entidade principal do IAM que inicia a automação deve ter permissão para a ação `iam:createServiceLinkedRole`. Para obter mais informações sobre alarmes do CloudWatch, consulte Usar alarmes do Amazon CloudWatch. Observe que a automação será interrompida se o alarme for ativado. Se você usar o AWS CloudTrail, você verá a chamada de API em sua trilha.

12. Clique em Executar.

Para visualizar automações iniciadas pela automação do controle de taxa, no painel de navegação, escolha Automation (Automação) e depois selecione Show child automations (Mostrar automações filho).

Executar uma automação com destinos e controles de taxa (linha de comando)

O procedimento a seguir descreve como usar a AWS CLI (no Linux ou no Windows) ou o Ferramentas da AWS para PowerShell para executar uma automação com destinos e controles de taxa.

Para executar uma automação com destinos e controles de taxa

1. Instale e configure a AWS CLI ou o Ferramentas da AWS para PowerShell, caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI e Instalar o Ferramentas da AWS para PowerShell.
2. Execute o comando a seguir para visualizar uma lista de documentos.

   Linux & macOS

   ```
   aws ssm list-documents
   ```

   Windows

   ```
   aws ssm list-documents
   ```

   PowerShell

   ```
   Get-SSMDocumentList
   ```

   Anote o nome do runbook que você quer usar.
3. Execute o comando a seguir para visualizar os detalhes sobre o runbook. Substitua o *runbook name* (nome do runbook) pelo nome do runbook cujos detalhes deseja visualizar. Além disso, anote o nome de um parâmetro (por exemplo, `InstanceId`) que deseja usar para a opção `--target-parameter-name`. Esse parâmetro determina o tipo de recurso em que a automação é executada.

Linux & macOS

```
aws ssm describe-document \
    --name runbook name
```

Windows

```
aws ssm describe-document ^
    --name runbook name
```

PowerShell

```
Get-SSMDocumentDescription `
    -Name runbook name
```

4. Crie um comando que use as opções de destino e controle de taxa que você deseja executar. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   Definir destino usando tags

   Linux & macOS

```
aws ssm start-automation-execution \
    --document-name runbook name \
    --targets Key=tag:key name,Values=value \
    --target-parameter-name parameter name \
    --parameters "input parameter name=input parameter value,input parameter 2
 name=input parameter 2 value" \
    --max-concurrency 10 \
    --max-errors 25%
```

   Windows

```
aws ssm start-automation-execution ^
    --document-name runbook name ^
    --targets Key=tag:key name,Values=value ^
    --target-parameter-name parameter name ^
    --parameters "input parameter name=input parameter value,input parameter 2
 name=input parameter 2 value" ^
```

```
    --max-concurrency 10 ^
    --max-errors 25%
```

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
$Targets.Key = "tag:key name"
$Targets.Values = "value"

Start-SSMAutomationExecution `
    DocumentName "runbook name" `
    -Targets $Targets `
    -TargetParameterName "parameter name" `
    -Parameter @{"input parameter name"="input parameter value";"input parameter
 2 name"="input parameter 2 value"} `
    -MaxConcurrency "10" `
    -MaxError "25%"
```

Definir destino usando valores de parâmetro

Linux & macOS

```
aws ssm start-automation-execution \
    --document-name runbook name \
    --targets Key=ParameterValues,Values=value,value 2,value 3 \
    --target-parameter-name parameter name \
    --parameters "input parameter name=input parameter value" \
    --max-concurrency 10 \
    --max-errors 25%
```

Windows

```
aws ssm start-automation-execution ^
    --document-name runbook name ^
    --targets Key=ParameterValues,Values=value,value 2,value 3 ^
    --target-parameter-name parameter name ^
    --parameters "input parameter name=input parameter value" ^
    --max-concurrency 10 ^
    --max-errors 25%
```

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
$Targets.Key = "ParameterValues"
$Targets.Values = "value","value 2","value 3"

Start-SSMAutomationExecution `
    -DocumentName "runbook name" `
    -Targets $Targets `
    -TargetParameterName "parameter name" `
    -Parameter @{"input parameter name"="input parameter value"} `
    -MaxConcurrency "10" `
    -MaxError "25%"
```

Definir destino usando AWS Resource Groups

Linux & macOS

```
aws ssm start-automation-execution \
    --document-name runbook name \
    --targets Key=ResourceGroup,Values=Resource group nname \
    --target-parameter-name parameter name \
    --parameters "input parameter name=input parameter value" \
    --max-concurrency 10 \
    --max-errors 25%
```

Windows

```
aws ssm start-automation-execution ^
    --document-name runbook name ^
    --targets Key=ResourceGroup,Values=Resource group name ^
    --target-parameter-name parameter name ^
    --parameters "input parameter name=input parameter value" ^
    --max-concurrency 10 ^
    --max-errors 25%
```

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
$Targets.Key = "ResourceGroup"
```

```
$Targets.Values = "Resource group name"

Start-SSMAutomationExecution `
    -DocumentName "runbook name" `
    -Targets $Targets `
    -TargetParameterName "parameter name" `
    -Parameter @{"input parameter name"="input parameter value"} `
    -MaxConcurrency "10" `
    -MaxError "25%"
```

Direcionar todas as instâncias do Amazon EC2 na Conta da AWS e na Região da AWS atuais

Linux & macOS

```
aws ssm start-automation-execution \
    --document-name runbook name \
    --targets "Key=AWS::EC2::Instance,Values=*"  \
    --target-parameter-name instanceId \
    --parameters "input parameter name=input parameter value" \
    --max-concurrency 10 \
    --max-errors 25%
```

Windows

```
aws ssm start-automation-execution ^
    --document-name runbook name ^
    --targets Key=AWS::EC2::Instance,Values=* ^
    --target-parameter-name instanceId ^
    --parameters "input parameter name=input parameter value" ^
    --max-concurrency 10 ^
    --max-errors 25%
```

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
$Targets.Key = "AWS::EC2::Instance"
$Targets.Values = "*"

Start-SSMAutomationExecution `
    -DocumentName "runbook name" `
    -Targets $Targets `
```

```
    -TargetParameterName "instanceId" `
    -Parameter @{"input parameter name"="input parameter value"} `
    -MaxConcurrency "10" `
    -MaxError "25%"
```

O comando retorna um ID de execução. Copie esse ID para a área de transferência. Você pode usar esse ID para visualizar o status da automação.

Linux & macOS

```
{
    "AutomationExecutionId": "a4a3c0e9-7efd-462a-8594-01234EXAMPLE"
}
```

Windows

```
{
    "AutomationExecutionId": "a4a3c0e9-7efd-462a-8594-01234EXAMPLE"
}
```

PowerShell

```
a4a3c0e9-7efd-462a-8594-01234EXAMPLE
```

5. Execute o comando a seguir para visualizar a automação. Substitua cada *automation execution ID* (ID de execução de automação) por suas próprias informações.

   Linux & macOS

   ```
   aws ssm describe-automation-executions \
       --filter Key=ExecutionId,Values=automation execution ID
   ```

   Windows

   ```
   aws ssm describe-automation-executions ^
       --filter Key=ExecutionId,Values=automation execution ID
   ```

PowerShell

```
Get-SSMAutomationExecutionList | `
    Where {$_.AutomationExecutionId -eq "automation execution ID"}
```

6. Para visualizar detalhes sobre o andamento da automação, execute o comando a seguir. Substitua cada *automation execution ID* (ID de execução de automação) por suas próprias informações.

   Linux & macOS

   ```
   aws ssm get-automation-execution \
       --automation-execution-id automation execution ID
   ```

   Windows

   ```
   aws ssm get-automation-execution ^
       --automation-execution-id automation execution ID
   ```

   PowerShell

   ```
   Get-SSMAutomationExecution `
       -AutomationExecutionId automation execution ID
   ```

   O sistema retorna informações como estas.

   Linux & macOS

   ```
   {
       "AutomationExecution": {
           "StepExecutionsTruncated": false,
           "AutomationExecutionStatus": "Success",
           "MaxConcurrency": "1",
           "Parameters": {},
           "MaxErrors": "1",
           "Outputs": {},
           "DocumentName": "AWS-StopEC2Instance",
           "AutomationExecutionId": "a4a3c0e9-7efd-462a-8594-01234EXAMPLE",
           "ResolvedTargets": {
   ```

```
            "ParameterValues": [
                "i-02573cafcfEXAMPLE"
            ],
            "Truncated": false
        },
        "ExecutionEndTime": 1564681619.915,
        "Targets": [
            {
                "Values": [
                    "DEV"
                ],
                "Key": "tag:ENV"
            }
        ],
        "DocumentVersion": "1",
        "ExecutionStartTime": 1564681576.09,
        "ExecutedBy": "arn:aws:sts::123456789012:assumed-role/Administrator/
Admin",
        "StepExecutions": [
            {
                "Inputs": {
                    "InstanceId": "i-02573cafcfEXAMPLE"
                },
                "Outputs": {},
                "StepName": "i-02573cafcfEXAMPLE",
                "ExecutionEndTime": 1564681619.093,
                "StepExecutionId": "86c7b811-3896-4b78-b897-01234EXAMPLE",
                "ExecutionStartTime": 1564681576.836,
                "Action": "aws:executeAutomation",
                "StepStatus": "Success"
            }
        ],
        "TargetParameterName": "InstanceId",
        "Mode": "Auto"
    }
}
```

Windows

```
{
    "AutomationExecution": {
        "StepExecutionsTruncated": false,
        "AutomationExecutionStatus": "Success",
```

```
        "MaxConcurrency": "1",
        "Parameters": {},
        "MaxErrors": "1",
        "Outputs": {},
        "DocumentName": "AWS-StopEC2Instance",
        "AutomationExecutionId": "a4a3c0e9-7efd-462a-8594-01234EXAMPLE",
        "ResolvedTargets": {
            "ParameterValues": [
                "i-02573cafcfEXAMPLE"
            ],
            "Truncated": false
        },
        "ExecutionEndTime": 1564681619.915,
        "Targets": [
            {
                "Values": [
                    "DEV"
                ],
                "Key": "tag:ENV"
            }
        ],
        "DocumentVersion": "1",
        "ExecutionStartTime": 1564681576.09,
        "ExecutedBy": "arn:aws:sts::123456789012:assumed-role/Administrator/
Admin",
        "StepExecutions": [
            {
                "Inputs": {
                    "InstanceId": "i-02573cafcfEXAMPLE"
                },
                "Outputs": {},
                "StepName": "i-02573cafcfEXAMPLE",
                "ExecutionEndTime": 1564681619.093,
                "StepExecutionId": "86c7b811-3896-4b78-b897-01234EXAMPLE",
                "ExecutionStartTime": 1564681576.836,
                "Action": "aws:executeAutomation",
                "StepStatus": "Success"
            }
        ],
        "TargetParameterName": "InstanceId",
        "Mode": "Auto"
    }
}
```

PowerShell

```
AutomationExecutionId       : a4a3c0e9-7efd-462a-8594-01234EXAMPLE
AutomationExecutionStatus   : Success
CurrentAction               :
CurrentStepName             :
DocumentName                : AWS-StopEC2Instance
DocumentVersion             : 1
ExecutedBy                  : arn:aws:sts::123456789012:assumed-role/
Administrator/Admin
ExecutionEndTime            : 8/1/2019 5:46:59 PM
ExecutionStartTime          : 8/1/2019 5:46:16 PM
FailureMessage              :
MaxConcurrency              : 1
MaxErrors                   : 1
Mode                        : Auto
Outputs                     : {}
Parameters                  : {}
ParentAutomationExecutionId :
ProgressCounters            :
ResolvedTargets             :
 Amazon.SimpleSystemsManagement.Model.ResolvedTargets
StepExecutions              : {i-02573cafcfEXAMPLE}
StepExecutionsTruncated     : False
Target                      :
TargetLocations             : {}
TargetMaps                  : {}
TargetParameterName         : InstanceId
Targets                     : {tag:Name}
```

> **Note**
>
> Você pode também monitorar o status da automação no console. Na lista de execuções do Automation, escolha a execução que você acabou de processar e depois escolha a guia Execution steps (Etapas da execução). Esta guia mostra o status das ações de automação.

Mapeamento de destino para uma automação

Use o parâmetro `Targets` para definir rapidamente quais recursos serão usados como destino por uma automação. Por exemplo, se você quiser executar uma automação que reinicia suas instâncias gerenciadas, em vez de selecionar manualmente dezenas de IDs de instância no console ou digitá-los em um comando, especifique as instâncias de destino especificando as tags do Amazon Elastic Compute Cloud (Amazon EC2) com o parâmetro `Targets`.

Quando você executa uma automação que usa um destino, o AWS Systems Manager cria uma automação filho para cada destino. Por exemplo, se você definir o destino dos volumes do Amazon Elastic Block Store (Amazon EBS) especificando tags, e essas tags forem resolvidas para 100 volumes do Amazon EBS, o Systems Manager criará 100 automações filho. A automação pai é concluída quando todas as automações filho alcançam um estado final.

> **Note**
>
> Quaisquer `input parameters` especificados em runtime (na seção Input parameters (parâmetros de entrada) do console ou usando a opção `parameters` na linha de comando) serão automaticamente processados por todas as automações filho.

Você pode definir recursos como o destino para uma automação usando tags, grupos de recursos e valores de parâmetros. Além disso, você pode usar a opção `TargetMaps` para definir o destino de vários valores de parâmetro na linha de comando ou em um arquivo. A seção a seguir descreve cada uma dessas opções de destino em mais detalhes.

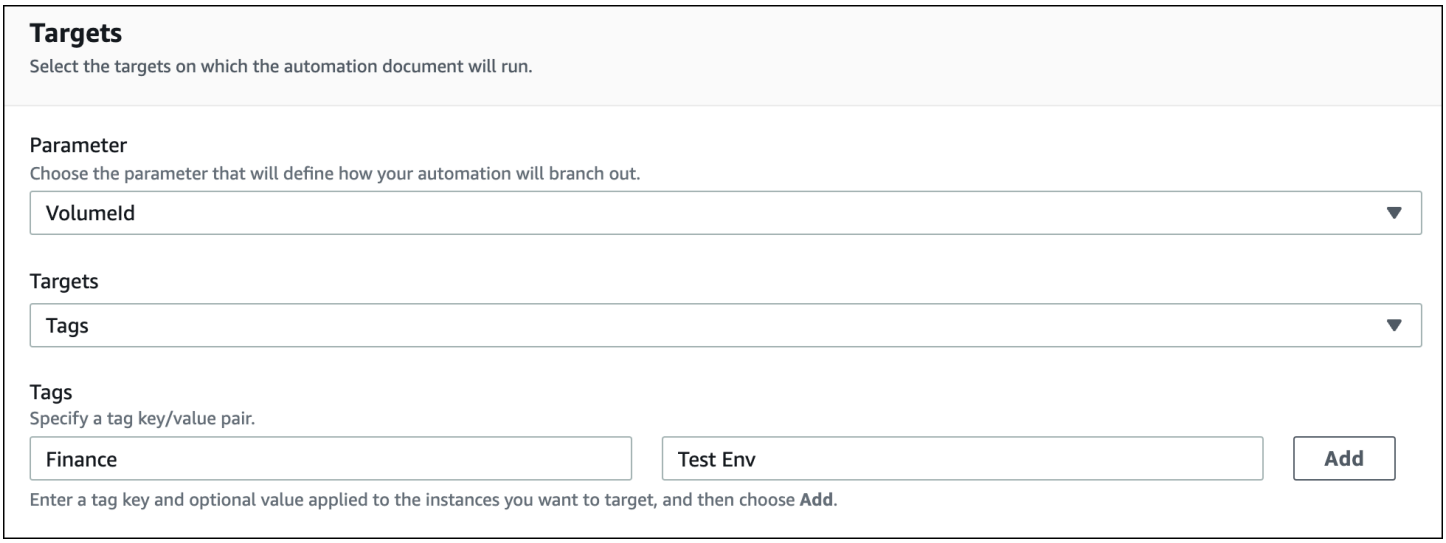
Definir uma etiqueta como destino

É possível especificar uma única etiqueta como o destino de uma automação. Muitos recursos da AWS oferecem suporte a tags, incluindo instâncias do Amazon Elastic Compute Cloud (Amazon EC2) e Amazon Relational Database Service (Amazon RDS), volumes e snapshots do Amazon Elastic Block Store (Amazon EBS), grupos de recursos e buckets do Amazon Simple Storage Service (Amazon S3), entre outros. É possível executar rapidamente a automação nos recursos da AWS definindo uma etiqueta como destino. Uma etiqueta é um par de chave-valor, como Operating_System:Linux ou Department:Finance. Se você atribuir um nome específico a um recurso, você também pode usar a palavra "Name" como chave, e o nome do recurso como valor.

Quando você especifica uma tag como o destino para uma automação, você também deve especificar um parâmetro de destino. O parâmetro de destino usa a opção `TargetParameterName`.

Ao escolher um parâmetro de destino, você define o tipo de recurso no qual a automação é executada. O parâmetro de destino especificado com a tag deve ser um parâmetro válido definido no runbook. Por exemplo, para ter dezenas de instâncias do EC2 como destino usando tags, escolha o parâmetro de destino `InstanceId`. Ao escolher este parâmetro, você define as instâncias como o tipo de recurso para a automação. Ao criar um runbook personalizado, é necessário especificar o Tipo de destino como `/AWS::EC2::Instance` para garantir que somente instâncias sejam usadas. Caso contrário, todos os recursos com a mesma etiqueta serão direcionados. Ao definir instâncias como destino usando uma etiqueta, é possível incluir instâncias encerradas.

A captura de tela a seguir usa o runbook `AWS-DetachEBSVolume`. O parâmetro de destino lógico é `VolumeId`.



O runbook `AWS-DetachEBSVolume` também inclui uma propriedade especial chamada Target type (Tipo de destino), que é definida como `/AWS::EC2::Volume`. Isso significa que, se o par de tag e chave `Finance:TestEnv` retornar diferentes tipos de recursos (por exemplo, instâncias do EC2, volumes do Amazon EBS e snapshots do Amazon EBS), apenas os volumes do Amazon EBS serão usados.

Important

Os nomes dos parâmetros de destino diferenciam maiúsculas de minúsculas. Se você executar automações usando a AWS Command Line Interface (AWS CLI) ou AWS Tools for Windows PowerShell, insira o nome do parâmetro de destino exatamente como ele é definido no runbook. Se não fizer isso, o sistema retornará um erro `InvalidAutomationExecutionParametersException`. Você pode usar a operação de API DescribeDocument para ver informações sobre os parâmetros de destino disponíveis

em um runbook específico. Veja a seguir um exemplo de comando da AWS CLI que fornece informações sobre o documento `AWS-DeleteSnapshot`.

```
aws ssm describe-document \
    --name AWS-DeleteSnapshot
```

Aqui estão alguns exemplos de comandos da AWS CLI que definem recursos como destino usando etiquetas.

Exemplo 1: definir uma etiqueta como destino usando um par de chave-valor para reiniciar instâncias do Amazon EC2

Este exemplo reinicia todas as instâncias do Amazon EC2 que estiverem marcadas com uma chave Department e um valor HumanResources. O parâmetro de destino usa o parâmetro InstanceId do runbook. O exemplo usa um parâmetro adicional para executar a automação usando uma função de serviço do Automation (também chamado de função assumida).

```
aws ssm start-automation-execution \
    --document-name AWS-RestartEC2Instance \
    --targets Key=tag:Department,Values=HumanResources \
    --target-parameter-name InstanceId \
    --parameters "AutomationAssumeRole=arn:aws:iam::111122223333:role/
AutomationServiceRole"
```

Exemplo 2: definir uma etiqueta como destino usando um par de chave-valor para excluir snapshots do Amazon EBS

O exemplo a seguir usa o runbook `AWS-DeleteSnapshot` para excluir todos os snapshots com uma chave de Nome e um valor de January2018Backups. O parâmetro de destino usa o parâmetro VolumeId.

```
aws ssm start-automation-execution \
    --document-name AWS-DeleteSnapshot \
    --targets Key=tag:Name,Values=January2018Backups \
    --target-parameter-name VolumeId
```

AWS Resource Groups como destino

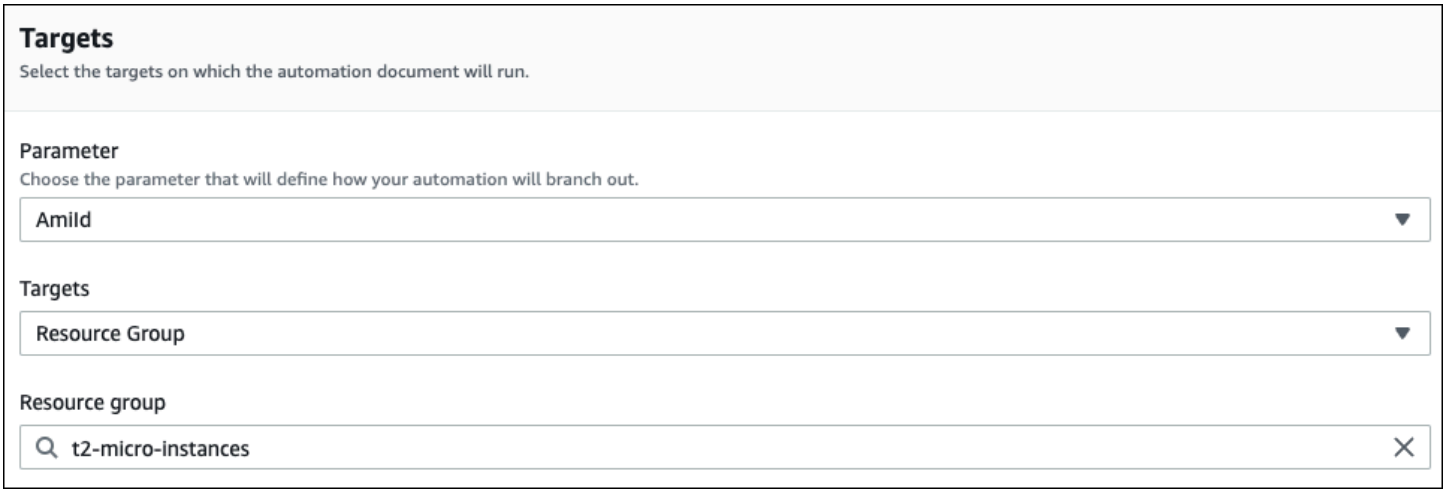
É possível especificar um único grupo de recursos da AWS como o destino de uma automação. O Systems Manager cria uma automação filho para cada objeto no grupo de recursos de destino.

Por exemplo, digamos que um de seus grupos de recursos tenha o nome de PatchedAMIs. Esse grupo de recursos inclui uma lista de 25 Amazon Machine Images (AMIs) do Windows que recebem patches rotineiramente. Se você executar uma automação que use o runbook `AWS-CreateManagedWindowsInstance` e tenha como destino esse grupo de recursos, o Systems Manager criará uma automação filho para cada uma das 25 AMIs. Isso significa que, ao especificar o grupo de recursos PatchedAMIs, a automação cria 25 instâncias em uma lista de AMIs corrigidas pelos patches. A automação pai é concluída quando todas as automações filho finalizam o processamento ou alcançam um estado final.

O comando da AWS CLI a seguir se aplica ao exemplo do grupo de recursos PatchAMIs. O comando usa o parâmetro AmiId para a opção `--target-parameter-name`. O comando não inclui um parâmetro adicional que define qual tipo de instância criar a partir de cada AMI. O runbook `AWS-CreateManagedWindowsInstance` usa como padrão o tipo de instância t2.medium. Esse comando criaria 25 instâncias t2.medium do Amazon EC2 para o Windows Server.

```
aws ssm start-automation-execution \
    --document-name AWS-CreateManagedWindowsInstance \
    --targets Key=ResourceGroup,Values=PatchedAMIs  \
    --target-parameter-name AmiId
```

O exemplo de console a seguir usa um grupo de recursos chamado t2-micro-instances.



Direcionar valores de parâmetro

Você também pode direcionar um valor de parâmetro. Insira `ParameterValues` como chave e, em seguida, insira o valor do recurso específico em que você deseja que a automação seja executada. Se você especificar vários valores, o Systems Manager executará uma automação secundária em cada valor especificado.

Por exemplo, digamos que o runbook inclui um parâmetro InstanceID. Se você direcionar os valores do parâmetro InstanceID ao executar o Automation, o Systems Manager executará uma automação filho para cada valor especificado para o ID da instância. A automação pai é concluída quando a automação conclui a execução de cada instância especificada, ou se a automação falha. Você pode direcionar um máximo de 50 valores de parâmetro.

O exemplo a seguir usa o runbook `AWS-CreateImage`. O nome do parâmetro de destino especificado é InstanceId. A chave usa ParameterValues. Os valores são dois IDs de instâncias do Amazon EC2. Esse comando cria uma automação para cada instância, que produz uma AMI em cada instância.

```
aws ssm start-automation-execution
    --document-name AWS-CreateImage \
    --target-parameter-name InstanceId \
    --targets Key=ParameterValues,Values=i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE
```

> **Note**
>
> `AutomationAssumeRole` não é um parâmetro válido. Não escolha esse item ao executar fluxos automações que direcionam um valor de parâmetro.

Direcionar mapas de valores de parâmetros

A opção `TargetMaps` expande sua capacidade de direcionar `ParameterValues`. Você pode inserir um conjunto de valores de parâmetro usando `TargetMaps` na linha de comando. Você pode especificar um máximo de 50 valores de parâmetro na linha de comando. Se quiser executar comandos que especificam mais de 50 valores de parâmetro, você poderá inserir os valores em um arquivo JSON. Você pode, então, chamar o arquivo da linha de comando.

> **Note**
>
> A opção `TargetMaps` não tem suporte no console.

Use o formato a seguir para especificar vários valores de parâmetro usando a opção `TargetMaps` em um comando. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

```
aws ssm start-automation-execution \
    --document-name runbook name \
    --target-maps “parameter=value, parameter 2=value, parameter 3=value”  “parameter
 4=value, parameter 5=value, parameter 6=value”
```

Se você deseja inserir mais de 50 valores de parâmetro para a opção `TargetMaps`, especifique os valores em um arquivo usando o seguinte formato JSON. Usar um arquivo JSON também melhora a legibilidade ao fornecer vários valores de parâmetro.

```
[

    {“parameter”: "value", “parameter 2”: "value", “parameter 3”: "value"},

    {“parameter 4”: "value", “parameter 5”: "value", "parameter 6": "value"}

]
```

Salve o arquivo com a extensão .json. Você pode chamar o arquivo usando o seguinte comando: Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

```
aws ssm start-automation-execution \
    --document-name runbook name \
    --parameters input parameters \
    --target-maps path to file/file name.json
```

Você também pode baixar o arquivo de um bucket do Amazon Simple Storage Service (Amazon S3), desde que tenha permissão para ler dados do bucket. Use o seguinte formato de comando: Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

```
aws ssm start-automation-execution \
```

```
    --document-name runbook name \
    --target-maps http://amzn-s3-demo-bucket.s3.amazonaws.com/file_name.json
```

Este é um cenário de exemplo para ajudar você a compreender a opção `TargetMaps`. Nesse cenário, um usuário deseja criar instâncias do Amazon EC2 de diferentes tipos em diferentes AMIs. Para executar essa tarefa, o usuário cria um runbook chamado AMI_Testing. Este runbook define dois parâmetros de entrada: `instanceType` e `imageId`.

```
{
  "description": "AMI Testing",
  "schemaVersion": "0.3",
  "assumeRole": "{{assumeRole}}",
  "parameters": {
    "assumeRole": {
      "type": "String",
      "description": "Role under which to run the automation",
      "default": ""
    },
    "instanceType": {
      "type": "String",
      "description": "Type of EC2 Instance to launch for this test"
    },
    "imageId": {
      "type": "String",
      "description": "Source AMI id from which to run instance"
    }
  },
  "mainSteps": [
    {
      "name": "runInstances",
      "action": "aws:runInstances",
      "maxAttempts": 1,
      "onFailure": "Abort",
      "inputs": {
        "ImageId": "{{imageId}}",
        "InstanceType": "{{instanceType}}",
        "MinInstanceCount": 1,
        "MaxInstanceCount": 1
      }
    }
  ],
  "outputs": [
    "runInstances.InstanceIds"
```

```
  ]
}
```

O usuário especifica os seguintes valores de parâmetro de destino em um arquivo chamado `AMI_instance_types.json`.

```
[
  {
    "instanceType" : ["t2.micro"],
    "imageId" : ["ami-b70554c8"]
  },
  {
    "instanceType" : ["t2.small"],
    "imageId" : ["ami-b70554c8"]
  },
  {
    "instanceType" : ["t2.medium"],
    "imageId" : ["ami-cfe4b2b0"]
  },
  {
    "instanceType" : ["t2.medium"],
    "imageId" : ["ami-cfe4b2b0"]
  },
  {
    "instanceType" : ["t2.medium"],
    "imageId" : ["ami-cfe4b2b0"]
  }
]
```

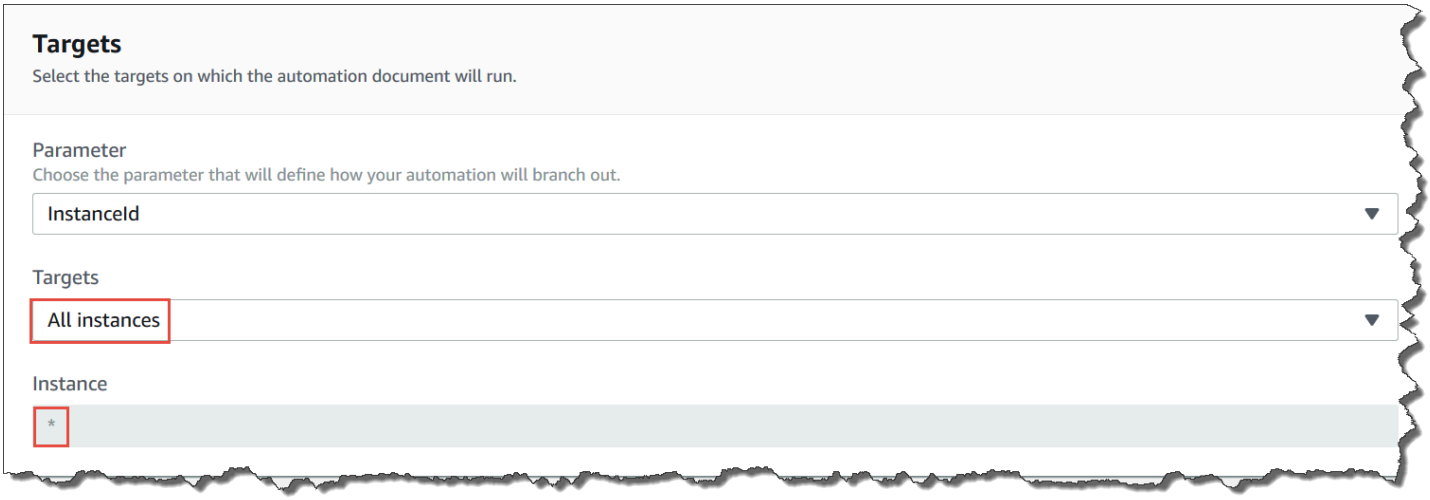
O usuário pode executar a automação e criar as cinco instâncias do EC2 definidas em `AMI_instance_types.json`, executando o seguinte comando:

```
aws ssm start-automation-execution \
    --document-name AMI_Testing \
    --target-parameter-name imageId \
    --target-maps file:///home/TestUser/workspace/runinstances/AMI_instance_types.json
```

Direcionar todas as instâncias do Amazon EC2

É possível executar uma automação em todas as instâncias do Amazon EC2 na Conta da AWS e na Região da AWS atuais ao escolher Todas as instâncias na lista Destinos. Por exemplo, se você

quiser reiniciar todas as instâncias do Amazon EC2 em sua Conta da AWS e nas Região da AWS atuais, escolha o runbook **`AWS-RestartEC2Instance`** e selecione Todas as instâncias na lista Destinos.



Depois de escolher All instances (Todas as instâncias), o Systems Manager preenche o campo Instância com um asterisco (*) e torna o campo indisponível para alterações (o campo ficará esmaecido). O Systems Manager também torna o campo InstanceId em Input parameters (Parâmetros de entrada) indisponível para alterações. Tornar esses campos indisponíveis para alterações será o comportamento esperado se você optar por direcionar todas as instâncias.

Automações de controle em grande escala

Você pode controlar a implantação de uma automação em uma frota de recursos da AWS, especificando um valor de simultaneidade e um limite de erro. Simultaneidade e limite de erro são coletivamente chamados de controles de taxa.

Simultaneidade

Use a simultaneidade para especificar quantos recursos podem executar uma automação simultaneamente. A simultaneidade ajuda a limitar o impacto ou o tempo de inatividade em seus recursos ao processar uma automação. Você pode especificar um número absoluto de recursos, por exemplo, 20 ou uma porcentagem do conjunto de destino, por exemplo, 10%.

O sistema de fila fornece a automação a um único recurso e aguarda até que a invocação inicial seja concluída antes de enviar a automação para mais dois recursos. O sistema envia a automação de forma exponencial para mais recursos até que o valor de simultaneidade seja atingido.

Limites de erro

Um limite de erro permite que você especifique quantas automações podem falhar antes que o AWS Systems Manager pare de enviá-las para outros recursos. Você pode especificar um número absoluto de erros, como 10, ou uma porcentagem do conjunto de destino, como 10%.

Se você especificar um número absoluto de 3 erros, por exemplo, o sistema deixará de executar a automação quando o quarto erro for recebido. Se você especificar 0, o sistema deixará de executar a automação em destinos adicionais depois que o primeiro resultado do erro for retornado.

Se você enviar uma automação para 50 instâncias, por exemplo, e definir o limite de erro como 10%, o sistema deixará de enviar o comando para instâncias adicionais quando o quinto erro for recebido. As invocações que já estiverem executando uma automação, quando um limite de erro for atingido, poderão ser concluídas, mas algumas dessas automações também podem falhar. Se você precisar garantir que não haverá mais erros do que o número especificado para o limite de erro, defina o valor de Concurrency (Simultaneidade) como 1 para que as automações prossigam uma de cada vez.

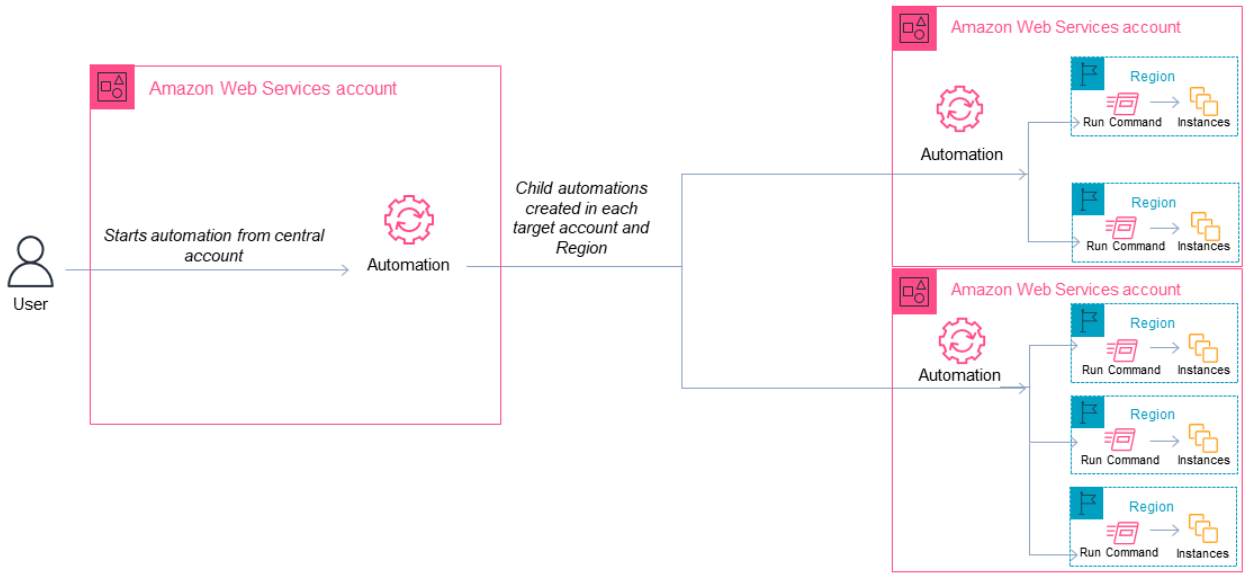
Executar automações em várias regiões e contas da Regiões da AWS

Você pode executar automações do AWS Systems Manager em várias Regiões da AWS e Contas da AWS ou em unidades organizacionais (UOs) do AWS Organizations em uma conta central. O Automation é uma ferramenta do AWS Systems Manager. Executar automações em várias regiões e contas ou UOs reduz o tempo necessário para administrar os recursos da AWS, ao mesmo tempo em que melhora a segurança do ambiente de computação.

Por exemplo, você pode fazer o seguinte usando runbooks de automação:

- Implementar atualizações de patches e segurança de maneira centralizada.
- Corrigir desvios de conformidade em configurações da VPC ou em políticas de bucket do Amazon S3.
- Gerenciar recursos, como instâncias do EC2 do Amazon Elastic Compute Cloud (Amazon EC2), em grande escala.

O diagrama a seguir mostra um exemplo de um usuário que está executando o runbook `AWS-RestartEC2Instances` em várias regiões e contas usando uma conta central. A automação localiza as instâncias usando as tags especificadas nas regiões e contas de destino.



Escolher uma conta central para o Automation

Se você quiser executar automações em UOs, a conta central deve ter permissões para listar todas as contas nas UOs. Isso só é possível usando uma conta de administrador delegado ou a conta de gerenciamento da organização. Recomendamos seguir as práticas recomendadas do AWS Organizations e usar uma conta de administrador delegado. Para obter mais informações sobre as práticas recomendadas do AWS Organizations, consulte Práticas recomendadas para a conta de gerenciamento no Guia do usuário do AWS Organizations. Para criar uma conta de administrador delegado para o Systems Manager, você pode usar o comando `register-delegated-administrator` com a AWS CLI, conforme exibido no exemplo a seguir.

```
aws organizations register-delegated-administrator \
    --account-id delegated admin account ID \
    --service-principal ssm.amazonaws.com
```

Se você quiser executar automações em várias contas que não sejam gerenciadas pelo AWS Organizations, recomendamos a criação de uma conta dedicada para o gerenciamento de automação. A execução de todas as automações entre contas usando uma conta dedicada simplifica o gerenciamento de permissões do IAM, os esforços de solução de problemas e cria uma camada de

separação entre operações e administração. Essa abordagem também é recomendada se você usar o AWS Organizations, mas só quiser segmentar contas individuais e não UOs.

Funcionamento da execução de automações

Executar automações em várias regiões e contas ou UOs ocorre da seguinte forma:

1. Acesse a conta que você deseja configurar como a conta central do Automation.
2. Siga o procedimento Configurar permissões da conta de gerenciamento para execução da automação de várias regiões e contas neste tópico para criar os seguintes perfis do IAM:
   - **`AWS-SystemsManager-AutomationAdministrationRole`**: essa função permite que o usuário execute automação em várias contas e UOs.
   - **`AWS-SystemsManager-AutomationExecutionRole`**: essa função permite que o usuário execute automação nas contas de destino.
3. Escolha o runbook, as regiões e as contas ou OUs em que você deseja executar a automação.

   > **Note**
   >
   > Certifique-se de que a UO de destino contenha as contas desejadas. Se você escolher um runbook personalizado, ele deverá ser compartilhado com todas as contas de destino. Para obter informações sobre como compartilhar runbooks, consulte Compartilhar documentos do Systems Manager. Para obter informações sobre como usar runbooks compartilhados, consulte Usar documentos compartilhados do .

4. Execute a automação.

   > **Note**
   >
   > Ao executar automações em várias regiões, contas ou UOs, a automação executada na conta principal inicia automações filho em cada uma das contas de destino. A automação na conta principal contém etapas de `aws:executeAutomation` para cada uma das contas de destino.

5. Use as operações de API GetAutomationExecution, DescribeAutomationStepExecutions e DescribeAutomationExecutions no console do AWS Systems Manager ou na AWS CLI para monitorar o progresso da automação. O resultado das etapas para a automação em sua conta principal será o `AutomationExecutionId`das automações filho. Para exibir a

saída das automações filho criadas em suas contas de destino, especifique a conta, região e `AutomationExecutionId` apropriadas na sua solicitação.

Configurar permissões da conta de gerenciamento para execução da automação de várias regiões e contas

Use o procedimento a seguir para criar as funções do IAM necessárias para a automação do Systems Manager Automation em várias regiões e várias contas, usando o AWS CloudFormation. Esse procedimento descreve como criar a função **`AWS-SystemsManager-AutomationAdministrationRole`**. Você só precisa criar essa função na conta de gerenciamento do Automation. Esse procedimento também descreve como criar a função **`AWS-SystemsManager-AutomationExecutionRole`**. Você deve criar essa função em todas as contas que deseja direcionar para executar automações em várias regiões e contas. Recomendamos usar o AWS CloudFormation StackSets para criar a função **`AWS-SystemsManager-AutomationExecutionRole`** nas contas que você quiser direcionar para executar automações em várias regiões e contas.

Criar as funções de administração do IAM necessárias para automações em várias regiões e várias contas usando o AWS CloudFormation

1. Faça download e descompacte `AWS-SystemsManager-AutomationAdministrationRole.zip`. Ou, se suas contas forem gerenciadas pelo AWS Organizations, `AWS-SystemsManager-AutomationAdministrationRole(org).zip`. Este arquivo contém o arquivo de modelo do `AWS-SystemsManager-AutomationAdministrationRole.yaml` AWS CloudFormation.
2. Abra o console do AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
3. Selecione Criar pilha.
4. Na seção Specify template (Especificar modelo) escolha Upload a template (Fazer upload de um modelo).
5. Selecione Choose file (Escolha o arquivo) e, depois, escolha o modelo de arquivo do `AWS-SystemsManager-AutomationAdministrationRole.yaml` AWS CloudFormation.
6. Escolha Próximo.
7. Na página Specify stack details (Especificar detalhes da tarefa), no campo Stack name (Nome da pilha), insira um nome.
8. Escolha Próximo.

9. Na página Configure stack options (Configurar opções de pilha), insira valores para as opções que você quiser usar. Escolha Próximo.
10. Na página Review (Revisão) role para baixo e escolha a opção I acknowledge that AWS CloudFormation might create IAM resources with custom names (Estou ciente de que o poderá criar recursos do IAM com nomes personalizados).
11. Selecione Criar pilha.

O AWS CloudFormation mostra o status CREATE_IN_PROGRESS por cerca de três minutos. O status mudará para CREATE_COMPLETE.

É necessário repetir esse procedimento em todas as contas que você quiser direcionar para executar automações em várias regiões e contas.

Criar as funções de automação do IAM necessárias para automações em várias regiões e várias contas usando o AWS CloudFormation

1. Faça download do `AWS-SystemsManager-AutomationExecutionRole.zip`. Ou, se suas contas forem gerenciadas pelo AWS Organizations, `AWS-SystemsManager-AutomationExecutionRole (org).zip`. Este arquivo contém o arquivo de modelo do `AWS-SystemsManager-AutomationExecutionRole.yaml` AWS CloudFormation.
2. Abra o console do AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
3. Selecione Criar pilha.
4. Na seção Specify template (Especificar modelo) escolha Upload a template (Fazer upload de um modelo).
5. Selecione Choose file (Escolha o arquivo) e, depois, escolha o modelo de arquivo do `AWS-SystemsManager-AutomationExecutionRole.yaml` AWS CloudFormation.
6. Escolha Próximo.
7. Na página Specify stack details (Especificar detalhes da tarefa), no campo Stack name (Nome da pilha), insira um nome.
8. Na seção Parameters (Parâmetros), no campo AdminAccountId, insira o ID da conta central do Automation.
9. Se você estiver configurando essa função para um ambiente AWS Organizations, haverá outro campo chamado OrganizationId na seção. Insira o ID da sua organização da AWS.
10. Escolha Próximo.

11. Na página Configure stack options (Configurar opções de pilha), insira valores para as opções que você quiser usar. Escolha Próximo.
12. Na página Review (Revisão) role para baixo e escolha a opção I acknowledge that AWS CloudFormation might create IAM resources with custom names (Estou ciente de que o poderá criar recursos do IAM com nomes personalizados).
13. Selecione Criar pilha.

O AWS CloudFormation mostra o status CREATE_IN_PROGRESS por cerca de três minutos. O status mudará para CREATE_COMPLETE.

Execute uma automação em várias regiões e contas (console)

O procedimento a seguir descreve como usar o console do Systems Manager para executar uma automação em várias regiões e contas da conta de gerenciamento do Automation.

Antes de começar

Antes de concluir o seguinte procedimento, anote as seguintes informações:

- O usuário ou o perfil usado para executar uma automação em diversas regiões ou diversas contas deve ter a permissão `iam:PassRole` para o perfil `AWS-SystemsManager-AutomationAdministrationRole`.
- IDs de contas da Conta da AWS ou UOs nos quais você deseja executar a automação.
- Regiões suportadas pelo Systems Manager onde você deseja executar a automação.
- A chave e o valor da tag, ou o nome do grupo de recursos, nos quais você deseja executar a automação.

Para executar uma automação em várias regiões e contas

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, selecione Automation e Execute automation (Executar automação).
3. Na lista Automation document (Documento do Automation), escolha um runbook. Escolha uma ou mais opções no painel Document categories (Categorias de documentos) para filtrar documentos SSM de acordo com sua finalidade. Para visualizar um runbook que você tenha, escolha a guia Owned by me (De minha propriedade). Para visualizar um runbook compartilhado com sua conta, escolha a guia Shared with me (Compartilhado comigo). Para visualizar todos os runbooks, escolha a guia All documents (Todos os documentos).

> **Note**
>
> Você pode visualizar informações sobre um runbook, selecionando o nome dele.

4. Na seção Document details (Detalhes do documento), verifique se Document version (Versão do documento) está definida como a versão que você quer executar. O sistema inclui as seguintes opções de versão:

   - Versão padrão no runtime: escolha essa opção se o runbook do Automation for atualizado periodicamente e uma nova versão padrão for atribuída.
   - Versão mais recente no runtime: escolha essa opção se o runbook do Automation for atualizado periodicamente e se você quiser executar a versão mais recente.
   - 1 (padrão): escolha esta opção para executar a primeira versão do documento, que é a versão padrão.

5. Escolha Próximo.
6. Na página Execute automation document (Executar documento de automação), escolha Multi-account and Region (Várias contas e região).
7. Na seção Target accounts and Regions (Regiões e contas de destino), use o campo Accounts and organizational (OUs) (Contas e unidades organizacionais, UOs) para especificar as diferentes Contas da AWS ou unidades organizacionais (UOs) da AWS em que você deseja executar a automação. Separe várias contas ou UOs com uma vírgula.
8. Use a lista de Regiões da AWS para escolher uma ou mais regiões onde você deseja executar a automação.
9. Use as opções Multi-Region and account rate control (Controle da taxa de várias contas e regiões) para restringir a execução da automação para um número limitado de contas em execução em um número limitado de regiões. Essas opções não restringem o número de recursos da AWS que podem executar as automações.

   a. Na seção Location (account-Region pair) concurrency (Simultaneidade da localização - par conta/região), escolha uma opção para restringir o número de automações que podem ser executadas em várias contas e regiões ao mesmo tempo. Por exemplo, se você optar por executar uma automação em cinco (5) contas da Contas da AWS que estiverem localizadas em quatro (4) Regiões da AWS, o Systems Manager executará as automações em um total de 20 pares de conta/região. Você pode usar essa opção para especificar um número absoluto, como **2**, para que a automação seja executada simultaneamente em apenas dois

pares de contas ou regiões. Outra opção é especificar uma porcentagem dos pares conta/região que podem ser executados ao mesmo tempo. Por exemplo, com 20 pares de conta/região, se você especificar 20%, a automação será executada simultaneamente em um máximo de cinco (5) pares de conta/região.

- Escolha targets (destinos) para inserir um número absoluto de pares de conta/região que podem executar a automação simultaneamente.
- Escolha percent (por cento) para inserir uma porcentagem do número total de pares de contas/regiões que podem executar a automação simultaneamente.

b. Na seção Error threshold (Limite de erro), escolha uma opção:

- Escolha errors (erros) para inserir um número absoluto de erros permitidos antes que o Automation pare de enviar a automação para outros recursos.
- Escolha percent (por cento) para inserir uma porcentagem de erros permitidos antes que o Automation deixe de enviar a automação para outros recursos.

10. Na seção Targets (Destinos), escolha como você quer definir os destinos dos recursos da AWS em que você deseja executar o Automation. Essas opções são obrigatórias.

a. Use a lista Parameter (Parâmetro) para escolher um parâmetro. Os itens na lista Parameter (Parâmetro) são determinados pelos parâmetros no runbook do Automation que você selecionou no início deste procedimento. Ao escolher um parâmetro, você define o tipo de recurso em que o fluxo de trabalho da Automação é executado.

b. Use a lista Targets (Destinos) para escolher como você deseja definir o destino dos recursos.

i. Se você optar por definir o destino dos recursos usando valores de parâmetro, insira o valor do parâmetro para aquele escolhido na seção Input parameters (Parâmetros de entrada).

ii. Se você escolher definir o destino dos recursos usando o AWS Resource Groups, escolha o nome do grupo na lista Resource Group (Grupo de recursos).

iii. Se você optar por definir o destino dos recursos usando tags, insira a chave de tag e (opcionalmente) o valor da tag nos campos fornecidos. Escolha Adicionar.

iv. Se você quiser executar um runbook do Automation em todas as instâncias nas Conta da AWS e Região da AWS atuais, escolha All instances (Todas as instâncias).

11. Na seção Input parameters (Parâmetros de entrada), especifique as entradas necessárias. Escolha o perfil de serviço do IAM `AWS-SystemsManager-AutomationAdministrationRole` na lista AutomationAssumeRole.

    > **Note**
    >
    > Pode ser que não seja necessário escolher algumas das opções na seção Input parameters (Parâmetros de entrada). Isso ocorre porque você definiu o destino de recursos em várias regiões e contas usando tags ou um grupo de recursos. Por exemplo, se você escolheu o runbook `AWS-RestartEC2Instance`, você não precisará especificar ou escolher IDs de instância na seção Input parameters (Parâmetros de entrada). A automação localiza as instâncias para reiniciar usando as tags que você especificou.

12. (Opcional) Escolha um alarme do CloudWatch a fim de aplicar à sua automação para monitoramento. Para anexar um alarme do CloudWatch à sua automação, a entidade principal do IAM que inicia a automação deve ter permissão para a ação `iam:createServiceLinkedRole`. Para obter mais informações sobre alarmes do CloudWatch, consulte Usar alarmes do Amazon CloudWatch. A automação será cancelada se o alarme for ativado, e as etapas `OnCancel` que você definiu serão executadas. Se você usar o AWS CloudTrail, você verá a chamada de API em sua trilha.
13. Use as opções na seção Rate control (Controle de taxa) para restringir o número de recursos da AWS que podem executar o Automation dentro de cada par conta-região.

    Na seção Concurrency (Simultaneidade), escolha uma opção:

    - Escolha Targets (Destinos) para inserir um número absoluto de destinos que podem executar o fluxo de trabalho de Automação simultaneamente.
    - Escolha Percentage (Porcentagem) para inserir uma porcentagem do conjunto de destino que pode executar o fluxo de trabalho de Automação simultaneamente.
14. Na seção Error threshold (Limite de erro), escolha uma opção:

    - Escolha errors (erros) para inserir um número absoluto de erros permitidos antes que a Automação deixe de enviar o fluxo de trabalho para outros recursos.
    - Escolha percentage (porcentagem) para inserir uma porcentagem de erros permitidos antes que a Automação deixe de enviar o fluxo de trabalho para outros recursos.
15. Clique em Executar.

Executar uma automação em várias regiões e contas (linha de comando)

O procedimento a seguir descreve como usar a AWS CLI (no Linux ou no Windows) ou o Ferramentas da AWS para PowerShell para executar uma automação em várias regiões e contas da conta de gerenciamento do Automation.

Antes de começar

Antes de concluir o seguinte procedimento, anote as seguintes informações:

- IDs de contas da Conta da AWS ou UOs nos quais você deseja executar a automação.
- Regiões suportadas pelo Systems Manager onde você deseja executar a automação.
- A chave e o valor da tag, ou o nome do grupo de recursos, nos quais você deseja executar a automação.

Para executar uma automação em várias regiões e contas

1. Instale e configure a AWS CLI ou o Ferramentas da AWS para PowerShell, caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI e Instalar o Ferramentas da AWS para PowerShell.

2. Use o formato a seguir para criar um comando para executar uma automação em várias regiões e contas: Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   Linux & macOS

```
aws ssm start-automation-execution \
        --document-name runbook name \
        --parameters AutomationAssumeRole=arn:aws:iam::management account
 ID:role/AWS-SystemsManager-AutomationAdministrationRole \
        --target-parameter-name parameter name \
        --targets Key=tag key,Values=value \
        --target-locations Accounts=account ID,account ID
 2,Regions=Region,Region 2,ExecutionRoleName=AWS-SystemsManager-
AutomationExecutionRole
```

Windows

```
aws ssm start-automation-execution ^
        --document-name runbook name ^
        --parameters AutomationAssumeRole=arn:aws:iam::management account
 ID:role/AWS-SystemsManager-AutomationAdministrationRole ^
        --target-parameter-name parameter name ^
        --targets Key=tag key,Values=value ^
        --target-locations Accounts=account ID,account ID
 2,Regions=Region,Region 2,ExecutionRoleName=AWS-SystemsManager-
AutomationExecutionRole
```

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
    $Targets.Key = "tag key"
    $Targets.Values = "value"

    Start-SSMAutomationExecution `
        -DocumentName "runbook name" `
        -Parameter @{
        "AutomationAssumeRole"="arn:aws:iam::management account ID:role/AWS-
SystemsManager-AutomationAdministrationRole" } `
        -TargetParameterName "parameter name" `
        -Target $Targets `
        -TargetLocation @{
        "Accounts"="account ID","account ID 2";
        "Regions"="Region","Region 2";
        "ExecutionRoleName"="AWS-SystemsManager-AutomationExecutionRole" }
```

Aqui estão alguns exemplos:

Exemplo 1: este exemplo reinicia instâncias do EC2 em três regiões em uma organização do AWS Organizations inteira. Para fazer isso, é necessário usar o ID da conta de usuário-raiz da organização como alvo e incluir OUs secundárias.

Linux & macOS

```
aws ssm start-automation-execution \
        --document-name "AWS-RestartEC2Instance" \
```

```
        --target-parameter-name InstanceId \
        --targets '[{"Key":"AWS::EC2::Instance","Values":["*"]}]' \
        --target-locations '[{
            "Accounts": ["r-example"],
            "IncludeChildOrganizationUnits": true,
            "Regions": ["us-east-1", "us-east-2", "us-west-2"]
        }]'
```

Windows

```
aws ssm start-automation-execution \
        --document-name "AWS-RestartEC2Instance" ^
        --target-parameter-name InstanceId ^
        --targets '[{"Key":"AWS::EC2::Instance","Values":["*"]}]' ^
        --target-locations '[{
            "Accounts": ["r-example"],
            "IncludeChildOrganizationUnits": true,
            "Regions": ["us-east-1", "us-east-2", "us-west-2"]
        }]'
```

PowerShell

```
Start-SSMAutomationExecution `
        -DocumentName "AWS-RestartEC2Instance" `
        -TargetParameterName "InstanceId" `
        -Targets '[{"Key":"AWS::EC2::Instance","Values":["*"]}]'
        -TargetLocation @{
            "Accounts"="r-example";
            "Regions"="us-east-1", "us-east-2", "us-west-2";
            "IncludeChildOrganizationUnits"=true}
```

Exemplo 2: este exemplo reinicia instâncias específicas do EC2 em contas e regiões diferentes.

Linux & macOS

```
aws ssm start-automation-execution \
        --document-name "AWS-RestartEC2Instance" \
        --target-parameter-name InstanceId \
        --target-locations '[{
            "Accounts": ["123456789012"],
            "Targets": [{
```

```
                "Key":"ParameterValues",
                "Values":["i-02573cafcfEXAMPLE", "i-0471e04240EXAMPLE"]
            }],
            "TargetLocationMaxConcurrency": "100%",
            "Regions": ["us-east-1"]
        }, {
            "Accounts": ["987654321098"],
            "Targets": [{
                "Key":"ParameterValues",
                "Values":["i-07782c72faEXAMPLE"]
            }],
            "TargetLocationMaxConcurrency": "100%",
            "Regions": ["us-east-2"]
        }]'
```

Windows

```
aws ssm start-automation-execution ^
        --document-name "AWS-RestartEC2Instance" ^
        --target-parameter-name InstanceId ^
        --target-locations '[{
            "Accounts": ["123456789012"],
            "Targets": [{
                "Key":"ParameterValues",
                "Values":["i-02573cafcfEXAMPLE", "i-0471e04240EXAMPLE"]
            }],
            "TargetLocationMaxConcurrency": "100%",
            "Regions": ["us-east-1"]
        }, {
            "Accounts": ["987654321098"],
            "Targets": [{
                "Key":"ParameterValues",
                "Values":["i-07782c72faEXAMPLE"]
            }],
            "TargetLocationMaxConcurrency": "100%",
            "Regions": ["us-east-2"]
        }]'
```

PowerShell

```
Start-SSMAutomationExecution `
        -DocumentName "AWS-RestartEC2Instance" `
        -TargetParameterName "InstanceId" `
```

```
        -Targets '[{"Key":"AWS::EC2::Instance","Values":["*"]}]'
        -TargetLocation @({
            "Accounts"="123456789012",
            "Targets"= @{
                "Key":"ParameterValues",
                "Values":["i-02573cafcfEXAMPLE", "i-0471e04240EXAMPLE"]
            },
            "TargetLocationMaxConcurrency"="100%",
            "Regions"=["us-east-1"]
        }, {
            "Accounts"="987654321098",
            "Targets": @{
                "Key":"ParameterValues",
                "Values":["i-07782c72faEXAMPLE"]
            },
            "TargetLocationMaxConcurrency": "100%",
            "Regions"=["us-east-2"]
        })
```

Exemplo 3: este exemplo reinicia instâncias do EC2 nas contas `123456789012` e `987654321098`, que estão localizadas nas regiões `us-east-2` e `us-west-1`. As instâncias devem ser marcadas com o valor do par de chaves da tag `Env-PROD`.

Linux & macOS

```
aws ssm start-automation-execution \
        --document-name AWS-RestartEC2Instance \
        --parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/AWS-
SystemsManager-AutomationAdministrationRole \
        --target-parameter-name InstanceId \
        --targets Key=tag:Env,Values=PROD \
        --target-locations Accounts=123456789012,987654321098,Regions=us-
east-2,us-west-1,ExecutionRoleName=AWS-SystemsManager-AutomationExecutionRole
```

Windows

```
aws ssm start-automation-execution ^
        --document-name AWS-RestartEC2Instance ^
        --parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/AWS-
SystemsManager-AutomationAdministrationRole ^
        --target-parameter-name InstanceId ^
```

```
        --targets Key=tag:Env,Values=PROD ^
        --target-locations Accounts=123456789012,987654321098,Regions=us-
east-2,us-west-1,ExecutionRoleName=AWS-SystemsManager-AutomationExecutionRole
```

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
    $Targets.Key = "tag:Env"
    $Targets.Values = "PROD"

    Start-SSMAutomationExecution `
        -DocumentName "AWS-RestartEC2Instance" `
        -Parameter @{
        "AutomationAssumeRole"="arn:aws:iam::123456789012:role/AWS-
SystemsManager-AutomationAdministrationRole" } `
        -TargetParameterName "InstanceId" `
        -Target $Targets `
        -TargetLocation @{
        "Accounts"="123456789012","987654321098";
        "Regions"="us-east-2","us-west-1";
        "ExecutionRoleName"="AWS-SystemsManager-AutomationExecutionRole" }
```

Exemplo 4: este exemplo reinicia instâncias do EC2 nas contas `123456789012` e `987654321098`, as quais estão localizadas na região `eu-central-1`. As instâncias devem ser membros do grupo de recursos `prod-instances` da AWS.

Linux & macOS

```
aws ssm start-automation-execution \
        --document-name AWS-RestartEC2Instance \
        --parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/AWS-
SystemsManager-AutomationAdministrationRole \
        --target-parameter-name InstanceId \
        --targets Key=ResourceGroup,Values=prod-instances \
        --target-locations Accounts=123456789012,987654321098,Regions=eu-
central-1,ExecutionRoleName=AWS-SystemsManager-AutomationExecutionRole
```

Windows

```
aws ssm start-automation-execution ^
        --document-name AWS-RestartEC2Instance ^
```

```
        --parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/AWS-
SystemsManager-AutomationAdministrationRole ^
        --target-parameter-name InstanceId ^
        --targets Key=ResourceGroup,Values=prod-instances ^
        --target-locations Accounts=123456789012,987654321098,Regions=eu-
central-1,ExecutionRoleName=AWS-SystemsManager-AutomationExecutionRole
```

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
    $Targets.Key = "ResourceGroup"
    $Targets.Values = "prod-instances"

    Start-SSMAutomationExecution `
        -DocumentName "AWS-RestartEC2Instance" `
        -Parameter @{
        "AutomationAssumeRole"="arn:aws:iam::123456789012:role/AWS-
SystemsManager-AutomationAdministrationRole" } `
        -TargetParameterName "InstanceId" `
        -Target $Targets `
        -TargetLocation @{
        "Accounts"="123456789012","987654321098";
        "Regions"="eu-central-1";
        "ExecutionRoleName"="AWS-SystemsManager-AutomationExecutionRole" }
```

Exemplo 5: este exemplo reinicia instâncias do EC2 na unidade organizacional (OU) `ou-1a2b3c-4d5e6c` da AWS. As instâncias estão localizadas nas regiões `us-west-1` e `us-west-2`. As instâncias devem ser membros do grupo de recursos `WebServices` da AWS.

Linux & macOS

```
aws ssm start-automation-execution \
        --document-name AWS-RestartEC2Instance \
        --parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/AWS-
SystemsManager-AutomationAdministrationRole \
        --target-parameter-name InstanceId \
        --targets Key=ResourceGroup,Values=WebServices \
        --target-locations Accounts=ou-1a2b3c-4d5e6c,Regions=us-west-1,us-
west-2,ExecutionRoleName=AWS-SystemsManager-AutomationExecutionRole
```

Windows

```
aws ssm start-automation-execution ^
        --document-name AWS-RestartEC2Instance ^
        --parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/AWS-
SystemsManager-AutomationAdministrationRole ^
        --target-parameter-name InstanceId ^
        --targets Key=ResourceGroup,Values=WebServices ^
        --target-locations Accounts=ou-1a2b3c-4d5e6c,Regions=us-west-1,us-
west-2,ExecutionRoleName=AWS-SystemsManager-AutomationExecutionRole
```

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
    $Targets.Key = "ResourceGroup"
    $Targets.Values = "WebServices"

    Start-SSMAutomationExecution `
        -DocumentName "AWS-RestartEC2Instance" `
        -Parameter @{
        "AutomationAssumeRole"="arn:aws:iam::123456789012:role/AWS-
SystemsManager-AutomationAdministrationRole" } `
        -TargetParameterName "InstanceId" `
        -Target $Targets `
        -TargetLocation @{
        "Accounts"="ou-1a2b3c-4d5e6c";
        "Regions"="us-west-1";
        "ExecutionRoleName"="AWS-SystemsManager-AutomationExecutionRole" }
```

O sistema retorna informações semelhantes às seguintes.

Linux & macOS

```
{
        "AutomationExecutionId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE"
    }
```

Windows

```
{
```

```
        "AutomationExecutionId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE"
    }
```

PowerShell

```
4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE
```

3. Execute o comando a seguir para visualizar os detalhes sobre a automação. Substitua *automation execution ID* (ID de execução da automação) por suas próprias informações.

   Linux & macOS

   ```
   aws ssm describe-automation-executions \
           --filters Key=ExecutionId,Values=automation execution ID
   ```

   Windows

   ```
   aws ssm describe-automation-executions ^
           --filters Key=ExecutionId,Values=automation execution ID
   ```

   PowerShell

   ```
   Get-SSMAutomationExecutionList | `
           Where {$_.AutomationExecutionId -eq "automation execution ID"}
   ```

4. Execute o comando a seguir para visualizar os detalhes sobre o andamento da automação.

   Linux & macOS

   ```
   aws ssm get-automation-execution \
           --automation-execution-id 4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE
   ```

   Windows

   ```
   aws ssm get-automation-execution ^
           --automation-execution-id 4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE
   ```

   PowerShell

   ```
   Get-SSMAutomationExecution `
   ```

```
    -AutomationExecutionId a4a3c0e9-7efd-462a-8594-01234EXAMPLE
```

> **Note**
>
> Você pode também monitorar o status da automação no console. Na lista de execuções do Automation, escolha a execução que você acabou de processar e depois escolha a guia Execution steps (Etapas da execução). Esta guia mostra o status das ações de automação.

Mais informações

- Aplicação de patch centralizada em várias contas e regiões com a Automação do AWS Systems Manager

Executar automações com base em eventos do EventBridge

Você pode iniciar uma automação especificando um runbook como o destino de um evento do Amazon EventBridge. É possível iniciar automações de acordo com um cronograma ou quando ocorrer um evento específico do sistema da AWS. Por exemplo, digamos que você crie um runbook chamado BootStrapInstances, que instala softwares em uma instância quando ela for iniciada. Para especificar o runbook BootStrapInstances (e a automação correspondente) como um destino de um evento do EventBridge, primeiro crie uma nova regra do EventBridge. (Aqui está uma regra de exemplo: Nome do serviço: EC2, Tipo de evento: Notificação de alteração de status da instância do EC2, Estado(s) específico(s): em execução, Qualquer instância.) Depois, use os seguintes procedimentos para especificar o runbook BootStrapInstances como o destino do evento, usando o console do EventBridge e a AWS Command Line Interface (AWS CLI). Quando uma nova instância for iniciada, o sistema executará a automação e instalará o software.

Para obter informações sobre como criar runbooks, consulte Criação dos seus próprios runbooks.

Criar um evento do EventBridge que use um runbook (console)

Use o procedimento a seguir para configurar um runbook como o destino de um evento do EventBridge.

Para configurar um runbook como destino de uma regra de evento do EventBridge

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Regras.
3. Escolha Create rule.
4. Insira um nome e uma descrição para a regra.

   Uma regra não pode ter o mesmo nome que outra na mesma Região e barramento de eventos.
5. Em Barramento de eventos, selecione o barramento de eventos que você deseja associar a essa regra. Se você quiser que essa regra responda a eventos correspondentes provenientes da sua Conta da AWS, selecione default (padrão). Quando um AWS service (Serviço da AWS) na sua conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta.
6. Escolha como a regra é acionada.

Para criar uma regra com base em…	Fazer isso...
Event	a. Em Rule type, escolha Rule with an event pattern. b. Escolha Próximo. c. Em Event source, escolha Eventos da AWS ou eventos de parceiro do EventBridge. d. Na seção Event pattern (Padrão de evento), siga um destes procedimentos: • Para usar um modelo para criar o padrão de eventos, escolha Formulário de padrão de eventos e selecione as opções Origem do

Para criar uma regra com base em…	Fazer isso...	
	evento, Serviço da AWS e Tipo de evento. Se você escolher All Events (Todos os eventos) como tipo de evento, todos os eventos emitidos por esse AWS service (Serviço da AWS) corresponderão à regra. Para personalizar o modelo, escolha Padrão personalizado (editor JSON) e faça as alterações. • Para utilizar um padrão de evento personalizado, escolha Para personalizar o modelo, escolha Custom pattern (JSON editor) (Padrão personalizado, editor JSON) e crie seu padrão de evento.	

Para criar uma regra com base em…	Fazer isso...	
Programação	a. Em Rule type (Tipo de regra), escolha Schedule (Programação). b. Escolha Próximo. c. Em Schedule pattern (Padrão de programação), siga um destes procedimentos: • Para usar uma expressão cron para definir a programação, escolha A fine-grained schedule that runs at a specific time, such as 8:00 a.m. PST on the first Monday of every month (Uma programação refinada que é executada em um horário específico, como 8:00 PST na primeira segunda-feira de cada mês) e insira a expressão cron. • Para usar uma expressão de intervalo para definir a programação, escolha A schedule that runs at a regular rate, such as every 10 minutes (Uma programação que é executada a um	

Para criar uma regra com base em…	Fazer isso...	
	intervalo regular, como a cada 10 minutos) e insira a expressão de intervalo.	

7. Escolha Next (Avançar).
8. Em Tipos de destino, escolha Serviço da AWS.
9. Para Select a target (Selecionar um destino), escolha Systems Manager Automation (Automation do Systems Manager).
10. Para Document (Documento), escolha um runbook a ser usado quando o destino for invocado.
11. Na seção Configure automation parameter(s) (Configurar parâmetros de automação), mantenha os valores dos parâmetros padrão (se disponíveis) ou insira seus próprios valores.

> **Note**
>
> Para criar um destino, você deve especificar um valor para cada parâmetro obrigatório. Se não fizer isso, o sistema criará a regra, mas a regra não será executada.

12. Para muitos tipos de destino, o Eventbridge precisa de permissões para enviar eventos ao destino. Nesses casos, o Eventbridge pode criar o perfil do IAM necessário para o perfil ser executado. Execute um destes procedimentos:

    - Para criar um perfil do IAM automaticamente, escolha Criar um novo perfil para este recurso específico.
    - Para usar um perfil do IAM que você criou anteriormente, escolha Use existing role (Usar função existente). Observe que talvez seja necessário atualizar a política de confiança do seu perfil do IAM para incluir o EventBridge. Veja um exemplo a seguir:

    JSON

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

```
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "events.amazonaws.com",
                    "ssm.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

13. Escolha Próximo.
14. (Opcional) Insira uma ou mais tags para a regra. Para obter mais informações, consulte Marcar recursos do Amazon EventBridge no Guia do usuário do Amazon EventBridge.
15. Escolha Próximo.
16. Analise os detalhes da regra e selecione Criar regra.

Crie um evento do EventBridge que use um runbook (linha de comando)

O procedimento a seguir descreve como usar a AWS CLI (no Linux ou no Windows) ou o Ferramentas da AWS para PowerShell para criar uma regra de evento do EventBridge e configurar um runbook como o destino.

Para configurar um runbook como destino de uma regra de evento do EventBridge

1. Instale e configure a AWS CLI ou o Ferramentas da AWS para PowerShell, caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI e Instalar o Ferramentas da AWS para PowerShell.
2. Crie um comando para especificar uma nova regra de evento do EventBridge. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   Triggers com base em uma programação

   Linux & macOS

   ```
   aws events put-rule \
   ```

```
--name "rule name" \
--schedule-expression "cron or rate expression"
```

Windows

```
aws events put-rule ^
--name "rule name" ^
--schedule-expression "cron or rate expression"
```

PowerShell

```
Write-CWERule `
-Name "rule name" `
-ScheduleExpression "cron or rate expression"
```

O exemplo a seguir cria uma regra de evento do EventBridge que é iniciada todos os dias às 9h (UTC).

Linux & macOS

```
aws events put-rule \
--name "DailyAutomationRule" \
--schedule-expression "cron(0 9 * * ? *)"
```

Windows

```
aws events put-rule ^
--name "DailyAutomationRule" ^
--schedule-expression "cron(0 9 * * ? *)"
```

PowerShell

```
Write-CWERule `
-Name "DailyAutomationRule" `
-ScheduleExpression "cron(0 9 * * ? *)"
```

Dispara com base em um evento

Linux & macOS

```
aws events put-rule \
--name "rule name" \
--event-pattern "{\"source\":[\"aws.service\"],\"detail-type\":[\"service event
 detail type\"]}"
```

Windows

```
aws events put-rule ^
--name "rule name" ^
--event-pattern "{\"source\":[\"aws.service\"],\"detail-type\":[\"service event
 detail type\"]}"
```

PowerShell

```
Write-CWERule `
-Name "rule name" `
-EventPattern '{"source":["aws.service"],"detail-type":["service event detail
 type"]}'
```

O exemplo a seguir cria uma regra de evento do EventBridge que é iniciada quando o estado de qualquer instância do EC2 na região é alterado.

Linux & macOS

```
aws events put-rule \
--name "EC2InstanceStateChanges" \
--event-pattern "{\"source\":[\"aws.ec2\"],\"detail-type\":[\"EC2 Instance
 State-change Notification\"]}"
```

Windows

```
aws events put-rule ^
--name "EC2InstanceStateChanges" ^
--event-pattern "{\"source\":[\"aws.ec2\"],\"detail-type\":[\"EC2 Instance
 State-change Notification\"]}"
```

PowerShell

```
Write-CWERule `
-Name "EC2InstanceStateChanges" `
-EventPattern '{"source":["aws.ec2"],"detail-type":["EC2 Instance State-change
 Notification"]}'
```

O comando retorna detalhes da nova regra do EventBridge, semelhantes aos seguintes:

Linux & macOS

```
{
"RuleArn": "arn:aws:events:us-east-1:123456789012:rule/automationrule"
}
```

Windows

```
{
"RuleArn": "arn:aws:events:us-east-1:123456789012:rule/automationrule"
}
```

PowerShell

```
arn:aws:events:us-east-1:123456789012:rule/EC2InstanceStateChanges
```

3. Crie um comando para especificar um runbook como o destino da regra de evento do EventBridge que você criou na etapa 2. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   Linux & macOS

   ```
   aws events put-targets \
   --rule rule name \
   --targets '{"Arn": " arn:aws:ssm:region:account ID:automation-definition/runbook
    name","Input":"{\"Message\":[\"{\\\"Key\\\":\\\"key name\\\",\\\"Values\\\":[\\
   \"value\\\"]}\"]}","Id": "target ID","RoleArn": "arn:aws:iam::123456789012:role/
   service-role/EventBridge service role"}'
   ```

Windows

```
aws events put-targets ^
--rule rule name ^
--targets '{"Arn": "arn:aws:ssm:region:account ID:automation-definition/runbook
 name","Input":"{\"Message\":[\"{\\\"Key\\\":\\\"key name\\\",\\\"Values\\\":[\\
\"value\\\"]}\"]}","Id": "target ID","RoleArn": "arn:aws:iam::123456789012:role/
service-role/EventBridge service role"}'
```

PowerShell

```
$Target = New-Object Amazon.CloudWatchEvents.Model.Target
$Target.Id = "target ID"
$Target.Arn = "arn:aws:ssm:region:account ID:automation-definition/runbook name"
$Target.RoleArn = "arn:aws:iam::123456789012:role/service-role/EventBridge
 service role"
$Target.Input = '{"input parameter":["value"],"AutomationAssumeRole":
["arn:aws:iam::123456789012:role/AutomationServiceRole"]}'

Write-CWETarget `
-Rule "rule name" `
-Target $Target
```

O exemplo a seguir cria um destino para um evento do EventBridge que inicia o ID de instância especificado usando o runbook `AWS-StartEC2Instance`.

Linux & macOS

```
aws events put-targets \
--rule DailyAutomationRule \
--targets '{"Arn": "arn:aws:ssm:region:*:automation-definition/AWS-
StartEC2Instance","Input":"{\"InstanceId\":[\"i-02573cafcfEXAMPLE\"],
\"AutomationAssumeRole\":[\"arn:aws:iam::123456789012:role/AutomationServiceRole
\"]}","Id": "Target1","RoleArn": "arn:aws:iam::123456789012:role/service-role/
AWS_Events_Invoke_Start_Automation_Execution_1213609520"}'
```

Windows

```
aws events put-targets ^
--rule DailyAutomationRule ^
```

```
--targets '{"Arn": "arn:aws:ssm:region:*:automation-definition/AWS-
StartEC2Instance","Input":"{\"InstanceId\":[\"i-02573cafcfEXAMPLE\"],
\"AutomationAssumeRole\":[\"arn:aws:iam::123456789012:role/AutomationServiceRole
\"]}","Id": "Target1","RoleArn": "arn:aws:iam::123456789012:role/service-role/
AWS_Events_Invoke_Start_Automation_Execution_1213609520"}'
```

PowerShell

```
$Target = New-Object Amazon.CloudWatchEvents.Model.Target
$Target.Id = "Target1"
$Target.Arn = "arn:aws:ssm:region:*:automation-definition/AWS-StartEC2Instance"
$Target.RoleArn = "arn:aws:iam::123456789012:role/service-role/
AWS_Events_Invoke_Start_Automation_Execution_1213609520"
$Target.Input = '{"InstanceId":["i-02573cafcfEXAMPLE"],"AutomationAssumeRole":
["arn:aws:iam::123456789012:role/AutomationServiceRole"]}'

Write-CWETarget `
-Rule "DailyAutomationRule" `
-Target $Target
```

O sistema retorna informações como estas.

Linux & macOS

```
{
"FailedEntries": [],
"FailedEntryCount": 0
}
```

Windows

```
{
"FailedEntries": [],
"FailedEntryCount": 0
}
```

PowerShell

Não haverá saída se o comando for bem-sucedido para PowerShell.

Executar uma automação passo a passo

Os procedimentos a seguir descrevem como usar o console do AWS Systems Manager e a AWS Command Line Interface (AWS CLI) para executar uma automação usando o modo de execução manual. Usando o modo de execução manual, a automação é iniciada em um status Waiting (Aguardando) e faz uma pausa no status Waiting (Aguardando) entre cada etapa. Isso permite que você controle quando automação deve prosseguir, o que é útil quando for necessário revisar o resultado de uma etapa antes de continuar.

A automação é executada no contexto do usuário atual. Isso significa que você não precisa configurar outras permissões do IAM, desde que tenha permissão para usar o runbook e qualquer ação chamada pelo runbook. Se você tiver permissões de administrador no IAM, isso significa que você já tem permissão para executar essa automação.

Executar uma automação passo a passo (console)

O procedimento a seguir mostra como usar o console do Systems Manager para executar manualmente uma automação passo a passo.

Para executar uma automação passo a passo

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, selecione Automation e Execute automation (Executar automação).
3. Na lista Automation document (Documento do Automation), escolha um runbook. Escolha uma ou mais opções no painel Document categories (Categorias de documentos) para filtrar documentos SSM de acordo com sua finalidade. Para visualizar um runbook que você tenha, escolha a guia Owned by me (De minha propriedade). Para visualizar um runbook compartilhado com sua conta, escolha a guia Shared with me (Compartilhado comigo). Para visualizar todos os runbooks, escolha a guia All documents (Todos os documentos).

   > **Note**
   >
   > Você pode visualizar informações sobre um runbook, selecionando o nome dele.

4. Na seção Document details (Detalhes do documento), verifique se Document version (Versão do documento) está definida como a versão que você quer executar. O sistema inclui as seguintes opções de versão:

- Versão padrão no runtime: escolha essa opção se o runbook do Automation for atualizado periodicamente e uma nova versão padrão for atribuída.
- Versão mais recente no runtime: escolha essa opção se o runbook do Automation for atualizado periodicamente e se você quiser executar a versão mais recente.
- 1 (padrão): escolha esta opção para executar a primeira versão do documento, que é a versão padrão.

5. Escolha Próximo.
6. Na seção Execution mode (Modo de execução), escolha Manual execution (Execução manual).
7. Na seção Input parameters (Parâmetros de entrada), especifique as entradas necessárias. Opcionalmente, você pode escolher uma função de serviço do IAM na lista AutomationAssumeRole.
8. Clique em Executar.
9. Selecione Execute this step (Execute esta etapa) quando estiver pronto para iniciar a primeira etapa da automação. A automação prossegue com a etapa 1 e pausa antes de executar qualquer etapa subsequente especificada no runbook que você escolheu na etapa 3 deste procedimento. Se o runbook tiver várias etapas, você deverá selecionar Execute this step (Executar esta etapa) em cada etapa para que a automação prossiga. Cada vez que você escolher Execute this step (Executar esta etapa) a ação será executada.

   > **Note**
   >
   > O console exibe o status da automação. Se a automação não executar uma etapa, consulte Solução de problemas do Systems Manager Automation.

10. Depois de concluir todas as etapas especificadas no runbook, escolha Complete and view results (Concluir e visualizar os resultados) para concluir a automação e visualizar os resultados.

Executar uma automação passo a passo (linha de comando)

O procedimento a seguir descreve como usar a AWS CLI (no Linux, macOS ou no Windows) ou o Ferramentas da AWS para PowerShell para executar manualmente uma automação passo a passo.

Para executar uma automação passo a passo

1. Instale e configure a AWS CLI ou o Ferramentas da AWS para PowerShell, caso ainda não o tenha feito.

Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI e Instalar o Ferramentas da AWS para PowerShell.

2. Execute o comando a seguir para iniciar uma automação manual. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   Linux & macOS

   ```
   aws ssm start-automation-execution \
       --document-name runbook name \
       --mode Interactive \
       --parameters runbook parameters
   ```

   Windows

   ```
   aws ssm start-automation-execution ^
       --document-name runbook name ^
       --mode Interactive ^
       --parameters runbook parameters
   ```

   PowerShell

   ```
   Start-SSMAutomationExecution `
       -DocumentName runbook name `
       -Mode Interactive `
       -Parameter runbook parameters
   ```

   Este é um exemplo que usa o runbook `AWS-RestartEC2Instance` para reiniciar a instância do EC2 especificada.

   Linux & macOS

   ```
   aws ssm start-automation-execution \
       --document-name "AWS-RestartEC2Instance" \
       --mode Interactive \
       --parameters "InstanceId=i-02573cafcfEXAMPLE"
   ```

Windows

```
aws ssm start-automation-execution ^
    --document-name "AWS-RestartEC2Instance" ^
    --mode Interactive ^
    --parameters "InstanceId=i-02573cafcfEXAMPLE"
```

PowerShell

```
Start-SSMAutomationExecution `
    -DocumentName AWS-RestartEC2Instance `
    -Mode Interactive
    -Parameter @{"InstanceId"="i-02573cafcfEXAMPLE"}
```

O sistema retorna informações como estas.

Linux & macOS

```
{
    "AutomationExecutionId": "ba9cd881-1b36-4d31-a698-0123456789ab"
}
```

Windows

```
{
    "AutomationExecutionId": "ba9cd881-1b36-4d31-a698-0123456789ab"
}
```

PowerShell

```
ba9cd881-1b36-4d31-a698-0123456789ab
```

3. Execute o comando a seguir quando estiver pronto para iniciar a primeira etapa da automação. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações. A automação prossegue com a etapa 1 e pausa antes de executar qualquer etapa subsequente especificada no runbook que você escolheu na etapa 1 deste procedimento. Se o runbook tiver várias etapas, você deverá executar o seguinte comando para cada etapa, para que a automação prossiga.

Linux & macOS

```
aws ssm send-automation-signal \
    --automation-execution-id ba9cd881-1b36-4d31-a698-0123456789ab \
    --signal-type StartStep \
    --payload StepName="stopInstances"
```

Windows

```
aws ssm send-automation-signal ^
    --automation-execution-id ba9cd881-1b36-4d31-a698-0123456789ab ^
    --signal-type StartStep ^
    --payload StepName="stopInstances"
```

PowerShell

```
Send-SSMAutomationSignal `
    -AutomationExecutionId ba9cd881-1b36-4d31-a698-0123456789ab `
    -SignalType StartStep
    -Payload @{"StepName"="stopInstances"}
```

Não haverá saída se o comando for bem-sucedido.

4. Execute o comando a seguir para recuperar o status da execução de cada etapa na automação.

Linux & macOS

```
aws ssm describe-automation-step-executions \
    --automation-execution-id ba9cd881-1b36-4d31-a698-0123456789ab
```

Windows

```
aws ssm describe-automation-step-executions ^
    --automation-execution-id ba9cd881-1b36-4d31-a698-0123456789ab
```

PowerShell

```
Get-SSMAutomationStepExecution `
    -AutomationExecutionId ba9cd881-1b36-4d31-a698-0123456789ab
```

O sistema retorna informações como estas.

Linux & macOS

```
{
    "StepExecutions": [
        {
            "StepName": "stopInstances",
            "Action": "aws:changeInstanceState",
            "ExecutionStartTime": 1557167178.42,
            "ExecutionEndTime": 1557167220.617,
            "StepStatus": "Success",
            "Inputs": {
                "DesiredState": "\"stopped\"",
                "InstanceIds": "[\"i-02573cafcfEXAMPLE\"]"
            },
            "Outputs": {
                "InstanceStates": [
                    "stopped"
                ]
            },
            "StepExecutionId": "654243ba-71e3-4771-b04f-0123456789ab",
            "OverriddenParameters": {},
            "ValidNextSteps": [
                "startInstances"
            ]
        },
        {
            "StepName": "startInstances",
            "Action": "aws:changeInstanceState",
            "ExecutionStartTime": 1557167273.754,
            "ExecutionEndTime": 1557167480.73,
            "StepStatus": "Success",
            "Inputs": {
                "DesiredState": "\"running\"",
                "InstanceIds": "[\"i-02573cafcfEXAMPLE\"]"
            },
            "Outputs": {
                "InstanceStates": [
                    "running"
                ]
            },
```

```
            "StepExecutionId": "8a4a1e0d-dc3e-4039-a599-0123456789ab",
            "OverriddenParameters": {}
        }
    ]
}
```

Windows

```
{
    "StepExecutions": [
        {
            "StepName": "stopInstances",
            "Action": "aws:changeInstanceState",
            "ExecutionStartTime": 1557167178.42,
            "ExecutionEndTime": 1557167220.617,
            "StepStatus": "Success",
            "Inputs": {
                "DesiredState": "\"stopped\"",
                "InstanceIds": "[\"i-02573cafcfEXAMPLE\"]"
            },
            "Outputs": {
                "InstanceStates": [
                    "stopped"
                ]
            },
            "StepExecutionId": "654243ba-71e3-4771-b04f-0123456789ab",
            "OverriddenParameters": {},
            "ValidNextSteps": [
                "startInstances"
            ]
        },
        {
            "StepName": "startInstances",
            "Action": "aws:changeInstanceState",
            "ExecutionStartTime": 1557167273.754,
            "ExecutionEndTime": 1557167480.73,
            "StepStatus": "Success",
            "Inputs": {
                "DesiredState": "\"running\"",
                "InstanceIds": "[\"i-02573cafcfEXAMPLE\"]"
            },
            "Outputs": {
                "InstanceStates": [
```

```
                    "running"
                ]
            },
            "StepExecutionId": "8a4a1e0d-dc3e-4039-a599-0123456789ab",
            "OverriddenParameters": {}
        }
    ]
}
```

PowerShell

```
Action: aws:changeInstanceState
ExecutionEndTime     : 5/6/2019 19:45:46
ExecutionStartTime   : 5/6/2019 19:45:03
FailureDetails       :
FailureMessage       :
Inputs               : {[DesiredState, "stopped"], [InstanceIds,
 ["i-02573cafcfEXAMPLE"]]}
IsCritical           : False
IsEnd                : False
MaxAttempts          : 0
NextStep             :
OnFailure            :
Outputs              : {[InstanceStates,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
OverriddenParameters : {}
Response             :
ResponseCode         :
StepExecutionId      : 8fcc9641-24b7-40b3-a9be-0123456789ab
StepName             : stopInstances
StepStatus           : Success
TimeoutSeconds       : 0
ValidNextSteps       : {startInstances}
```

5. Execute o comando a seguir para concluir a automação depois que todas as etapas especificadas no runbook escolhido tiverem sido concluídas. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   Linux & macOS

```
aws ssm stop-automation-execution \
    --automation-execution-id ba9cd881-1b36-4d31-a698-0123456789ab \
```

```
    --type Complete
```

Windows

```
aws ssm stop-automation-execution ^
    --automation-execution-id ba9cd881-1b36-4d31-a698-0123456789ab ^
    --type Complete
```

PowerShell

```
Stop-SSMAutomationExecution `
    -AutomationExecutionId ba9cd881-1b36-4d31-a698-0123456789ab `
    -Type Complete
```

Não haverá saída se o comando for bem-sucedido.

Programação de automações com associações do State Manager

É possível iniciar uma automação criando uma associação do State Manager com um runbook. O State Manager é uma ferramenta do AWS Systems Manager. Ao criar uma associação do State Manager com um runbook, você pode direcionar diferentes tipos de recursos da AWS. Por exemplo, é possível criar associações que aplicam um estado desejado a um recurso da AWS, incluindo o seguinte:

- Anexe uma função do Systems Manager às instâncias do Amazon Elastic Compute Cloud (Amazon EC2) para torná-las instâncias gerenciadas.
- Aplicar regras de entrada e saída desejadas a um grupo de segurança.
- Crie ou exclua backups do Amazon DynamoDB.
- Crie ou exclua snapshots do Amazon Elastic Block Store (Amazon EBS).
- Desative as permissões de leitura e gravação em buckets do Amazon Simple Storage Service (Amazon S3).
- Inicie, reinicie ou interrompa instâncias gerenciadas e instâncias do Amazon Relational Database Service (Amazon RDS).
- Aplique patches às macOS do Linux, AMIs e Windows.

Use os procedimentos a seguir para criar uma associação do State Manager que executa uma automação usando o console do AWS Systems Manager e a AWS Command Line Interface (AWS CLI). Para obter informações gerais sobre associações e informações sobre como criar uma associação que usa um documento SSM `Command` ou um documento `Policy`, consulte Criar associações.

Antes de começar

Observe estes detalhes importantes, antes de executar uma automação usando o State Manager.

- Antes de criar uma associação que usa um runbook, verifique se você configurou permissões para o Automation, uma ferramenta do AWS Systems Manager. Para obter mais informações, consulte Configurar a automação.
- As associações do State Manager que executam runbooks contribuem para o número máximo de automações em execução simultânea na sua conta da Conta da AWS. É possível ter no máximo 100 automações em execução ao mesmo tempo. Para obter informações, consulte as cotas de serviço do Systems Manager no Referência geral da Amazon Web Services.
- Ao executar uma automação, o State Manager não registra em log as operações de API iniciadas pela automação em AWS CloudTrail.
- O Systems Manager cria automaticamente uma função vinculada ao serviço para que o State Manager tenha permissão para chamar operações de API do Systems Manager Automation. Se desejar, você pode criar a função vinculada ao serviço por conta própria executando o seguinte comando na AWS CLI ou no Ferramentas da AWS para PowerShell.

  Linux & macOS

  ```
  aws iam create-service-linked-role \
  --aws-service-name ssm.amazonaws.com
  ```

  Windows

  ```
  aws iam create-service-linked-role ^
  --aws-service-name ssm.amazonaws.com
  ```

  PowerShell

  ```
  New-IAMServiceLinkedRole `
  -AWSServiceName ssm.amazonaws.com
  ```

Para obter mais informações sobre funções vinculadas ao serviço, consulte Usar perfis vinculados a serviço do Systems Manager.

Criar uma associação que executa uma automação (console)

O procedimento a seguir descreve como usar o console do Systems Manager para criar uma associação do State Manager que executa uma automação.

Como criar uma associação do State Manager que execute uma automação

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha State Manager e selecione Create association.
3. No campo Name, especifique um nome. Isso é opcional, mas recomendado.
4. Na lista Document (Documento), escolha um runbook. Use a barra de pesquisa para filtrar os runboks Document type : Equal : Automation. Para visualizar mais runbooks, use os números à direita da barra de pesquisa.

   > **Note**
   >
   > Você pode visualizar informações sobre um runbook, selecionando o nome dele.

5. Escolha Simple execution (Execução simples) para executar a automação em um ou mais destinos, especificando o ID do recurso desses destinos. Escolha Rate control (Controle de taxa) para executar a automação em uma frota de recursos da AWS, especificando uma opção de direcionamento, como tags ou o AWS Resource Groups. Você também pode controlar a operação da automação nos seus recursos, especificando a simultaneidade e limites de erro.

   Se você escolheu Rate control (Controle de taxa), a seção Targets (Destinos) será exibida.
6. Na seção Targets (Destinos), escolha um método para definir recursos de direcionamento.

   a. (Necessário) Na lista Parameter (Parâmetro), escolha um parâmetro. Os itens na lista Parameter (Parâmetro) são determinados pelos parâmetros no runbook que você selecionou no início deste procedimento. Ao escolher um parâmetro, você define o tipo de recurso no qual a automação é executada.

   b. (Necessário) Na lista Targets (Destinos), escolha um método para definir destinos para os recursos.

- Resource Group (Grupo de recursos): escolha o nome do grupo na lista Resource Group (Grupo de recursos). Para obter mais informações sobre o direcionamento do AWS Resource Groups em runbooks, consulte AWS Resource Groups como destino.
- Tags: insira a chave de tag e (opcionalmente) o valor da tag nos campos fornecidos. Escolha Adicionar. Para obter mais informações sobre o direcionamento de tags em runbooks, consulte Definir uma etiqueta como destino.
- Parameter Values (Valores de parâmetros): insira valores na seção Input parameters (Parâmetros de entrada). Se você especificar vários valores, o Systems Manager executará uma automação secundária em cada valor especificado.

  Por exemplo, digamos que o runbook inclui um parâmetro InstanceID. Se você direcionar os valores do parâmetro InstanceID ao executar a automação, o Systems Manager executará uma automação filho para cada valor especificado para o ID da instância. A automação pai é concluída quando a automação conclui a execução de cada instância especificada, ou se a automação falha. Você pode direcionar um máximo de 50 valores de parâmetro. Para obter mais informações sobre direcionar valores de parâmetros em runbooks, consulte Direcionar valores de parâmetro.

7. Na seção Input parameters (Parâmetros de entrada), especifique os parâmetros de entrada necessários.

   Se você optar por definir o destino dos recursos usando tags ou um grupo de recursos, talvez não precise escolher algumas das opções na seção Input parameters (Parâmetros de entrada). Por exemplo, se você tiver escolhido o runbook `AWS-RestartEC2Instance`, e definiu o destino das instâncias usando tags, você não precisará especificar ou escolher IDs de instância na seção Input parameters (Parâmetros de entrada). A automação localiza as instâncias para reiniciar usando as tags que você especificou.

   > **Important**
   >
   > Especifique um ARN de função no campo AutomationAssumeRole. O State Manager assume a função para chamar Serviços da AWS especificados no runbook e para executar associações do Automation em seu nome.

8. Na seção Specify schedule (Especificar programação), escolha On Schedule (Na programação) se quiser executar a associação em intervalos regulares. Se você escolher essa opção, use as opções fornecidas para criar a programação usando expressões Cron ou Rate. Para obter mais

informações sobre expressões Cron e Rate para o State Manager, consulte Expressões cron e rate para associações.

> **Note**
>
> Expressões de taxa são o mecanismo preferencial de programação para associações do State Manager que usam runbooks. Essas expressões proporcionam mais flexibilidade para a execução de associações, caso você atinja o número máximo de automações executadas ao mesmo tempo. Com uma programação de taxas, o Systems Manager pode tentar a automação novamente logo depois de receber a notificação de que as automações simultâneas atingiram seu limite máximo e foram controladas.

Escolha No schedule (Sem programação) se quiser executar a associação apenas uma vez.

9. (Opcional) Na seção Rate Control (Controle de taxa), escolha as opções Concurrency (Simultaneidade) e Error threshold (Limiar de erros) para controlar a implantação da automação nos seus recursos da AWS.

   a. Na seção Concurrency (Simultaneidade), escolha uma opção:

      - Escolha targets (destinos) para inserir um número absoluto de destinos que podem executar a automação simultaneamente.
      - Escolha percentage (porcentagem) para inserir uma porcentagem do conjunto de destino que pode executar a automação simultaneamente.

   b. Na seção Error threshold (Limite de erro), escolha uma opção:

      - Escolha errors (erros) para inserir um número absoluto de erros permitidos antes que o Automation pare de enviar a automação para outros recursos.
      - Escolha percent (por cento) para inserir uma porcentagem de erros permitidos antes que o Automation pare de enviar a automação para outros recursos.

   Para obter mais informações sobre o uso de destinos e controles de taxa com a Automação, consulte Executar operações automatizadas em escala.

10. Escolha Create Association (Criar associação).

> **Important**
>
> Quando você cria uma associação, ela é executada imediatamente nos destinos especificados. Em seguida, a associação é executada com base na expressão Cron ou Rate que você escolheu. Se você escolheu No schedule (Sem programação), a associação não será executada novamente.

Criar uma associação que executa uma automação (linha de comando)

O procedimento a seguir descreve como usar a AWS CLI (no Linux ou no Windows) ou o Ferramentas da AWS para PowerShell para criar uma associação do State Manager que executa uma automação.

Antes de começar

Antes de concluir o procedimento a seguir, certifique-se de ter criado um perfil de serviço do IAM com as permissões necessárias para executar o runbook e configurado uma relação de confiança para o Automation, uma ferramenta do AWS Systems Manager. Para obter mais informações, consulte Tarefa 1: Criar uma função de serviço para a automação.

Para criar uma associação que executa uma automação

1. Instale e configure a AWS CLI ou o Ferramentas da AWS para PowerShell, caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI e Instalar o Ferramentas da AWS para PowerShell.

2. Execute o comando a seguir para visualizar uma lista de documentos.

   Linux & macOS

   ```
   aws ssm list-documents
   ```

   Windows

   ```
   aws ssm list-documents
   ```

PowerShell

```
Get-SSMDocumentList
```

Anote o nome do runbook que você quer usar para a associação.

3. Execute o comando a seguir para visualizar os detalhes sobre o runbook. No comando a seguir, substitua *runbook name* por suas próprias informações.

   Linux & macOS

   ```
   aws ssm describe-document \
   --name runbook name
   ```

   Anote o nome de um parâmetro (por exemplo, `InstanceId`) que você deseja usar para a opção `--automation-target-parameter-name`. Esse parâmetro determina o tipo de recurso em que a automação é executada.

   Windows

   ```
   aws ssm describe-document ^
   --name runbook name
   ```

   Anote o nome de um parâmetro (por exemplo, `InstanceId`) que você deseja usar para a opção `--automation-target-parameter-name`. Esse parâmetro determina o tipo de recurso em que a automação é executada.

   PowerShell

   ```
   Get-SSMDocumentDescription `
   -Name runbook name
   ```

   Anote o nome de um parâmetro (por exemplo, `InstanceId`) que você deseja usar para a opção `AutomationTargetParameterName`. Esse parâmetro determina o tipo de recurso em que a automação é executada.

4. Crie um comando que execute uma automação usando uma associação do State Manager. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

Definir destino usando tags

Linux & macOS

```
aws ssm create-association \
--association-name association name \
--targets Key=tag:key name,Values=value \
--name runbook name \
--parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/
RunbookAssumeRole \
--automation-target-parameter-name target parameter \
--schedule "cron or rate expression"
```

> **Note**
>
> Se você criar uma associação usando o AWS CLI, use o parâmetro `--targets` para as instâncias de destino da associação. Não use o parâmetro `--instance-id`. O parâmetro `--instance-id` é um parâmetro legado.

Windows

```
aws ssm create-association ^
--association-name association name ^
--targets Key=tag:key name,Values=value ^
--name runbook name ^
--parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/
RunbookAssumeRole ^
--automation-target-parameter-name target parameter ^
--schedule "cron or rate expression"
```

> **Note**
>
> Se você criar uma associação usando o AWS CLI, use o parâmetro `--targets` para as instâncias de destino da associação. Não use o parâmetro `--instance-id`. O parâmetro `--instance-id` é um parâmetro legado.

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
$Targets.Key = "tag:key name"
$Targets.Values = "value"

New-SSMAssociation `
-AssociationName "association name" `
-Target $Targets `
-Name "runbook name" `
-Parameters @{
"AutomationAssumeRole"="arn:aws:iam::123456789012:role/RunbookAssumeRole" } `
-AutomationTargetParameterName "target parameter" `
-ScheduleExpression "cron or rate expression"
```

> **Note**
>
> Se você criar uma associação usando o Ferramentas da AWS para PowerShell, use o parâmetro `Target` para as instâncias de destino da associação. Não use o parâmetro `InstanceId`. O parâmetro `InstanceId` é um parâmetro legado.

Definir destino usando valores de parâmetro

Linux & macOS

```
aws ssm create-association \
--association-name association name \
--targets Key=ParameterValues,Values=value,value 2,value 3 \
--name runbook name \
--parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/
RunbookAssumeRole \
--automation-target-parameter-name target parameter \
--schedule "cron or rate expression"
```

Windows

```
aws ssm create-association ^
--association-name association name ^
```

```
--targets Key=ParameterValues,Values=value,value 2,value 3 ^
--name runbook name ^
--parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/
RunbookAssumeRole ^
--automation-target-parameter-name target parameter ^
--schedule "cron or rate expression"
```

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
$Targets.Key = "ParameterValues"
$Targets.Values = "value","value 2","value 3"

New-SSMAssociation `
-AssociationName "association name" `
-Target $Targets `
-Name "runbook name" `
-Parameters @{
"AutomationAssumeRole"="arn:aws:iam::123456789012:role/RunbookAssumeRole"} `
-AutomationTargetParameterName "target parameter" `
-ScheduleExpression "cron or rate expression"
```

Definir destino usando AWS Resource Groups

Linux & macOS

```
aws ssm create-association \
--association-name association name \
--targets Key=ResourceGroup,Values=resource group name \
--name runbook name \
--parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/
RunbookAssumeRole \
--automation-target-parameter-name target parameter \
--schedule "cron or rate expression"
```

Windows

```
aws ssm create-association ^
--association-name association name ^
--targets Key=ResourceGroup,Values=resource group name ^
--name runbook name ^
```

```
--parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/
RunbookAssumeRole ^
--automation-target-parameter-name target parameter ^
--schedule "cron or rate expression"
```

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
$Targets.Key = "ResourceGroup"
$Targets.Values = "resource group name"

New-SSMAssociation `
-AssociationName "association name" `
-Target $Targets `
-Name "runbook name" `
-Parameters @{
"AutomationAssumeRole"="arn:aws:iam::123456789012:role/RunbookAssumeRole"} `
-AutomationTargetParameterName "target parameter" `
-ScheduleExpression "cron or rate expression"
```

Selecionar várias contas e regiões como destino

Linux & macOS

```
aws ssm create-association \
--association-name association name \
--targets Key=ResourceGroup,Values=resource group name \
--name runbook name \
--parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/
RunbookAssumeRole \
--automation-target-parameter-name target parameter \
--schedule "cron or rate expression" \
--target-locations
 Accounts=111122223333,444455556666,444455556666,Regions=region,region
```

Windows

```
aws ssm create-association ^
--association-name association name ^
--targets Key=ResourceGroup,Values=resource group name ^
--name runbook name ^
```

```
--parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/
RunbookAssumeRole ^
--automation-target-parameter-name target parameter ^
--schedule "cron or rate expression" ^
--target-locations
 Accounts=111122223333,444455556666,444455556666,Regions=region,region
```

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
$Targets.Key = "ResourceGroup"
$Targets.Values = "resource group name"

New-SSMAssociation `
-AssociationName "association name" `
-Target $Targets `
-Name "runbook name" `
-Parameters @{
"AutomationAssumeRole"="arn:aws:iam::123456789012:role/RunbookAssumeRole"} `
-AutomationTargetParameterName "target parameter" `
-ScheduleExpression "cron or rate expression" `
-TargetLocations @{
    "Accounts"=["111122223333,444455556666,444455556666"],
    "Regions"=["region,region"]
```

O comando retorna detalhes da nova associação semelhantes ao seguinte.

Linux & macOS

```
{
"AssociationDescription": {
    "ScheduleExpression": "cron(0 7 ? * MON *)",
    "Name": "AWS-StartEC2Instance",
    "Parameters": {
        "AutomationAssumeRole": [
            "arn:aws:iam::123456789012:role/RunbookAssumeRole"
        ]
    },
    "Overview": {
        "Status": "Pending",
        "DetailedStatus": "Creating"
    },
```

```
    "AssociationId": "1450b4b7-bea2-4e4b-b340-01234EXAMPLE",
    "DocumentVersion": "$DEFAULT",
    "AutomationTargetParameterName": "InstanceId",
    "LastUpdateAssociationDate": 1564686638.498,
    "Date": 1564686638.498,
    "AssociationVersion": "1",
    "AssociationName": "CLI",
    "Targets": [
        {
            "Values": [
                "DEV"
            ],
            "Key": "tag:ENV"
        }
    ]
}
}
```

Windows

```
{
"AssociationDescription": {
    "ScheduleExpression": "cron(0 7 ? * MON *)",
    "Name": "AWS-StartEC2Instance",
    "Parameters": {
        "AutomationAssumeRole": [
            "arn:aws:iam::123456789012:role/RunbookAssumeRole"
        ]
    },
    "Overview": {
        "Status": "Pending",
        "DetailedStatus": "Creating"
    },
    "AssociationId": "1450b4b7-bea2-4e4b-b340-01234EXAMPLE",
    "DocumentVersion": "$DEFAULT",
    "AutomationTargetParameterName": "InstanceId",
    "LastUpdateAssociationDate": 1564686638.498,
    "Date": 1564686638.498,
    "AssociationVersion": "1",
    "AssociationName": "CLI",
    "Targets": [
        {
            "Values": [
```

```
                "DEV"
            ],
            "Key": "tag:ENV"
        }
    ]
}
}
```

PowerShell

```
Name                 : AWS-StartEC2Instance
InstanceId           :
Date                 : 8/1/2019 7:31:38 PM
Status.Name          :
Status.Date          :
Status.Message       :
Status.AdditionalInfo :
```

> **Note**
>
> Se você usar tags para criar uma associação em uma ou mais instâncias de destino e, em seguida, remover as tags de uma instância, essa instância não executará mais a associação. A instância será dissociada do documento do State Manager.

Automações de solução de problemas executadas por associações do State Manager

O Systems Manager Automation impõe um limite de 100 automações simultâneas e 1.000 automações em fila por conta e região. Se uma associação do State Manager que executa um runbook para mostrar um status Failed (Falha) e um status detalhado de AutomationExecutionLimitExceeded, isso significa que a automação pode ter atingido o limite. Como resultado, o Systems Manager controlará as automações. Para resolver esse problema, faça o seguinte:

- Use uma expressão Cron ou Rate diferente para a sua associação. Por exemplo, se a associação estiver programada para execução a cada 30 minutos, altere-a para que ela seja executada a cada uma ou duas horas.
- Exclua as automações existentes com um status Pending (Pendente). Ao excluir essas automações, você limpa a fila atual.

Agendar automações com janelas de manutenção

Você pode iniciar uma automação configurando um runbook como uma tarefa registrada para uma janela de manutenção. Com o registro do runbook como uma tarefa registrada, a janela de manutenção executa a automação durante o período de manutenção programada.

Por exemplo, digamos que você crie um runbook chamado `CreateAMI` que cria uma Amazon Machine Image (AMI) de instâncias registradas como destinos para a janela de manutenção. Para especificar o runbook `CreateAMI` (e a automação correspondente) como uma tarefa registrada de uma janela de manutenção, primeiro você deve criar uma janela de manutenção e registrar destinos. Depois, use o procedimento a seguir para especificar o documento `CreateAMI` como uma tarefa registrada na janela de manutenção. Quando a janela de manutenção for iniciada durante o período programado, o sistema executará a automação e criará uma AMI dos destinos registrados.

Para obter informações sobre como criar runbooks do Automation, consulte Criação dos seus próprios runbooks. O Automation é uma ferramenta do AWS Systems Manager.

Use os procedimentos a seguir para configurar uma automação como uma tarefa registrada para uma janela de manutenção, usando o console do AWS Systems Manager, a AWS Command Line Interface (AWS CLI) ou o AWS Tools for Windows PowerShell.

Registrar uma tarefa de automação em uma janela de manutenção (console)

O procedimento a seguir descreve como usar o console do Systems Manager para configurar uma automação como uma tarefa registrada para uma janela de manutenção.

Antes de começar

Antes de concluir o procedimento a seguir, você deve criar uma janela de manutenção e registrar pelo menos um destino. Para obter mais informações, consulte os procedimentos a seguir:

- Crie uma janela de manutenção usando o console.
- Atribua destinos a uma janela de manutenção usando o console

Para configurar uma automação como uma tarefa registrada para uma janela de manutenção

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação à esquerda, selecione Maintenance Windows e depois selecione a janela de manutenção com a qual você deseja registrar uma tarefa Automação.

3. Escolha Ações. Em seguida, escolha Register Automation task (Registrar tarefa do Automation para executar sua opção de automação nos destinos, usando um runbook.
4. Em Name (Nome), insira um nome para a tarefa.
5. Em Descrição, insira uma descrição.
6. Em Document (Documento), escolha o runbook que define as tarefas a serem executadas.
7. Em Document version (Versão do documento), escolha a versão do runbook a ser usada.
8. Em Task priority (Prioridade de tarefas), escolha uma prioridade. 1 é a prioridade mais alta. As tarefas em uma janela de manutenção são programadas em ordem de prioridade, as tarefas que têm a mesma prioridade são programadas em paralelo.
9. Na seção Targets (Destinos), se o runbook selecionado executar tarefas nos recursos, identifique os destinos nos quais você deseja executar essa automação, especificando tags ou selecionando instâncias manualmente.

   > **Note**
   >
   > Se você quiser passar os recursos pelos parâmetros de entrada em vez de destinos, não é necessário especificar um destino de janela de manutenção.
   > Em muitos casos, você não precisa especificar explicitamente um destino para uma tarefa de automação. Por exemplo, digamos que você esteja criando uma tarefa do tipo Automation para atualizar uma Amazon Machine Image (AMI) para Linux, usando o runbook `AWS-UpdateLinuxAmi`. Quando a tarefa for executada, a AMI será atualizada com os pacotes de distribuição Linux e o software da Amazon mais recentes disponível. As novas instâncias criadas na AMI já têm essas atualizações instaladas. Como o ID da AMI a ser atualizado é especificado nos parâmetros de entrada para o runbook, não há necessidade de especificar um destino novamente na tarefa da janela de manutenção.

   Para obter informações sobre tarefas da janela de manutenção que não exigem destinos, consulte the section called “Registrar tarefas da janela de manutenção sem destinos”.
10. (Opcional) Em Rate control (Controle de taxa):

    > **Note**
    >
    > Se a tarefa que você estiver executando não especificar destinos, você não precisará especificar controles de taxa.

- Para Concurrency (Simultaneidade), especifique um número ou uma porcentagem de destinos nos quais executar a automação ao mesmo tempo.

  Se você selecionou destinos escolhendo pares de chave/valor de tags, e não tem certeza de quantos destinos usam as tags selecionadas, limite o número de automações que podem ser executadas ao mesmo tempo, especificando uma porcentagem.

  Quando a janela de manutenção é executada, uma nova execução da automação é iniciada por destino. Há um limite de 100 automações simultâneas por Conta da AWS. Se você especificar uma taxa de simultaneidade maior que 100, as automações simultâneas superiores a 100 serão automaticamente adicionadas à fila de automação. Para obter informações, consulte as cotas de serviço do Systems Manager no Referência geral da Amazon Web Services.
- Para Error threshold (Limite de erros), especifique quando interromper a execução da automação em outros destinos depois que ela falhar em um número ou em uma porcentagem de destinos. Por exemplo, se você especificar três erros, o Systems Manager deixará de executar as automação quando o quarto erro for recebido. Os destinos que ainda estiverem processando a automação também poderão enviar erros.

11. Na seção Input Parameters (Parâmetros de entrada), especifique os parâmetros para o runbook. Para runbooks, o sistema preenche automaticamente alguns dos valores. Você pode manter ou substituir esses valores.

> **Important**
>
> Para runbooks, existe a opção de especificar uma função assumida no Automation. Se você não especificar uma função para esse parâmetro, a automação assumirá a função de serviço da janela de manutenção que você escolher na etapa 11. Dessa forma, você deve garantir que a função de serviço da janela de manutenção escolhida tenha as permissões apropriadas do AWS Identity and Access Management (IAM) para realizar as ações definidas no runbook.
> Por exemplo, a função vinculada ao serviço para o Systems Manager não tem a permissão `ec2:CreateSnapshot` do IAM, que é necessária para executar o runbook `AWS-CopySnapshot`. Nesse cenário, você deve usar uma função de serviço da janela de manutenção personalizada ou especificar uma função de admissão de Automação

que tenha permissões `ec2:CreateSnapshot`. Para mais informações, consulte Configurar a automação.

12. Na área IAM service role (Perfil de serviço do IAM), escolha um perfil para fornecer permissões ao Systems Manager para iniciar a automação.

    Para criar um perfil de serviço para tarefas de janela de manutenção, consulte Configurar o Maintenance Windows.
13. Escolha Register Automation task (Registrar tarefa de Automação).

Registrar uma tarefa do Automation em uma janela de manutenção (linha de comando)

O procedimento a seguir descreve como usar a AWS CLI (no Linux ou no Windows) ou o Ferramentas da AWS para PowerShell para configurar uma automação como uma tarefa registrada para uma janela de manutenção.

Antes de começar

Antes de concluir o procedimento a seguir, você deve criar uma janela de manutenção e registrar pelo menos um destino. Para obter mais informações, consulte os procedimentos a seguir:

- Etapa 1: Criar a janela de manutenção usando a AWS CLI.
- Etapa 2: Registrar um nó de destino na janela de manutenção usando a AWS CLI

Para configurar uma automação como uma tarefa registrada para uma janela de manutenção

1. Instale e configure a AWS CLI ou o Ferramentas da AWS para PowerShell, caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI e Instalar o Ferramentas da AWS para PowerShell.
2. Crie um comando para configurar uma automação como uma tarefa registrada para uma janela de manutenção. Substitua cada *`espaço reservado para recurso de exemplo`* por suas próprias informações.

   Linux & macOS

```
aws ssm register-task-with-maintenance-window \
--window-id window ID \
```

```
--name task name \
--task-arn runbook name \
--targets Key=targets,Values=value \
--service-role-arn IAM role arn \
--task-type AUTOMATION \
--task-invocation-parameters task parameters \
--priority task priority \
--max-concurrency 10% \
--max-errors 5
```

> **Note**
>
> Se você configurar uma automação como uma tarefa registrada usando a AWS CLI, use o parâmetro `--Task-Invocation-Parameters` para especificar os parâmetros a serem passados para uma tarefa quando ela for executada. Não use o parâmetro `--Task-Parameters`. O parâmetro `--Task-Parameters` é um parâmetro legado.
>
> Para tarefas de janela de manutenção sem um destino especificado, você não pode fornecer valores para `--max-errors` e `--max-concurrency`. Em vez disso, o sistema insere um valor de espaço reservado de 1, que pode ser relatado na resposta a comandos como describe-maintenance-window-tasks e get-maintenance-window-task. Esses valores não afetam a execução da tarefa e podem ser ignorados. Para obter informações sobre tarefas da janela de manutenção que não exigem destinos, consulte Registrar tarefas da janela de manutenção sem destinos.

Windows

```
aws ssm register-task-with-maintenance-window ^
--window-id window ID ^
--name task name ^
--task-arn runbook name ^
--targets Key=targets,Values=value ^
--service-role-arn IAM role arn ^
--task-type AUTOMATION ^
--task-invocation-parameters task parameters ^
--priority task priority ^
--max-concurrency 10% ^
--max-errors 5
```

> **Note**
>
> Se você configurar uma automação como uma tarefa registrada usando a AWS CLI, use o parâmetro `--task-invocation-parameters` para especificar os parâmetros a serem passados para uma tarefa quando ela for executada. Não use o parâmetro `--task-parameters`. O parâmetro `--task-parameters` é um parâmetro legado.
>
> Para tarefas de janela de manutenção sem um destino especificado, você não pode fornecer valores para `--max-errors` e `--max-concurrency`. Em vez disso, o sistema insere um valor de espaço reservado de 1, que pode ser relatado na resposta a comandos como describe-maintenance-window-tasks e get-maintenance-window-task. Esses valores não afetam a execução da tarefa e podem ser ignorados. Para obter informações sobre tarefas da janela de manutenção que não exigem destinos, consulte Registrar tarefas da janela de manutenção sem destinos.

PowerShell

```
Register-SSMTaskWithMaintenanceWindow `
-WindowId window ID `
-Name "task name" `
-TaskArn "runbook name" `
-Target @{ Key="targets";Values="value" } `
-ServiceRoleArn "IAM role arn" `
-TaskType "AUTOMATION" `
-Automation_Parameter @{ "task parameter"="task parameter value"} `
-Priority task priority `
-MaxConcurrency 10% `
-MaxError 5
```

> **Note**
>
> Se você configurar uma automação como uma tarefa registrada usando o Ferramentas da AWS para PowerShell, use o parâmetro `-Automation_Parameter` para especificar os parâmetros a serem passados para uma tarefa quando ela for executada. Não use o parâmetro `-TaskParameters`. O parâmetro `-TaskParameters` é um parâmetro legado.

Para tarefas de janela de manutenção sem um destino especificado, você não pode fornecer valores para `-MaxError` e `-MaxConcurrency`. Em vez disso, o sistema insere um valor de espaço reservado de 1, que pode ser relatado na resposta a comandos como `Get-SSMMaintenanceWindowTaskList` e `Get-SSMMaintenanceWindowTask`. Esses valores não afetam a execução da tarefa e podem ser ignorados.
Para obter informações sobre tarefas da janela de manutenção que não exigem destinos, consulte Registrar tarefas da janela de manutenção sem destinos.

O exemplo a seguir configura uma automação como uma tarefa registrada para uma janela de manutenção com prioridade 1. Ele também demonstra a omissão das opções de `--targets`, `--max-errors` e `--max-concurrency` para uma tarefa da janela de manutenção sem destino. A automação usa o runbook `AWS-StartEC2Instance` e a função assumida do Automation especificada para iniciar as instâncias do EC2 registradas como destinos para a janela de manutenção. A janela de manutenção executa a automação simultaneamente em 5 instâncias, no máximo, a qualquer momento. Além disso, a tarefa registrada interromperá a execução em mais instâncias por um intervalo específico, se a contagem de erros exceder 1.

Linux & macOS

```
aws ssm register-task-with-maintenance-window \
--window-id mw-0c50858d01EXAMPLE \
--name StartEC2Instances \
--task-arn AWS-StartEC2Instance \
--service-role-arn arn:aws:iam::123456789012:role/MaintenanceWindowRole \
--task-type AUTOMATION \
--task-invocation-parameters "{\"Automation\":{\"Parameters\":{\"InstanceId\":
[\"{{TARGET_ID}}\"],\"AutomationAssumeRole\":[\"arn:aws:iam::123456789012:role/
AutomationAssumeRole\"]}}}" \
--priority 1
```

Windows

```
aws ssm register-task-with-maintenance-window ^
--window-id mw-0c50858d01EXAMPLE ^
--name StartEC2Instances ^
--task-arn AWS-StartEC2Instance ^
--service-role-arn arn:aws:iam::123456789012:role/MaintenanceWindowRole ^
```

```
--task-type AUTOMATION ^
--task-invocation-parameters "{\"Automation\":{\"Parameters\":{\"InstanceId\":
[\"{{TARGET_ID}}\"],\"AutomationAssumeRole\":[\"arn:aws:iam::123456789012:role/
AutomationAssumeRole\"]}}}" ^
--priority 1
```

PowerShell

```
Register-SSMTaskWithMaintenanceWindow `
-WindowId mw-0c50858d01EXAMPLE `
-Name "StartEC2" `
-TaskArn "AWS-StartEC2Instance" `
-ServiceRoleArn "arn:aws:iam::123456789012:role/MaintenanceWindowRole" `
-TaskType "AUTOMATION" `
-Automation_Parameter
 @{ "InstanceId"="{{TARGET_ID}}";"AutomationAssumeRole"="arn:aws:iam::123456789012:role/
AutomationAssumeRole" } `
-Priority 1
```

O comando retorna detalhes da nova tarefa registrada semelhantes aos seguintes.

Linux & macOS

```
{
"WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE"
}
```

Windows

```
{
"WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE"
}
```

PowerShell

```
4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE
```

3. Para visualizar a tarefa registrada, execute o seguinte comando. Substitua *maintenance windows ID* (ID das janelas de manutenção) por suas próprias informações.

Linux & macOS

```
aws ssm describe-maintenance-window-tasks \
--window-id maintenance window ID
```

Windows

```
aws ssm describe-maintenance-window-tasks ^
--window-id maintenance window ID
```

PowerShell

```
Get-SSMMaintenanceWindowTaskList `
-WindowId maintenance window ID
```

O sistema retorna informações como estas.

Linux & macOS

```
{
"Tasks": [
    {
        "ServiceRoleArn": "arn:aws:iam::123456789012:role/
MaintenanceWindowRole",
        "MaxErrors": "1",
        "TaskArn": "AWS-StartEC2Instance",
        "MaxConcurrency": "1",
        "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
        "TaskParameters": {},
        "Priority": 1,
        "WindowId": "mw-0c50858d01EXAMPLE",
        "Type": "AUTOMATION",
        "Targets": [
        ],
        "Name": "StartEC2"
    }
]
}
```

Windows

```
{
"Tasks": [
    {
        "ServiceRoleArn": "arn:aws:iam::123456789012:role/
MaintenanceWindowRole",
        "MaxErrors": "1",
        "TaskArn": "AWS-StartEC2Instance",
        "MaxConcurrency": "1",
        "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
        "TaskParameters": {},
        "Priority": 1,
        "WindowId": "mw-0c50858d01EXAMPLE",
        "Type": "AUTOMATION",
        "Targets": [
        ],
        "Name": "StartEC2"
    }
]
}
```

PowerShell

```
Description    :
LoggingInfo    :
MaxConcurrency : 5
MaxErrors      : 1
Name           : StartEC2
Priority       : 1
ServiceRoleArn : arn:aws:iam::123456789012:role/MaintenanceWindowRole
Targets        : {}
TaskArn        : AWS-StartEC2Instance
TaskParameters : {}
Type           : AUTOMATION
WindowId       : mw-0c50858d01EXAMPLE
WindowTaskId   : 4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE
```

Referência de ações do Systems Manager Automation

Essa referência descreve as ações do Automation que você pode especificar em um runbook do Automation. O Automation é uma ferramenta do AWS Systems Manager. Essas ações não podem ser usadas em outros tipos de documentos do Systems Manager (SSM). Para obter informações sobre plugins para outros tipos de documento do SSM, consulte Referência de plug-ins de documentos de comando.

O Systems Manager Automation executa as etapas definidas em runbooks de automação. Cada etapa está associada a uma ação específica. A ação determina as entradas, o comportamento e as saídas da etapa. As etapas são definidas na seção `mainSteps` do seu runbook.

Não é necessário especificar as saídas de uma ação ou etapa. As saídas são predeterminadas pela ação associada à etapa. Quando você especifica entradas de etapa em seus runbooks, você pode fazer referência a um ou mais resultados de uma etapa anterior. Por exemplo, você pode disponibilizar a saída de `aws:runInstances` para uma ação `aws:runCommand` subsequente. Você também pode referenciar resultados de etapas anteriores na seção `Output` do runbook.

> **Important**
>
> Se você executar um fluxo de trabalho de automação que chama outros serviços usando uma função de serviço do AWS Identity and Access Management (IAM), esteja ciente de que esta função deve ser configurada com permissão para chamar esses serviços. Esse requisito aplica-se a todos os runbooks do Automation da AWS (runbooks da `AWS-*`), como os runbooks `AWS-ConfigureS3BucketLogging`, `AWS-CreateDynamoDBBackup` e `AWS-RestartEC2Instance`, entre outros. Esse requisito também se aplica a todos os runbooks personalizados do Automation criados que invoquem outros Serviços da AWS, usando ações que chamam outros serviços. Por exemplo, se você usar as ações `aws:executeAwsApi`, `aws:createStack` ou `aws:copyImage`, configure a função de serviço com permissão para invocar esses serviços. É possível habilitar permissões para outros Serviços da AWS, adicionando uma política em linha do IAM à função. Para obter mais informações, consulte (Opcional) Adicione uma política em linha ou uma política gerenciada pelo cliente para invocar outros Serviços da AWS.

Tópicos

- Propriedades compartilhadas por todas as ações
- aws:approve – Pausa uma automação para aprovação manual

- aws:assertAwsResourceProperty: define um estado do recurso da AWS ou o estado do evento
- aws:branch – Executa etapas de automação condicionais
- aws:changeInstanceState: altera ou declara o estado da instância
- aws:copyImage: copia ou criptografa um Amazon Machine Image
- aws:createImage: cria uma imagem de máquina da Amazon
- aws:createStack: cria uma pilha do AWS CloudFormation
- aws:createTags: cria tags para os recursos da AWS
- aws:deleteImage: exclui uma Imagem de máquina da Amazon
- aws:deleteStack: exclui uma pilha do AWS CloudFormation
- aws:executeAutomation – Executa outra automação
- aws:executeAwsApi: chama e executa as operações de API do AWS
- aws:executeScript – Executa um script
- aws:executeStateMachine – Executa uma máquina de estado do AWS Step Functions
- aws:invokeWebhook — Invoque uma integração de webhook do Automation
- aws:invokeLambdaFunction – Invoca uma função do AWS Lambda
- aws:loop: itera nas etapas de uma automação
- aws:pause – Pausa uma automação
- aws:runCommand – Executa um comando em uma instância gerenciada
- aws:runInstances – Executa uma instância do Amazon EC2
- aws:sleep: atrasa uma automação
- aws:updateVariable: atualiza um valor para uma variável do runbook
- aws:waitForAwsResourceProperty: aguarde uma propriedade de recurso da AWS
- Variáveis de sistema de automação

Propriedades compartilhadas por todas as ações

Propriedades comuns são parâmetros ou opções encontradas em todas as ações. Algumas opções definem o comportamento para uma etapa, como o tempo de espera para que uma etapa seja concluída e o que fazer se a etapa falhar. As seguintes propriedades são comuns a todas as ações.

description

Informações que você fornece para descrever a finalidade de um runbook ou de uma etapa.

Tipo: String

Obrigatório: Não

name

Um identificador que deve ser exclusivo em todos os nomes de etapas do runbook.

Tipo: String

Padrão permitido: [a-zA-Z0-9_]+$

Obrigatório: Sim

action

O nome da ação que a etapa deve executar. `aws:runCommand` – Executa um comando em uma instância gerenciada é um exemplo de uma ação que você pode especificar aqui. Esse documento fornece informações detalhadas sobre todas as ações disponíveis.

Tipo: string

Obrigatório: sim

maxAttempts

Quantas vezes a etapa deve ser repetida em caso de falha. Se o valor for maior que 1, a etapa não será considerada falha até que todas as novas tentativas tenham falhado. O valor padrão é 1.

Tipo: inteiro

Obrigatório: não

timeoutSeconds

O valor de tempo limite para a etapa. Se o tempo limite for atingido, e o valor de `maxAttempts` for maior que 1, a etapa não será considerada expirada até que todas as novas tentativas tenham sido feitas.

Tipo: inteiro

Obrigatório: Não

onFailure

Indica se a automação deve ser interrompida, se deve continuar ou seguir para outra etapa, no caso de falha. O valor padrão desta opção é anular.

Tipo: String

Valores válidos: Anular | Continuar | etapa:*step_name*

Obrigatório: Não

onCancel

Indica para qual etapa a automação deve passar no caso de um usuário cancelar a automação. A automação executa o fluxo de trabalho de cancelamento por um máximo de dois minutos.

Tipo: String

Valores válidos: Abort | step:*step_name*

Obrigatório: Não

A propriedade `onCancel` não oferece suporte para mover para as seguintes ações:

- `aws:approve`
- `aws:copyImage`
- `aws:createImage`
- `aws:createStack`
- `aws:createTags`
- `aws:loop`
- `aws:pause`
- `aws:runInstances`
- `aws:sleep`

isEnd

Essa opção interrompe automação no final de determinada etapa. A automação é interrompida se a execução da etapa falhar ou for bem-sucedida. O valor padrão é falso.

Tipo: booliano

Valores válidos: verdadeiro | falso

Obrigatório: Não

nextStep

Especifica qual etapa de uma automação deve ser processada imediatamente após a conclusão bem-sucedida de uma etapa.

Tipo: String

Obrigatório: Não

isCritical

Designa uma etapa como essencial para a conclusão bem sucedida da automação. Se uma etapa com essa designação falhar, a automação relatará o status final do Automation como Failed (com falha). Essa propriedade só será avaliada se você a definir explicitamente em sua etapa. Se o a propriedade `onFailure` for definida como `Continue` em uma etapa, o valor padrão será falso. Caso contrário, valor padrão desta opção é verdadeiro.

Tipo: booliano

Valores válidos: verdadeiro | falso

Obrigatório: Não

inputs

As propriedades específicas da ação.

Tipo: mapa

Obrigatório: sim

Exemplo

```
---
description: "Custom Automation Example"
schemaVersion: '0.3'
assumeRole: "{{ AutomationAssumeRole }}"
parameters:
  AutomationAssumeRole:
    type: String
    description: "(Required) The ARN of the role that allows Automation to perform
      the actions on your behalf. If no role is specified, Systems Manager Automation
      uses your IAM permissions to run this runbook."
    default: ''
  InstanceId:
      type: String
      description: "(Required) The Instance Id whose root EBS volume you want to
 restore the latest Snapshot."
      default: ''
```

```
mainSteps:
- name: getInstanceDetails
  action: aws:executeAwsApi
  onFailure: Abort
  inputs:
    Service: ec2
    Api: DescribeInstances
    InstanceIds:
    - "{{ InstanceId }}"
  outputs:
    - Name: availabilityZone
      Selector: "$.Reservations[0].Instances[0].Placement.AvailabilityZone"
      Type: String
    - Name: rootDeviceName
      Selector: "$.Reservations[0].Instances[0].RootDeviceName"
      Type: String
  nextStep: getRootVolumeId
- name: getRootVolumeId
  action: aws:executeAwsApi
  maxAttempts: 3
  onFailure: Abort
  inputs:
    Service: ec2
    Api: DescribeVolumes
    Filters:
    -  Name: attachment.device
       Values: ["{{ getInstanceDetails.rootDeviceName }}"]
    -  Name: attachment.instance-id
       Values: ["{{ InstanceId }}"]
  outputs:
    - Name: rootVolumeId
      Selector: "$.Volumes[0].VolumeId"
      Type: String
  nextStep: getSnapshotsByStartTime
- name: getSnapshotsByStartTime
  action: aws:executeScript
  timeoutSeconds: 45
  onFailure: Abort
  inputs:
    Runtime: python3.8
    Handler: getSnapshotsByStartTime
    InputPayload:
      rootVolumeId : "{{ getRootVolumeId.rootVolumeId }}"
    Script: |-
```

```
      def getSnapshotsByStartTime(events,context):
        import boto3

        #Initialize client
        ec2 = boto3.client('ec2')
        rootVolumeId = events['rootVolumeId']
        snapshotsQuery = ec2.describe_snapshots(
          Filters=[
            {
              "Name": "volume-id",
              "Values": [rootVolumeId]
            }
          ]
        )
        if not snapshotsQuery['Snapshots']:
          noSnapshotFoundString = "NoSnapshotFound"
          return { 'noSnapshotFound' : noSnapshotFoundString }
        else:
          jsonSnapshots = snapshotsQuery['Snapshots']
          sortedSnapshots = sorted(jsonSnapshots, key=lambda k: k['StartTime'],
 reverse=True)
          latestSortedSnapshotId = sortedSnapshots[0]['SnapshotId']
          return { 'latestSnapshotId' : latestSortedSnapshotId }
  outputs:
  - Name: Payload
    Selector: $.Payload
    Type: StringMap
  - Name: latestSnapshotId
    Selector: $.Payload.latestSnapshotId
    Type: String
  - Name: noSnapshotFound
    Selector: $.Payload.noSnapshotFound
    Type: String
  nextStep: branchFromResults
- name: branchFromResults
  action: aws:branch
  onFailure: Abort
  onCancel: step:startInstance
  inputs:
    Choices:
    - NextStep: createNewRootVolumeFromSnapshot
      Not:
        Variable: "{{ getSnapshotsByStartTime.noSnapshotFound }}"
        StringEquals: "NoSnapshotFound"
```

```
  isEnd: true
- name: createNewRootVolumeFromSnapshot
  action: aws:executeAwsApi
  onFailure: Abort
  inputs:
    Service: ec2
    Api: CreateVolume
    AvailabilityZone: "{{ getInstanceDetails.availabilityZone }}"
    SnapshotId: "{{ getSnapshotsByStartTime.latestSnapshotId }}"
  outputs:
    - Name: newRootVolumeId
      Selector: "$.VolumeId"
      Type: String
  nextStep: stopInstance
- name: stopInstance
  action: aws:executeAwsApi
  onFailure: Abort
  inputs:
    Service: ec2
    Api: StopInstances
    InstanceIds:
    - "{{ InstanceId }}"
  nextStep: verifyVolumeAvailability
- name: verifyVolumeAvailability
  action: aws:waitForAwsResourceProperty
  timeoutSeconds: 120
  inputs:
    Service: ec2
    Api: DescribeVolumes
    VolumeIds:
    - "{{ createNewRootVolumeFromSnapshot.newRootVolumeId }}"
    PropertySelector: "$.Volumes[0].State"
    DesiredValues:
    - "available"
  nextStep: verifyInstanceStopped
- name: verifyInstanceStopped
  action: aws:waitForAwsResourceProperty
  timeoutSeconds: 120
  inputs:
    Service: ec2
    Api: DescribeInstances
    InstanceIds:
    - "{{ InstanceId }}"
    PropertySelector: "$.Reservations[0].Instances[0].State.Name"
```

```
    DesiredValues:
    - "stopped"
  nextStep: detachRootVolume
- name: detachRootVolume
  action: aws:executeAwsApi
  onFailure: Abort
  isCritical: true
  inputs:
    Service: ec2
    Api: DetachVolume
    VolumeId: "{{ getRootVolumeId.rootVolumeId }}"
  nextStep: verifyRootVolumeDetached
- name: verifyRootVolumeDetached
  action: aws:waitForAwsResourceProperty
  timeoutSeconds: 30
  inputs:
    Service: ec2
    Api: DescribeVolumes
    VolumeIds:
    - "{{ getRootVolumeId.rootVolumeId }}"
    PropertySelector: "$.Volumes[0].State"
    DesiredValues:
    - "available"
  nextStep: attachNewRootVolume
- name: attachNewRootVolume
  action: aws:executeAwsApi
  onFailure: Abort
  inputs:
    Service: ec2
    Api: AttachVolume
    Device: "{{ getInstanceDetails.rootDeviceName }}"
    InstanceId: "{{ InstanceId }}"
    VolumeId: "{{ createNewRootVolumeFromSnapshot.newRootVolumeId }}"
  nextStep: verifyNewRootVolumeAttached
- name: verifyNewRootVolumeAttached
  action: aws:waitForAwsResourceProperty
  timeoutSeconds: 30
  inputs:
    Service: ec2
    Api: DescribeVolumes
    VolumeIds:
    - "{{ createNewRootVolumeFromSnapshot.newRootVolumeId }}"
    PropertySelector: "$.Volumes[0].Attachments[0].State"
    DesiredValues:
```

```
    - "attached"
  nextStep: startInstance
- name: startInstance
  action: aws:executeAwsApi
  onFailure: Abort
  inputs:
    Service: ec2
    Api: StartInstances
    InstanceIds:
    - "{{ InstanceId }}"
```

`aws:approve` – Pausa uma automação para aprovação manual

Pausa temporariamente uma automação, até que as entidades principais designadas aprovem ou rejeitem a ação. Depois que o número necessário de aprovações for atingido, a execução da automação será retomada. Você pode inserir a etapa de aprovação em qualquer lugar na seção `mainSteps` do runbook.

> **Note**
>
> Essa ação não é compatível com automações de várias contas e regiões. O tempo limite padrão para essa ação é de 7 dias (604.800 segundos) e o valor máximo é de 30 dias (2.592.000 segundos). Você pode limitar ou prolongar o tempo limite especificando o parâmetro `timeoutSeconds` para uma etapa `aws:approve`.

No exemplo a seguir, a ação `aws:approve` pausa temporariamente a automação até que um aprovador a aceite ou rejeite. Após a aprovação, a automação executa um comando simples do PowerShell.

YAML

```
---
description: RunInstancesDemo1
schemaVersion: '0.3'
assumeRole: "{{ assumeRole }}"
parameters:
  assumeRole:
    type: String
  message:
    type: String
```

```
mainSteps:
- name: approve
  action: aws:approve
  timeoutSeconds: 1000
  onFailure: Abort
  inputs:
    NotificationArn: arn:aws:sns:us-east-2:12345678901:AutomationApproval
    Message: "{{ message }}"
    MinRequiredApprovals: 1
    Approvers:
    - arn:aws:iam::12345678901:user/AWS-User-1
- name: run
  action: aws:runCommand
  inputs:
    InstanceIds:
    - i-1a2b3c4d5e6f7g
    DocumentName: AWS-RunPowerShellScript
    Parameters:
      commands:
      - date
```

JSON

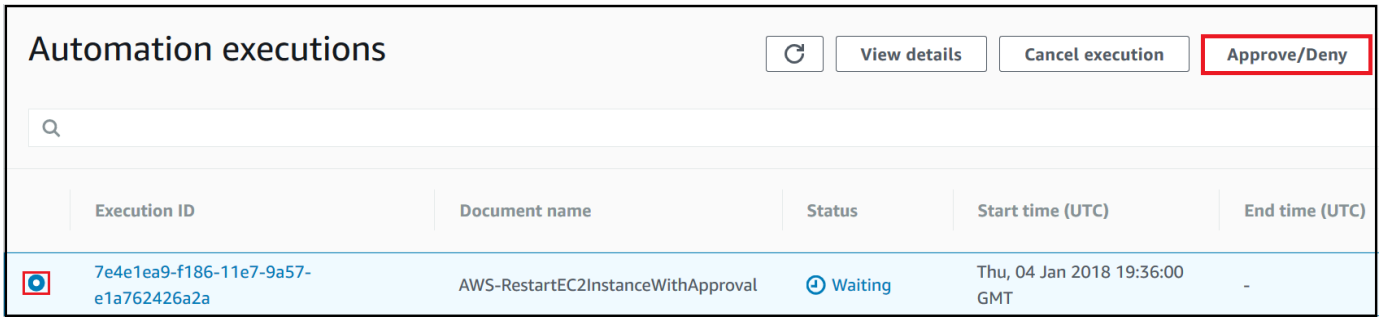
```
{
   "description":"RunInstancesDemo1",
   "schemaVersion":"0.3",
   "assumeRole":"{{ assumeRole }}",
   "parameters":{
      "assumeRole":{
         "type":"String"
      },
      "message":{
         "type":"String"
      }
   },
   "mainSteps":[
      {
         "name":"approve",
         "action":"aws:approve",
         "timeoutSeconds":1000,
         "onFailure":"Abort",
         "inputs":{
```

```
            "NotificationArn":"arn:aws:sns:us-
east-2:12345678901:AutomationApproval",
            "Message":"{{ message }}",
            "MinRequiredApprovals":1,
            "Approvers":[
               "arn:aws:iam::12345678901:user/AWS-User-1"
            ]
         }
      },
      {
         "name":"run",
         "action":"aws:runCommand",
         "inputs":{
            "InstanceIds":[
               "i-1a2b3c4d5e6f7g"
            ],
            "DocumentName":"AWS-RunPowerShellScript",
            "Parameters":{
               "commands":[
                  "date"
               ]
            }
         }
      }
   ]
}
```

É possível aprovar ou negar automações que estão aguardando aprovação no console.

Para aprovar ou negar automações em espera

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação à esquerda, escolha Automation (Automação).
3. Escolha a opção ao lado de uma automação com o status Waiting (Em espera).



4. Escolha Approve/Deny (Aprovar/negar).
5. Analise os detalhes da automação.
6. Escolha Approve (Aprovar) ou Deny (Negar), digite um comentário opcional e escolha Submit (Enviar).

Exemplo de entrada

YAML

```
NotificationArn: arn:aws:sns:us-west-1:12345678901:Automation-ApprovalRequest
Message: Please approve this step of the Automation.
MinRequiredApprovals: 3
Approvers:
- IamUser1
- IamUser2
- arn:aws:iam::12345678901:user/IamUser3
- arn:aws:iam::12345678901:role/IamRole
```

JSON

```
{
   "NotificationArn":"arn:aws:sns:us-west-1:12345678901:Automation-ApprovalRequest",
   "Message":"Please approve this step of the Automation.",
   "MinRequiredApprovals":3,
   "Approvers":[
      "IamUser1",
      "IamUser2",
      "arn:aws:iam::12345678901:user/IamUser3",
      "arn:aws:iam::12345678901:role/IamRole"
   ]
}
```

NotificationArn

O tópico do nome do recurso da Amazon (ARN de um Amazon Simple Notification Service, Amazon SNS) para aprovações do Automation. Quando você especifica uma etapa `aws:approve` em um runbook, o Automation envia uma mensagem a esse tópico, permitindo que as entidades principais saibam se devem aprovar ou rejeitar uma etapa do Automation. O título do tópico do Amazon SNS deve ser prefixado com "Automation".

Tipo: string

Obrigatório: Não

Message

As informações que você deseja incluir no tópico do Amazon SNS quando a solicitação de aprovação é enviada. O comprimento máximo da mensagem é de 4096 caracteres.

Tipo: string

Obrigatório: Não

MinRequiredApprovals

O número mínimo de aprovações necessárias para retomar a automação. Se você não especificar um valor, o sistema assumirá 1 como padrão. O valor desse parâmetro deve ser um número positivo. O valor desse parâmetro não pode exceder o número de aprovadores definidas pelo parâmetro `Approvers`.

Tipo: número inteiro

Obrigatório: Não

Approvers

Uma lista de entidades principais autenticadas da AWS que podem aprovar ou rejeitar a ação. O número máximo de aprovadores é 10. É possível especificar entidades principais usando qualquer um dos seguintes formatos:

- Um nome de usuário
- Um ARN do usuário
- Um ARN de função do IAM

- Um IAM assume o perfil do ARN

Tipo: StringList

Obrigatório: Sim

EnhancedApprovals

Essa entrada é usada somente para modelos do Change Manager. Uma lista de entidades principais autenticadas pela AWS que podem aprovar ou rejeitar a ação, o tipo de entidade principal do IAM e o número mínimo de aprovadores. Veja um exemplo a seguir:

```
schemaVersion: "0.3"
emergencyChange: false
autoApprovable: false
mainSteps:
    - name: ApproveAction1
    action: aws:approve
    timeoutSeconds: 604800
    inputs:
        Message: Please approve this change request
        MinRequiredApprovals: 3
        EnhancedApprovals:
        Approvers:
            - approver: John Stiles
            type: IamUser
            minRequiredApprovals: 0
            - approver: Ana Carolina Silva
            type: IamUser
            minRequiredApprovals: 0
            - approver: GroupOfThree
            type: IamGroup
            minRequiredApprovals: 0
            - approver: RoleOfTen
            type: IamRole
            minRequiredApprovals: 0
```

Tipo: StringList

Obrigatório: Sim

Saída

ApprovalStatus

O status de aprovação da etapa. O status pode ser um dos seguintes: Approved, Rejected ou Waiting. Waiting significa que a Automação está aguardando a entrada de aprovadores.

Tipo: string

ApproverDecisions

Um mapa JSON que inclui a decisão de aprovação de cada aprovador.

Tipo: MapList

`aws:assertAwsResourceProperty`: define um estado do recurso da AWS ou o estado do evento

A ação `aws:assertAwsResourceProperty` permite que você declare um estado de recurso específico ou estado de evento para uma determinada etapa do Automation.

Para obter mais exemplos de como usar essa ação, consulte Exemplos adicionais de runbook.

Entrada

As entradas são definidas pela operação de API que você escolher.

YAML

```
action: aws:assertAwsResourceProperty
inputs:
  Service: The official namespace of the service
  Api: The API operation or method name
  API operation inputs or parameters: A value
  PropertySelector: Response object
  DesiredValues:
  - Desired property values
```

JSON

```
{
  "action": "aws:assertAwsResourceProperty",
  "inputs": {
    "Service":"The official namespace of the service",
    "Api":"The API operation or method name",
    "API operation inputs or parameters":"A value",
```

```
    "PropertySelector": "Response object",
    "DesiredValues": [
      "Desired property values"
    ]
  }
}
```

Serviço

O namespace do AWS service (Serviço da AWS) que contém a operação de API que você deseja executar. Por exemplo, o namespace para o Systems Manager é `ssm`. O namespace do Amazon EC2 é `ec2`. Você pode visualizar uma lista de namespaces de AWS service (Serviço da AWS) compatíveis na seção Available Services (Serviços disponíveis) da Referência de comandos da AWS CLI.

Tipo: string

Obrigatório: Sim

API

O nome da operação de API que você deseja executar. Você pode visualizar as operações de API (também chamadas de métodos), escolhendo um serviço na navegação à esquerda na seguinte página de Referência de serviços: Escolha um método na seção Client (Cliente) para o serviço que você deseja invocar. Por exemplo, todas as operações de API (métodos) do Amazon Relational Database Service (Amazon RDS) estão listadas na seguinte página: Amazon RDS methods (Métodos do Amazon RDS).

Tipo: string

Obrigatório: Sim

Entradas de operação da API

Uma ou mais entradas de operação da API. Você pode visualizar as entradas disponíveis (também chamadas de parâmetros), escolhendo um serviço na navegação à esquerda na seguinte página de Referência de serviços. Escolha um método na seção Client (Cliente) para o serviço que você deseja invocar. Por exemplo, todos os métodos de API estão listados na página a seguir: Métodos do Amazon RDS. Escolha o método describe_db_instances e role para baixo para ver os parâmetros disponíveis, como DBInstanceIdentifier, Name (Nome) e Values (Valores). Use o formato a seguir para especificar mais de uma entrada.

YAML

```
inputs:
  Service: The official namespace of the service
  Api: The API operation name
  API input 1: A value
  API Input 2: A value
  API Input 3: A value
```

JSON

```
"inputs":{
      "Service":"The official namespace of the service",
      "Api":"The API operation name",
      "API input 1":"A value",
      "API Input 2":"A value",
      "API Input 3":"A value"
}
```

Tipo: determinado pela ação de API escolhida

Obrigatório: Sim

PropertySelector

O JSONPath para um determinado atributo no objeto de resposta. Você pode visualizar os objetos de resposta escolhendo um serviço na navegação à esquerda na seguinte página de Referência de serviços. Escolha um método na seção Client (Cliente) para o serviço que você deseja invocar. Por exemplo, todos os métodos de API estão listados na página a seguir: Métodos do Amazon RDS. Escolha o método describe_db_instances e role para baixo até a seção Response Structure (Estrutura de resposta). DBInstances é listado como um objeto de resposta.

Tipo: string

Obrigatório: Sim

DesiredValues

O status ou estado esperado no qual a automação deve continuar. Se você especificar um valor booleano, você deve usar uma letra maiúscula, como Verdadeiro ou Falso.

Tipo: StringList

Obrigatório: Sim

`aws:branch` – Executa etapas de automação condicionais

A ação `aws:branch` permite que você crie uma automação dinâmica que avalia diferentes opções em uma única etapa e, em seguida, salta para outra etapa no runbook, com base nos resultados da avaliação.

Quando você especifica a ação `aws:branch` para uma etapa, você especifica as `Choices` que a automação deve avaliar. As `Choices` podem ser baseadas no valor especificado na seção `Parameters` do runbook ou em um valor dinâmico gerado como saída da etapa anterior. A automação avalia cada opção usando uma expressão booleana. Se a primeira opção for verdadeira, a automação pulará para a etapa designada para essa opção. Se a primeira opção for falsa, a automação avaliará a próxima opção. A automação continua a avaliar cada opção até processar uma opção verdadeira. A automação pula para a etapa designada para a opção true (verdadeira).

Se nenhuma das opções for verdadeira, a automação verificará se a etapa contém um valor `default`. Um valor padrão define uma etapa para a qual a automação deve saltar se nenhuma das opções for verdadeira. Se nenhum valor `default` for especificado para a etapa, a automação processará a próxima etapa no runbook.

A ação `aws:branch` oferece suporte a avaliações de opções complexas usando uma combinação de operadores `And`, `Not` e `Or`. Para obter mais informações sobre como usar o `aws:branch`, incluindo runbooks de exemplo e exemplos que usam operadores diferentes, consulte Uso de instruções condicionais em runbooks.

Entrada

Especifique uma ou mais `Choices` em uma etapa. As `Choices` podem ser baseadas no valor especificado na seção `Parameters` do runbook ou em um valor dinâmico gerado como saída da etapa anterior. Aqui está um exemplo de YAML que avalia um parâmetro.

```
mainSteps:
- name: chooseOS
  action: aws:branch
  inputs:
    Choices:
    - NextStep: runWindowsCommand
      Variable: "{{Name of a parameter defined in the Parameters section. For example:
 OS_name}}"
```

```
      StringEquals: windows
    - NextStep: runLinuxCommand
      Variable: "{{Name of a parameter defined in the Parameters section. For example:
 OS_name}}"
      StringEquals: linux
    Default:
      sleep3
```

Aqui está um exemplo de YAML que avalia a saída de uma etapa anterior.

```
mainSteps:
- name: chooseOS
  action: aws:branch
  inputs:
    Choices:
    - NextStep: runPowerShellCommand
      Variable: "{{Name of a response object. For example: GetInstance.platform}}"
      StringEquals: Windows
    - NextStep: runShellCommand
      Variable: "{{Name of a response object. For example: GetInstance.platform}}"
      StringEquals: Linux
    Default:
      sleep3
```

Choices

Uma ou mais expressões que a Automação deve avaliar ao determinar a próxima etapa para a ser processada. Choices são avaliadas usando uma expressão booliana. Cada opção deve definir as seguintes opções:

- NextStep: a próxima etapa no runbook para processar se a opção designada é verdadeira.
- Variable: especifique o nome de um parâmetro definido na seção `Parameters` do runbook. Ou especifique um objeto de saída de uma etapa anterior no runbook. Para obter mais informações sobre como criar variáveis para `aws:branch`, consulte Sobre a criação de variáveis de saída.
- Operation: os critérios usados para avaliar a opção. A ação `aws:branch` oferece suporte às seguintes operações:

  Operações de string

  - StringEquals
  - EqualsIgnoreCase

- StartsWith
- EndsWith
- Contém

Operações numéricas

- NumericEquals
- NumericGreater
- NumericLesser
- NumericGreaterOrEquals
- NumericLesser
- NumericLesserOrEquals

Operação booleana

- BooleanEquals

> **Important**
>
> Quando você cria um runbook, o sistema valida cada operação dele. Se uma operação não for suportada, o sistema retornará um erro ao tentar criar o runbook.

Padrão

O nome de uma etapa para a qual a automação deve saltar, se nenhuma das `Choices` forem verdadeiras.

Tipo: string

Obrigatório: Não

> **Note**
>
> A ação `aws:branch` oferece suporte a operadores `And`, `Or` e `Not`. Para obter exemplos de `aws:branch` que usam operadores, consulte Uso de instruções condicionais em runbooks.

`aws:changeInstanceState`: altera ou declara o estado da instância

Altera ou declara o estado da instância.

Essa ação pode ser usada no modo de declaração (não executa a API para alterar o estado, mas verifica se a instância está no estado desejado). Para usar o modo de declaração, defina o parâmetro `CheckStateOnly` como true. Esse modo é útil no Windows ao executar o comando Sysprep, um comando assíncrono que pode ser executado em segundo plano por um longo tempo. Você pode garantir que a instância seja interrompida antes de criar uma Amazon Machine Image (AMI).

> **Note**
>
> O valor de tempo limite padrão para esta ação é 3600 segundos (uma hora). Você pode limitar ou prolongar o tempo limite especificando o parâmetro `timeoutSeconds` para uma etapa `aws:changeInstanceState`.

Entrada

YAML

```
name: stopMyInstance
action: aws:changeInstanceState
maxAttempts: 3
timeoutSeconds: 3600
onFailure: Abort
inputs:
  InstanceIds:
  - i-1234567890abcdef0
  CheckStateOnly: true
  DesiredState: stopped
```

JSON

```
{
    "name":"stopMyInstance",
    "action": "aws:changeInstanceState",
    "maxAttempts": 3,
    "timeoutSeconds": 3600,
    "onFailure": "Abort",
```

```
    "inputs": {
        "InstanceIds": ["i-1234567890abcdef0"],
        "CheckStateOnly": true,
        "DesiredState": "stopped"
    }
}
```

InstanceIds

Os IDs das instâncias.

Tipo: StringList

Obrigatório: Sim

CheckStateOnly

Se false, define o estado da instância como o estado desejado. Se true, declara o estado desejado usando sondagem.

Padrão: `false`

Tipo: Booliano

Obrigatório: Não

DesiredState

O estado desejado. Quando definida como `running`, essa ação aguarda que o estado do Amazon EC2 seja `Running`, o status da instância seja `OK` e o status do sistema seja `OK`, antes de concluir.

Tipo: string

Valores válidos: `running` | `stopped` | `terminated`

Obrigatório: Sim

Force

Se configurado, força a interrupção das instâncias. As instâncias não têm a oportunidade de liberar os caches ou metadados do sistema de arquivos. Se você usar essa opção, deve executar a verificação do sistema de arquivos e os procedimentos de reparo. Essa opção não é recomendada para instâncias do EC2 para Windows Server.

Tipo: booliano

Obrigatório: Não

AdditionalInfo

Reservado.

Tipo: string

Obrigatório: Não

Saída

Nenhum

`aws:copyImage`: copia ou criptografa um Amazon Machine Image

Copia uma Amazon Machine Image (AMI) de qualquer Região da AWS para a região atual. Essa ação também pode criptografar a nova AMI.

Entrada

Essa ação oferece suporte para a maioria dos parâmetros `CopyImage`. Para obter mais informações, consulte CopyImage.

O exemplo a seguir cria uma cópia de uma AMI na região de Seul (`SourceImageID`: ami-0fe10819. `SourceRegion`: ap-northeast-2). A nova AMI é copiada para a região em que você iniciou a ação do Automation. A AMI copiada será criptografada, pois o sinalizador `Encrypted` opcional está definido como `true`.

YAML

```
name: createEncryptedCopy
action: aws:copyImage
maxAttempts: 3
onFailure: Abort
inputs:
  SourceImageId: ami-0fe10819
  SourceRegion: ap-northeast-2
  ImageName: Encrypted Copy of LAMP base AMI in ap-northeast-2
  Encrypted: true
```

JSON

```
{
    "name": "createEncryptedCopy",
    "action": "aws:copyImage",
    "maxAttempts": 3,
    "onFailure": "Abort",
    "inputs": {
        "SourceImageId": "ami-0fe10819",
        "SourceRegion": "ap-northeast-2",
        "ImageName": "Encrypted Copy of LAMP base AMI in ap-northeast-2",
        "Encrypted": true
    }
}
```

SourceRegion

A região em que a AMI de origem existe.

Tipo: string

Obrigatório: Sim

SourceImageId

O ID de AMI a ser copiado da região de origem.

Tipo: string

Obrigatório: Sim

ImageName

O nome da nova imagem.

Tipo: string

Obrigatório: Sim

ImageDescription

Uma descrição da imagem de destino.

Tipo: string

Obrigatório: Não

Criptografado

Criptografe a AMI de destino.

Tipo: booliano

Obrigatório: Não

KmsKeyId

O nome do recurso da Amazon (ARN) completo da AWS KMS key a ser usado para criptografar os snapshots de uma imagem durante uma operação de cópia. Para obter mais informações, consulte CopyImage.

Tipo: string

Obrigatório: Não

ClientToken

Um identificador único e com diferenciação entre maiúsculas de minúsculas que você fornece para garantir a idempotência da solicitação. Para obter mais informações, consulte CopyImage.

Tipo: string

Obrigatório: Não

Saída

ImageId

O ID da imagem copiada.

ImageState

O estado da imagem copiada.

Valores válidos: `available` | `pending` | `failed`

`aws:createImage`: cria uma imagem de máquina da Amazon

Cria uma Amazon Machine Image (AMI) de uma instância que está em execução, parando ou parada e solicita que `ImageState` seja `available`.

Entrada

Essa ação oferece suporte aos seguintes parâmetros de `CreateImage`: Para obter mais informações, consulte CreateImage.

YAML

```
name: createMyImage
action: aws:createImage
maxAttempts: 3
onFailure: Abort
inputs:
  InstanceId: i-1234567890abcdef0
  ImageName: AMI Created on{{global:DATE_TIME}}
  NoReboot: true
  ImageDescription: My newly created AMI
```

JSON

```
{
    "name": "createMyImage",
    "action": "aws:createImage",
    "maxAttempts": 3,
    "onFailure": "Abort",
    "inputs": {
        "InstanceId": "i-1234567890abcdef0",
        "ImageName": "AMI Created on{{global:DATE_TIME}}",
        "NoReboot": true,
        "ImageDescription": "My newly created AMI"
    }
}
```

InstanceId

O ID da instância.

Tipo: string

Obrigatório: Sim

ImageName

O nome da imagem.

Tipo: string

Obrigatório: Sim

ImageDescription

Uma descrição da imagem.

Tipo: string

Obrigatório: Não

NoReboot

Um literal booleana.

Por padrão, o Amazon Elastic Compute Cloud (Amazon EC2) tenta desligar e reinicializar a instância antes de criar a imagem. Se a opção No Reboot (Não reinicializar) estiver definida como `true`, o Amazon EC2 não desligará a instância antes de criar a imagem. Quando esta opção é usada, não é possível garantir a integridade do sistema de arquivos na imagem criada.

Se você não quiser executar a instância depois de criar uma AMI nela, primeiro use a ação `aws:changeInstanceState`: altera ou declara o estado da instância para interromper a instância e, depois, use a ação `aws:createImage` com a opção NoReboot (Não inicializar) definida como `true`.

Tipo: booliano

Obrigatório: Não

BlockDeviceMappings

Os dispositivos de bloco para a instância.

Tipo: mapa

Obrigatório: Não

Saída

ImageId

O ID da imagem recém-criada.

Tipo: string

ImageState

O estado atual da imagem. Se o estado estiver disponível, a imagem será registrada com êxito e poderá ser usada para executar uma instância.

Tipo: string

`aws:createStack`: cria uma pilha do AWS CloudFormation

Cria uma pilha do AWS CloudFormation a partir de um modelo.

Para obter informações adicionais sobre como criar pilhas do CloudFormation, consulte CreateStack na Referência da API no AWS CloudFormation.

Entrada

YAML

```
name: makeStack
action: aws:createStack
maxAttempts: 1
onFailure: Abort
inputs:
  Capabilities:
  - CAPABILITY_IAM
  StackName: myStack
  TemplateURL: http://s3.amazonaws.com/amzn-s3-demo-bucket/myStackTemplate
  TimeoutInMinutes: 5
  Parameters:
    - ParameterKey: LambdaRoleArn
      ParameterValue: "{{LambdaAssumeRole}}"
    - ParameterKey: createdResource
      ParameterValue: createdResource-{{automation:EXECUTION_ID}}
```

JSON

```
{
    "name": "makeStack",
    "action": "aws:createStack",
    "maxAttempts": 1,
    "onFailure": "Abort",
```

```
    "inputs": {
        "Capabilities": [
            "CAPABILITY_IAM"
        ],
        "StackName": "myStack",
        "TemplateURL": "http://s3.amazonaws.com/amzn-s3-demo-bucket/
myStackTemplate",
        "TimeoutInMinutes": 5,
        "Parameters": [
          {
            "ParameterKey": "LambdaRoleArn",
            "ParameterValue": "{{LambdaAssumeRole}}"
          },
          {
            "ParameterKey": "createdResource",
            "ParameterValue": "createdResource-{{automation:EXECUTION_ID}}"
          }
    }
}
```

Capacidades

Uma lista de valores que você especifica antes que o CloudFormation possa criar certas pilhas. Alguns modelos de pilha podem incluir recursos que podem afetar as permissões na sua Conta da AWS. Para essas pilhas, você deve confirmar explicitamente seus recursos especificando esse parâmetro.

Os valores válidos são `CAPABILITY_IAM`, `CAPABILITY_NAMED_IAM` e `CAPABILITY_AUTO_EXPAND`.

CAPABILITY_IAM e CAPABILITY_NAMED_IAM

Se você tiver recursos do IAM, poderá especificar qualquer recurso. Se tiver recursos do IAM com nomes personalizados, você deverá especificar `CAPABILITY_NAMED_IAM`. Se você não especificar esse parâmetro, essa ação retornará um erro `InsufficientCapabilities`. Os seguintes recursos exigem especificar `CAPABILITY_IAM` ou `CAPABILITY_NAMED_IAM`.

- `AWS::IAM::AccessKey`
- `AWS::IAM::Group`
- `AWS::IAM::InstanceProfile`
- `AWS::IAM::Policy`

- AWS::IAM::Role
- AWS::IAM::User
- AWS::IAM::UserToGroupAddition

Se o seu modelo de pilha contiver esses recursos, recomendamos que você reveja todas as permissões associadas a eles e edite suas permissões, se necessário.

Para obter mais informações, consulte o tópico sobre como Reconhecer recursos do IAM em modelos do AWS CloudFormation.

CAPABILITY_AUTO_EXPAND

Alguns modelos contêm macros. Macros executam o processamento personalizado em modelos. Isso inclui ações simples, como operações de localizar e substituir até transformações extensas de modelos inteiros. Por isso, os usuários geralmente criam um conjunto de alterações no modelo processado para que seja possível revisar as alterações resultantes das macros antes de criar a pilha de fato. Se o modelo de pilha tiver uma ou mais macros, e você optar por criar uma pilha diretamente do modelo processado sem primeiro revisar as alterações resultantes em um conjunto de alterações, será necessário reconhecer esse recurso.

Para obter mais informações, consulte Usar macros do AWS CloudFormation para executar processamento personalizado em modelos no Manual do usuário do AWS CloudFormation.

Tipo: matriz de strings

Valores Válidos: `CAPABILITY_IAM | CAPABILITY_NAMED_IAM | CAPABILITY_AUTO_EXPAND`

Obrigatório: Não

ClientRequestToken

Um identificador exclusivo para essa solicitação CreateStack. Especifique este token se definir maxAttempts nesta etapa como um valor maior que 1. Especificando esse token, o CloudFormation saberá que você não está tentando criar uma nova pilha com o mesmo nome.

Tipo: String

Obrigatório: Não

Restrições de Tamanho: Tamanho mínimo 1. O tamanho máximo é 128.

Padrão: [a-zA-Z0-9][-a-zA-Z0-9]*

DisableRollback

Defina como `true` para desativar a reversão da pilha se a criação da pilha tiver falhado.

Condicional: é possível especificar o parâmetro `DisableRollback` ou `OnFailure`, mas não ambos.

Padrão: `false`

Tipo: booliano

Obrigatório: Não

NotificationARNs

Os ARNs de tópicos do Amazon Simple Notification Service (Amazon SNS) para publicar eventos relacionados à pilha. Você pode encontrar ARNs de tópicos do SNS usando o console do Amazon SNS, https://console.aws.amazon.com/sns/v3/home.

Tipo: matriz de strings

Membros da matriz: número máximo de 5 itens.

Obrigatório: Não

OnFailure

Determina a ação a ser realizada se a criação da pilha falhar. Você deve especificar `DO_NOTHING`, `ROLLBACK` ou `DELETE`.

Condicional: é possível especificar o parâmetro `OnFailure` ou `DisableRollback`, mas não ambos.

Padrão: `ROLLBACK`

Tipo: String

Valores válidos: `DO_NOTHING | ROLLBACK | DELETE`

Obrigatório: Não

Parâmetros

Uma lista de estruturas `Parameter` que especificam parâmetros de entrada para a pilha. Para obter mais informações, consulte o tipo de dados Parameter.

Tipo: matriz de objetos Parameter

Obrigatório: Não

ResourceTypes

Os tipos de recursos de modelo com os quais você tem permissões para trabalhar para essa ação de criação de pilha. Por exemplo: `AWS::EC2::Instance`, `AWS::EC2::*` ou `Custom::MyCustomInstance`. Use a seguinte sintaxe para descrever tipos de recursos de modelo.

- Para todos os recursos da AWS:

```
AWS::*
```

- Para todos os recursos personalizados:

```
Custom::*
```

- Para um recurso personalizado específico:

```
Custom::logical_ID
```

- Para todos os recursos de um AWS service (Serviço da AWS) específico:

```
AWS::service_name::*
```

- Para um recurso da AWS específico:

```
AWS::service_name::resource_logical_ID
```

Se a lista de tipos de recursos não incluir um recurso que você está criando, a criação da pilha falhará. Por padrão, o CloudFormation concede permissões a todos os tipos de recursos. O IAM usa esse parâmetro para chaves de condição específicas do CloudFormation nas políticas do IAM. Para obter mais informações, consulte Controlar o acesso com o AWS Identity and Access Management.

Tipo: matriz de strings

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 256.

Obrigatório: Não

RoleARN

O nome do recurso da Amazon (ARN) de uma função do IAM assumida pelo CloudFormation para criar a pilha. O CloudFormation usa as credenciais da função para fazer chamadas em seu nome. O CloudFormation sempre usará essa função para todas as futuras operações na pilha. Desde que os usuários tenham permissão para operar na pilha, o CloudFormation usará essa função mesmo que os usuários não tenham permissão para transmiti-la. Certifique-se de que a função conceda a menor quantidade de privilégios.

Se você não especificar um valor, o CloudFormation usará a função anteriormente associada à pilha. Se nenhuma função estiver disponível, o CloudFormation usará uma sessão temporária gerada a partir das suas credenciais de usuário.

Tipo: String

Restrições de tamanho: tamanho mínimo 20. Tamanho máximo de 2.048.

Obrigatório: Não

StackName

O nome que está associado à pilha. O nome deve ser exclusivo na região na qual você estiver criando a pilha.

> **Note**
>
> Um nome de pilha pode conter apenas caracteres alfanuméricos (sensíveis a maiúsculas e minúsculas) e hífens. Ele deve começar com um caractere alfabético e não pode ter mais de 128 caracteres.

Tipo: string

Obrigatório: Sim

StackPolicyBody

Estrutura que contém o corpo da política de pilha. Para obter mais informações, consulte o tópico sobre como Prevenir atualizações para recursos de pilha.

Condicional: é possível especificar o parâmetro `StackPolicyBody` ou `StackPolicyURL`, mas não ambos.

Tipo: String

Restrições de comprimento: tamanho mínimo de 1. Tamanho máximo de 16384.

Obrigatório: Não

StackPolicyURL

Localização de um arquivo contendo a política de pilha. O URL deve apontar para uma política localizada em um bucket do S3 na mesma região que a pilha. O tamanho do arquivo máximo permitido para a política de pilha é de 16 KB.

Condicional: é possível especificar o parâmetro `StackPolicyBody` ou `StackPolicyURL`, mas não ambos.

Tipo: String

Restrições de comprimento: tamanho mínimo de 1. Tamanho máximo de 1350.

Obrigatório: Não

Tags

Pares de chave/valor para associar a essa pilha. O CloudFormation também propaga essas tags para os recursos criados na pilha. Você pode especificar um número máximo de 10 tags.

Tipo: matriz de objetos Tag

Obrigatório: Não

TemplateBody

Estrutura que contém o corpo do modelo com um comprimento mínimo de 1 byte e um comprimento máximo de 51.200 bytes. Para obter mais informações, consulte Anatomia do modelo.

Condicional: é possível especificar o parâmetro `TemplateBody` ou `TemplateURL`, mas não ambos.

Tipo: String

Restrições de comprimento: tamanho mínimo de 1.

Obrigatório: Não

TemplateURL

Localização de um arquivo contendo o corpo do modelo. O URL deve apontar para um modelo que esteja localizado em um bucket do S3. O tamanho máximo permitido para o modelo é 460.800 bytes. Para obter mais informações, consulte Anatomia do modelo.

Condicional: é possível especificar o parâmetro `TemplateBody` ou `TemplateURL`, mas não ambos.

Tipo: String

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Obrigatório: Não

TimeoutInMinutes

O tempo permitido antes que o status da pilha se torne `CREATE_FAILED`. Se `DisableRollback` não estiver definido ou estiver definido como `false`, a pilha será revertida.

Tipo: inteiro

Intervalo válido: valor mínimo de 1.

Obrigatório: Não

Saídas

StackId

Identificador exclusivo da pilha.

Tipo: String

StackStatus

Status atual da pilha.

Tipo: String

Valores Válidos: `CREATE_IN_PROGRESS | CREATE_FAILED | CREATE_COMPLETE | ROLLBACK_IN_PROGRESS | ROLLBACK_FAILED | ROLLBACK_COMPLETE`

```
| DELETE_IN_PROGRESS | DELETE_FAILED | DELETE_COMPLETE |
UPDATE_IN_PROGRESS | UPDATE_COMPLETE_CLEANUP_IN_PROGRESS |
UPDATE_COMPLETE | UPDATE_ROLLBACK_IN_PROGRESS | UPDATE_ROLLBACK_FAILED |
UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS | UPDATE_ROLLBACK_COMPLETE
| REVIEW_IN_PROGRESS
```

Obrigatório: Sim

StackStatusReason

Mensagem de sucesso ou falha associada ao status da pilha.

Tipo: String

Obrigatório: Não

Para obter mais informações, consulte CreateStack.

Considerações sobre segurança

Antes de poder usar a ação `aws:createStack`, você deve atribuir a seguinte política à função assumida do Automation para IAM. Para obter mais informações sobre a função de admissão, consulte Tarefa 1: Criar uma função de serviço para a automação.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "sqs:*",
            "cloudformation:CreateStack",
            "cloudformation:DescribeStacks"
         ],
         "Resource":"*"
      }
   ]
}
```

`aws:createTags`: cria tags para os recursos da AWS

Cria novas tags para instâncias do Amazon Elastic Compute Cloud (Amazon EC2) ou instâncias gerenciadas do AWS Systems Manager.

Entrada

Essa ação oferece suporte para a maioria dos parâmetros `CreateTags` do Amazon EC2 e `AddTagsToResource` do Systems Manager. Para obter mais informações, consulte CreateTags e AddTagsToResource.

O exemplo a seguir mostra como marcar uma Amazon Machine Image (AMI) e uma instância como recursos de produção para um departamento específico.

YAML

```
name: createTags
action: aws:createTags
maxAttempts: 3
onFailure: Abort
inputs:
  ResourceType: EC2
  ResourceIds:
  - ami-9a3768fa
  - i-02951acd5111a8169
  Tags:
  - Key: production
    Value: ''
  - Key: department
    Value: devops
```

JSON

```
{
    "name": "createTags",
    "action": "aws:createTags",
    "maxAttempts": 3,
    "onFailure": "Abort",
    "inputs": {
        "ResourceType": "EC2",
        "ResourceIds": [
            "ami-9a3768fa",
            "i-02951acd5111a8169"
```

```
        ],
        "Tags": [
            {
                "Key": "production",
                "Value": ""
            },
            {
                "Key": "department",
                "Value": "devops"
            }
        ]
    }
}
```

ResourceIds

Os IDs dos recursos a serem marcados. Se o tipo de recurso não for "EC2", esse campo poderá conter apenas um item.

Tipo: String List

Obrigatório: Sim

Tags

As tags a serem associadas aos recursos.

Tipo: lista de mapas

Obrigatório: Sim

ResourceType

Os tipos de recursos a serem marcados. Se não for fornecido, o valor padrão de "EC2" será usado.

Tipo: string

Obrigatório: Não

Valores válidos: `EC2` | `ManagedInstance` | `MaintenanceWindow` | `Parameter`

Saída

Nenhum

`aws:deleteImage`: exclui uma Imagem de máquina da Amazon

Exclua a Amazon Machine Image (AMI) especificada e todos os snapshots relacionados.

Entrada

Esta ação oferece suporte para apenas um parâmetro. Para obter mais informações, consulte a documentação de DeregisterImage e DeleteSnapshot.

YAML

```
name: deleteMyImage
action: aws:deleteImage
maxAttempts: 3
timeoutSeconds: 180
onFailure: Abort
inputs:
  ImageId: ami-12345678
```

JSON

```
{
    "name": "deleteMyImage",
    "action": "aws:deleteImage",
    "maxAttempts": 3,
    "timeoutSeconds": 180,
    "onFailure": "Abort",
    "inputs": {
        "ImageId": "ami-12345678"
    }
}
```

ImageId

O ID da imagem a ser excluída.

Tipo: string

Obrigatório: Sim

Saída

Nenhum

`aws:deleteStack`: exclui uma pilha do AWS CloudFormation

Exclui uma pilha do AWS CloudFormation.

Entrada

YAML

```
name: deleteStack
action: aws:deleteStack
maxAttempts: 1
onFailure: Abort
inputs:
  StackName: "{{stackName}}"
```

JSON

```
{
   "name":"deleteStack",
   "action":"aws:deleteStack",
   "maxAttempts":1,
   "onFailure":"Abort",
   "inputs":{
      "StackName":"{{stackName}}"
   }
}
```

ClientRequestToken

Um identificador exclusivo para essa solicitação de `DeleteStack`. Especifique esse token se você planeja repetir solicitações, para que o CloudFormation saiba que você não está tentando excluir uma pilha com o mesmo nome. Você pode repetir as solicitações `DeleteStack` para verificar se o CloudFormation as recebeu.

Tipo: String

Restrições de tamanho: tamanho mínimo 1. O tamanho máximo é 128.

Padrão: [a-zA-Z][-a-zA-Z0-9]*

Obrigatório: Não

RetainResources.member.N

Essa entrada aplica-se apenas a pilhas que estão em um estado `DELETE_FAILED`. Uma lista de IDs de recursos lógicos para os recursos que você deseja manter. Durante a exclusão, o CloudFormation exclui a pilha, mas não exclui os recursos mantidos.

A retenção de recursos é útil quando você não pode excluir um recurso, como um bucket do S3 não vazio, mas deseja excluir a pilha.

Tipo: matriz de strings

Obrigatório: Não

RoleARN

O nome do recurso da Amazon (ARN) de uma função do AWS Identity and Access Management (IAM) assumida pelo CloudFormation para criar a pilha. O CloudFormation usa as credenciais da função para fazer chamadas em seu nome. O CloudFormation sempre usará essa função para todas as futuras operações na pilha. Desde que os usuários tenham permissão para operar na pilha, o CloudFormation usará essa função mesmo que os usuários não tenham permissão para transmiti-la. Certifique-se de que a função conceda a menor quantidade de privilégios.

Se você não especificar um valor, o CloudFormation usará a função anteriormente associada à pilha. Se nenhuma função estiver disponível, o CloudFormation usará uma sessão temporária gerada a partir das suas credenciais de usuário.

Tipo: String

Restrições de tamanho: tamanho mínimo 20. Tamanho máximo de 2.048.

Obrigatório: Não

StackName

O nome ou o ID de pilha exclusivo que está associado à pilha.

Tipo: string

Obrigatório: Sim

Considerações sobre segurança

Antes de poder usar a ação `aws:deleteStack`, você deve atribuir a seguinte política à função assumida do Automation para IAM. Para obter mais informações sobre a função de admissão, consulte Tarefa 1: Criar uma função de serviço para a automação.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "sqs:*",
            "cloudformation:DeleteStack",
            "cloudformation:DescribeStacks"
         ],
         "Resource":"*"
      }
   ]
}
```

`aws:executeAutomation` – Executa outra automação

Executa uma automação secundária chamando um runbook secundário. Com essa ação, você pode criar runbooks para suas operações mais comuns e fazer referência a esses runbooks durante uma automação. Essa ação pode simplificar seus runbooks, removendo a necessidade de duplicar etapas em runbooks semelhantes.

A automação secundária é executada no contexto do usuário que iniciou a automação primária. Isso significa que a automação secundária usa o mesmo usuário ou perfil do AWS Identity and Access Management (IAM) usado pelo usuário que iniciou a primeira automação.

> **Important**
>
> Se você especificar parâmetros em uma automação secundária que usa uma função assumida (uma função que usa a política iam:passRole), o usuário ou a função que iniciou a automação primária deverá ter permissão para transmitir a função assumida especificada na automação secundária. Para obter mais informações sobre como configurar uma função

assumida para o Automation, consulte Criar perfis de serviço para o Automation usando o console.

Entrada

YAML

```
name: Secondary_Automation
action: aws:executeAutomation
maxAttempts: 3
timeoutSeconds: 3600
onFailure: Abort
inputs:
  DocumentName: secondaryAutomation
  RuntimeParameters:
    instanceIds:
    - i-1234567890abcdef0
```

JSON

```
{
   "name":"Secondary_Automation",
   "action":"aws:executeAutomation",
   "maxAttempts":3,
   "timeoutSeconds":3600,
   "onFailure":"Abort",
   "inputs":{
      "DocumentName":"secondaryAutomation",
      "RuntimeParameters":{
         "instanceIds":[
            "i-1234567890abcdef0"
         ]
      }
   }
}
```

DocumentName

O nome do runbook secundário a ser executado durante a etapa. Para runbooks na mesma Conta da AWS, especifique o nome do runbook. Para runbooks compartilhados de uma

Conta da AWS diferente, especifique o nome do recurso da Amazon (ARN) do runbook. Para obter informações sobre como usar runbooks compartilhados, consulte Usar documentos compartilhados do .

Tipo: string

Obrigatório: Sim

DocumentVersion

A versão do runbook secundário a ser executada. Se não for especificada, o Automation executará a versão padrão do runbook.

Tipo: String

Obrigatório: Não

MaxConcurrency

O número máximo de destinos que podem executar essa tarefa em paralelo. Você pode especificar um número, como 10, ou uma porcentagem, como 10%.

Tipo: String

Obrigatório: Não

MaxErrors

O número de erros permitidos antes que o sistema interrompa a execução de automações em outros destinos. Você pode especificar um número absoluto de erros, como 10, ou uma porcentagem do conjunto de destino, como 10%. Se você especificar 3, por exemplo, o sistema deixará de executar a automação quando o quarto erro for recebido. Se você especificar 0, o sistema deixará de executar a automação em destinos adicionais depois que o primeiro resultado do erro for retornado. Se você executar uma automação em 50 recursos e definir `MaxErrors` para 10%, o sistema interromperá a execução da automação em destinos adicionais quando o sexto erro for recebido.

As automações que já estiverem em execução quando o limite de `MaxErrors` for atingido poderão ser concluídas, mas algumas dessas automações também poderão falhar. Se você precisar garantir que não haverá mais automações com falhas do que o especificado em `MaxErrors`, defina `MaxConcurrency` para 1, para que as automações prossigam uma de cada vez.

Tipo: String

Obrigatório: Não

RuntimeParameters

Parâmetros necessários para o runbook secundário. O mapeamento usa o seguinte formato: {"parameter1" : "value1", "parameter2" : "value2" }

Tipo: mapa

Obrigatório: Não

Tags

Metadados opcional que você atribui a um recurso. É possível especificar um máximo de cinco tags para uma automação.

Tipo: MapList

Obrigatório: Não

TargetLocations

Um local é uma combinação de Regiões da AWS e/ou Contas da AWS onde você deseja executar a automação. Um número mínimo de 1 item deve ser especificado e um número máximo de 100 itens pode ser especificado. Ao especificar um valor para esse parâmetro, as saídas não são retornadas para a automação pai. Se necessário, você deve fazer chamadas subsequentes às operações da API para recuperar a saída das automações filhas.

Tipo: MapList

Obrigatório: Não

TargetMaps

Uma lista de mapeamentos de valor-chave de parâmetros do documento para recursos de destino. Ambos `Targets` e `TargetMaps` não podem ser especificados juntos.

Tipo: MapList

Obrigatório: Não

TargetParameterName

O nome do parâmetro usado como recurso de destino para a automação controlada por taxa. Obrigatório se você especificar `Targets`.

Tipo: String

Obrigatório: Não

Destinos

Uma lista de mapeamentos de valor-chave para recursos de destino. Obrigatório se você especificar `TargetParameterName`.

Tipo: MapList

Obrigatório: Não

Saída

Saída

O resultado gerado pela automação secundária. Você pode fazer referência à saída usando o seguinte formato: *`Secondary_Automation_Step_Name`*.Output

Tipo: StringList

Exemplo:

```
- name: launchNewWindowsInstance
  action: 'aws:executeAutomation'
  onFailure: Abort
  inputs:
    DocumentName: launchWindowsInstance
  nextStep: getNewInstanceRootVolume
- name: getNewInstanceRootVolume
  action: 'aws:executeAwsApi'
  onFailure: Abort
  inputs:
    Service: ec2
    Api: DescribeVolumes
    Filters:
    - Name: attachment.device
      Values:
      - /dev/sda1
    - Name: attachment.instance-id
      Values:
      - '{{launchNewWindowsInstance.Output}}'
  outputs:
```

```
  - Name: rootVolumeId
    Selector: '$.Volumes[0].VolumeId'
    Type: String
  nextStep: snapshotRootVolume
- name: snapshotRootVolume
  action: 'aws:executeAutomation'
  onFailure: Abort
  inputs:
    DocumentName: AWS-CreateSnapshot
    RuntimeParameters:
    VolumeId:
    - '{{getNewInstanceRootVolume.rootVolumeId}}'
    Description:
    - 'Initial root snapshot for {{launchNewWindowsInstance.Output}}'
```

ExecutionId

O ID da automação secundária.

Tipo: String

Status

O status da automação secundária.

Tipo: String

`aws:executeAwsApi`: chama e executa as operações de API do AWS

Chama e executa as operações de API da AWS. A maioria das operações de API são suportadas, embora nem todas elas tenham sido testadas. Operações de API por transmissão, como a operação Get Object, não são compatíveis. Se você não tiver certeza se uma operação de API que deseja usar é uma operação de transmissão, revise a documentação do Boto3 do serviço para determinar se uma API requer entradas ou saídas de transmissão. Atualizamos regularmente a versão do Boto3 que essa ação usa. No entanto, após o lançamento de uma nova versão do Boto3, poderá levar algumas semanas para que as alterações sejam refletidas na ação. A execução de cada ação `aws:executeAwsApi` pode durar, no máximo, 25 segundos. Para obter mais exemplos de como usar essa ação, consulte Exemplos adicionais de runbook.

Entradas

As entradas são definidas pela operação de API que você escolher.

YAML

```
action: aws:executeAwsApi
inputs:
  Service: The official namespace of the service
  Api: The API operation or method name
  API operation inputs or parameters: A value
outputs: # These are user-specified outputs
- Name: The name for a user-specified output key
  Selector: A response object specified by using jsonpath format
  Type: The data type
```

JSON

```
{
   "action":"aws:executeAwsApi",
   "inputs":{
      "Service":"The official namespace of the service",
      "Api":"The API operation or method name",
      "API operation inputs or parameters":"A value"
   },
   "outputs":[ These are user-specified outputs
      {
         "Name":"The name for a user-specified output key",
         "Selector":"A response object specified by using JSONPath format",
         "Type":"The data type"
      }
   ]
}
```

Serviço

O namespace do AWS service (Serviço da AWS) que contém a operação de API que você deseja executar. Você pode visualizar uma lista de namespaces de AWS service (Serviço da AWS) compatíveis em Available services (Serviços disponíveis) no AWS SDK para Python (Boto3). O namespace pode ser encontrado na seção Cliente. Por exemplo, o namespace para o Systems Manager é `ssm`. O namespace do Amazon Elastic Compute Cloud (Amazon EC2) é `ec2`.

Tipo: string

Obrigatório: Sim

API

O nome da operação de API que você deseja executar. Você pode visualizar as operações de API (também chamadas de métodos), escolhendo um serviço na navegação à esquerda na seguinte página de Referência de serviços: Escolha um método na seção Client (Cliente) para o serviço que você deseja invocar. Por exemplo, todas as operações de API (métodos) do Amazon Relational Database Service (Amazon RDS) estão listadas na seguinte página: Amazon RDS methods (Métodos do Amazon RDS).

Tipo: string

Obrigatório: Sim

Entradas de operação da API

Uma ou mais entradas de operação da API. Você pode visualizar as entradas disponíveis (também chamadas de parâmetros), escolhendo um serviço na navegação à esquerda na seguinte página de Referência de serviços. Escolha um método na seção Client (Cliente) para o serviço que você deseja invocar. Por exemplo, todos os métodos de API estão listados na página a seguir: Métodos do Amazon RDS. Escolha o método describe_db_instances e role para baixo para ver os parâmetros disponíveis, como DBInstanceIdentifier, Name (Nome) e Values (Valores).

YAML

```
inputs:
  Service: The official namespace of the service
  Api: The API operation name
  API input 1: A value
  API Input 2: A value
  API Input 3: A value
```

JSON

```
"inputs":{
      "Service":"The official namespace of the service",
      "Api":"The API operation name",
      "API input 1":"A value",
      "API Input 2":"A value",
      "API Input 3":"A value"
}
```

Tipo: determinado pela ação de API escolhida

Obrigatório: Sim

Outputs

As saídas são especificadas pelo usuário com base na resposta da operação da API escolhida.

Nome

Um nome para a saída.

Tipo: string

Obrigatório: Sim

Selector

O JSONPath para um determinado atributo no objeto de resposta. Você pode visualizar os objetos de resposta escolhendo um serviço na navegação à esquerda na seguinte página de Referência de serviços. Escolha um método na seção Client (Cliente) para o serviço que você deseja invocar. Por exemplo, todos os métodos de API estão listados na página a seguir: Métodos do Amazon RDS. Escolha o método describe_db_instances e role para baixo até a seção Response Structure (Estrutura de resposta). DBInstances é listado como um objeto de resposta.

Tipo: inteiro, booliano, String, StringList, StringMap ou MapList

Obrigatório: Sim

Tipo

O tipo de dados para o elemento de resposta.

Tipo: Varia

Obrigatório: Sim

`aws:executeScript` – Executa um script

Executa o script do Python ou PowerShell fornecido, usando o runtime e o handler especificados. Cada ação `aws:executeScript` pode ser executada por uma duração máxima de dez minutos (600 segundos). Você pode limitar o tempo limite especificando o parâmetro `timeoutSeconds` para uma etapa `aws:executeScript`.

Use instruções de retorno em sua função para adicionar saídas à carga útil da saída. Para obter exemplos de definição de saídas para sua ação `aws:executeScript`, consulte Exemplo 2: Runbook com script. Também é possível enviar o resultado das ações do `aws:executeScript` nos runbooks para o grupo de logs do Amazon CloudWatch Logs especificado. Para obter mais informações, consulte Registro de saída de ações do Automation em log com o CloudWatch Logs.

Se quiser enviar a saída de ações `aws:executeScript` ao CloudWatch Logs ou se os scripts que você especificar para ações `aws:executeScript` chamarem opções de API da AWS, um perfil de serviço do AWS Identity and Access Management (IAM) (ou uma função de recepção) sempre será necessário para executar o runbook.

A ação `aws:executeScript` contém os módulos pré-instalados do PowerShell Core a seguir.

- Microsoft.PowerShell.Host
- Microsoft.PowerShell.Management
- Microsoft.PowerShell.Security
- Microsoft.PowerShell.Utility
- PackageManagement
- PowerShellGet

Para usar módulos do PowerShell Core que não estão pré-instalados, o script deve instalar o módulo com o sinalizador `-Force`, conforme mostrado no comando a seguir. Não há suporte ao módulo `AWSPowerShell.NetCore`. Substitua *`ModuleName`* pelo nome do módulo que deseja instalar.

```
Install-Module ModuleName -Force
```

Para usar cmdlets do PowerShell Core em seu script, recomendamos o uso dos módulos `AWS.Tools`, conforme mostrado nos comandos a seguir. Substitua cada *`espaço reservado para recurso de exemplo`* por suas próprias informações.

- Cmdlets do Amazon S3.

```
Install-Module AWS.Tools.S3 -Force
Get-S3Bucket -BucketName amzn-s3-demo-bucket
```

- cmdlets do Amazon EC2

```
Install-Module AWS.Tools.EC2 -Force
```

```
Get-EC2InstanceStatus -InstanceId instance-id
```

- cmdlets do AWS Tools for Windows PowerShell comuns ou independentes de serviço.

```
Install-Module AWS.Tools.Common -Force
Get-AWSRegion
```

Se o script inicializar novos objetos além de usar cmdlets do PowerShell Core, você também deverá importar o módulo conforme mostrado no comando a seguir.

```
Install-Module AWS.Tools.EC2 -Force
Import-Module AWS.Tools.EC2

$tag = New-Object Amazon.EC2.Model.Tag
$tag.Key = "Tag"
$tag.Value = "TagValue"

New-EC2Tag -Resource i-02573cafcfEXAMPLE -Tag $tag
```

Para obter exemplos de instalação e importação dos módulos `AWS.Tools` e do uso de cmdlets do PowerShell Core em runbooks, consulte Experiência de design visual para runbooks de automação.

Entrada

Forneça as informações necessárias para executar o script. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

> **Note**
>
> O anexo de um script Python pode ser um arquivo .py ou um arquivo .zip contendo o script. É necessário armazenar os scripts do PowerShell em arquivos .zip.

YAML

```
action: "aws:executeScript"
inputs:
 Runtime: runtime
 Handler: "functionName"
```

```
 InputPayload:
  scriptInput: '{{parameterValue}}'
 Script: |-
   def functionName(events, context):
   ...
 Attachment: "scriptAttachment.zip"
```

JSON

```
{
    "action": "aws:executeScript",
    "inputs": {
        "Runtime": "runtime",
        "Handler": "functionName",
        "InputPayload": {
            "scriptInput": "{{parameterValue}}"
        },
        "Attachment": "scriptAttachment.zip"
    }
}
```

Runtime

A linguagem do runtime a ser usada para executar o script fornecido. O `aws:executeScript` oferece suporte a scripts Python 3.7 (`python3.7`), Python 3.8 (`python3.8`), Python 3.9 (`python3.9`)Python 3.10 (`python3.10`), Python 3.11 (`python3.11`) PowerShell Core 6.0 (`dotnetcore2.1`) e PowerShell 7.0 (`dotnetcore3.1`) e a scripts PowerShell 7.4 (`dotnet8`).

Valores compatíveis: **`python3.7`** | **`python3.8`** | **`python3.9`** | **`python3.10`** | **`python3.11`** | **`PowerShell Core 6.0`** | **`PowerShell 7.0`** | **`PowerShell 7.4`**

Tipo: string

Obrigatório: Sim

> **Note**
>
> Para runtimes Python, o ambiente fornece 512 MB de memória e 512 MB de espaço em disco. Para runtimes PowerShell, o ambiente fornece 1024 MB de memória e 512 MB de espaço em disco.

Manipulador

Escolha o nome da função. É necessário garantir que a função definida no manipulador tenha dois parâmetros, `events` e `context`. O runtime do PowerShell não oferece suporte a este parâmetro.

Tipo: String

Obrigatório: sim (Python) | sem suporte (PowerShell)

InputPayload

Um objeto JSON ou YAML que será passado para o primeiro parâmetro do manipulador. Isso pode ser usado para passar dados de entrada para o script.

Tipo: String

Obrigatório: não

Python

```
description: Tag an instance
schemaVersion: '0.3'
assumeRole: '{{AutomationAssumeRole}}'
parameters:
    AutomationAssumeRole:
        type: String
        description: '(Required) The Amazon Resource Name (ARN) of the IAM role
 that allows Automation to perform the actions on your behalf. If no role is
 specified, Systems Manager Automation uses your IAM permissions to operate this
 runbook.'
    InstanceId:
        type: String
        description: (Required) The ID of the EC2 instance you want to tag.
mainSteps:
  - name: tagInstance
    action: 'aws:executeScript'
    inputs:
        Runtime: "python3.11"
        Handler: tagInstance
        InputPayload:
            instanceId: '{{InstanceId}}'
        Script: |-
          def tagInstance(events,context):
```

```
            import boto3

            #Initialize client
            ec2 = boto3.client('ec2')
            instanceId = events['instanceId']
            tag = {
                "Key": "Env",
                "Value": "ExamplePython"
            }
            print(f"Adding tag {tag} to instance id {instanceId}")
            ec2.create_tags(
                Resources=[instanceId],
                Tags=[tag]
            )
            return tag
    outputs:
      - Type: String
        Name: TagKey
        Selector: $.Payload.Key
outputs:
  - tagInstance.TagKey
```

PowerShell

```
description: Tag an instance
schemaVersion: '0.3'
assumeRole: '{{AutomationAssumeRole}}'
parameters:
  AutomationAssumeRole:
    type: String
    description: (Required) The Amazon Resource Name (ARN) of the IAM role
 that allows Automation to perform the actions on your behalf. If no role is
 specified, Systems Manager Automation uses your IAM permissions to operate this
 runbook.
  InstanceId:
    type: String
    description: (Required) The ID of the EC2 instance you want to tag.
mainSteps:
  - name: tagInstance
    action: aws:executeScript
    isEnd: true
    inputs:
      Runtime: PowerShell 7.0
```

```
      InputPayload:
        instanceId: '{{InstanceId}}'
      Script: |-
        Install-Module AWS.Tools.EC2 -Force
        Import-Module AWS.Tools.EC2

        $input = $env:InputPayload | ConvertFrom-Json

        $tag = New-Object Amazon.EC2.Model.Tag
        $tag.Key = "Env"
        $tag.Value = "ExamplePowerShell"

        Write-Information "Adding tag key: $($tag.Key) and value: $($tag.Value)
 to instance id $($input.instanceId)"
        New-EC2Tag -Resource $input.instanceId -Tag $tag

        return $tag
    outputs:
      - Type: String
        Name: TagKey
        Selector: $.Payload.Key
outputs:
  - tagInstance.TagKey
```

Script

Um script incorporado que você precisa executar durante a execução da automação.

Tipo: String

Obrigatório: não (Python) | sim (PowerShell)

Attachment

O nome de um arquivo de script autônomo ou arquivo .zip que pode ser invocado pela ação. Especifique o mesmo valor do `Name` do arquivo de anexo de documento especificado no parâmetro de solicitação de `Attachments`. Para obter mais informações, consulte Anexos na Referência de API do AWS Systems Manager. Se você estiver fornecendo um script usando um anexo, você deverá definir uma seção `files` nos elementos de nível superior do seu runbook. Para obter mais informações, consulte Versão 0.3 do esquema.

Para invocar um arquivo para Python, use o formato `filename.method_name` no `Handler`.

> **Note**
>
> O anexo de um script Python pode ser um arquivo .py ou um arquivo .zip contendo o script. É necessário armazenar os scripts do PowerShell em arquivos .zip.

Ao incluir bibliotecas Python em seu anexo, recomendamos adicionar um arquivo `__init__.py` em cada diretório do módulo. Isso permite que você importe os módulos da biblioteca em seu anexo dentro do conteúdo do script. Por exemplo: `from library import module`

Tipo: String

Obrigatório: Não

Saída

Carga útil

A representação JSON do objeto retornado pela função. Até 100 KB é retornado. Se você gerar uma lista, haverá o retorno de no máximo de 100 itens.

Usar anexos com aws:executeScript

Os anexos fornecem uma maneira poderosa de empacotar e reutilizar scripts complexos, vários módulos e dependências externas com suas ações `aws:executeScript`. Use anexos quando precisar:

- Empacotar vários módulos Python ou scripts PowerShell juntos.
- Reutilizar a mesma lógica de script em vários runbooks.
- Incluir bibliotecas ou dependências externas em seus scripts.
- Manter a definição do seu runbook limpa separando a lógica complexa do script.
- Compartilhar pacotes de scripts entre equipes ou fluxos de trabalho de automação.

Estrutura e empacotamento de anexos

Você pode anexar arquivos únicos ou pacotes zip contendo vários arquivos. A estrutura depende do seu caso de uso:

Anexos de arquivo único

Para scripts simples, você pode anexar um único arquivo `.py` (Python) ou um arquivo `.zip` contendo um único script PowerShell.

Pacotes de vários módulos

Para automação complexa que requer vários módulos, crie um pacote zip com a seguinte estrutura recomendada:

```
my-automation-package.zip
### main.py                     # Entry point script
### utils/
#   ### __init__.py            # Required for Python module imports
#   ### helper_functions.py    # Utility functions
#   ### aws_operations.py      # AWS-specific operations
### config/
#   ### __init__.py
#   ### settings.py            # Configuration settings
### requirements.txt           # Optional: document dependencies
```

> **Important**
>
> Para pacotes Python, você deve incluir um arquivo `__init__.py` vazio em cada diretório que contém módulos Python. Isso permite importar módulos usando a sintaxe de importação padrão do Python, como `from utils import helper_functions`.

Estrutura do pacote PowerShell

Os anexos PowerShell devem ser empacotados em arquivos zip com a seguinte estrutura:

```
my-powershell-package.zip
### Main.ps1                    # Entry point script
### Modules/
#   ### HelperFunctions.ps1    # Utility functions
#   ### AWSOperations.ps1      # AWS-specific operations
### Config/
    ### Settings.ps1           # Configuration settings
```

Criar runbooks com anexos

Siga estas etapas para criar runbooks que usam anexos:

1. Faça o upload do seu anexo para o Amazon S3

   Carregue seu arquivo de script ou pacote zip em um bucket do S3 que possa ser acessado por seu perfil de automação. Anote o URI do S3 para uso na próxima etapa.

   ```
   aws s3 cp my-automation-package.zip s3://my-automation-bucket/scripts/
   ```

2. Calcule a soma de verificação do anexo

   Calcule a soma de verificação SHA-256 do arquivo de anexo para verificação de segurança:

   ```
   # Linux/macOS
   shasum -a 256 my-automation-package.zip

   # Windows PowerShell
   Get-FileHash -Algorithm SHA256 my-automation-package.zip
   ```

3. Defina a seção de arquivos em seu runbook

   Adicione uma seção `files` no nível superior do seu runbook para fazer referência ao seu anexo:

   ```
   files:
     my-automation-package.zip:
       sourceType: "S3"
       sourceInfo:
         path: "s3://my-automation-bucket/scripts/my-automation-package.zip"
         checksums:
           sha256: "your-calculated-checksum-here"
   ```

4. Faça referência ao anexo em sua etapa executeScript

   Use o parâmetro `Attachment` para fazer referência ao seu arquivo enviado:

   ```
   - name: runMyScript
     action: aws:executeScript
     inputs:
       Runtime: python3.11
       Handler: main.process_data
   ```

```
    Attachment: my-automation-package.zip
    InputPayload:
      inputData: "{{InputParameter}}"
```

Exemplos de anexos aws:executeScript

Os exemplos a seguir demonstram maneiras diferentes de usar anexos com a ação `aws:executeScript`.

Exemplo 1: anexo de arquivo único

Este exemplo mostra como usar um único arquivo Python como anexo para processar dados da instância do EC2.

Arquivo do anexo: process_instance.py

Crie um arquivo Python com o seguinte conteúdo:

```
import boto3
import json

def process_instance_data(events, context):
    """Process EC2 instance data and return formatted results."""
    try:
        instance_id = events.get('instanceId')
        if not instance_id:
            raise ValueError("instanceId is required")

        ec2 = boto3.client('ec2')

        # Get instance details
        response = ec2.describe_instances(InstanceIds=[instance_id])
        instance = response['Reservations'][0]['Instances'][0]

        # Format the response
        result = {
            'instanceId': instance_id,
            'instanceType': instance['InstanceType'],
            'state': instance['State']['Name'],
            'availabilityZone': instance['Placement']['AvailabilityZone'],
            'tags': {tag['Key']: tag['Value'] for tag in instance.get('Tags', [])}
        }
```

```
        print(f"Successfully processed instance {instance_id}")
        return result

    except Exception as e:
        print(f"Error processing instance: {str(e)}")
        raise
```

Runbook completo

Aqui está o runbook completo que usa o anexo de arquivo único:

```
description: Process EC2 instance data using single file attachment
schemaVersion: '0.3'
assumeRole: '{{AutomationAssumeRole}}'
parameters:
  AutomationAssumeRole:
    type: String
    description: (Required) IAM role for automation execution
  InstanceId:
    type: String
    description: (Required) EC2 instance ID to process

files:
  process_instance.py:
    sourceType: "S3"
    sourceInfo:
      path: "s3://my-automation-bucket/scripts/process_instance.py"
      checksums:
        sha256: "abc123def456..."

mainSteps:
  - name: processInstance
    action: aws:executeScript
    inputs:
      Runtime: python3.11
      Handler: process_instance.process_instance_data
      Attachment: process_instance.py
      InputPayload:
        instanceId: '{{InstanceId}}'
    outputs:
      - Type: StringMap
        Name: InstanceData
        Selector: $.Payload
```

```
outputs:
  - processInstance.InstanceData
```

Exemplo 2: pacote de vários módulos

Este exemplo demonstra o uso de um pacote zip contendo vários módulos Python para operações complexas de bucket do S3.

Estrutura do pacote

Crie um pacote zip com a seguinte estrutura:

```
s3-operations.zip
### main.py
### utils/
#   ### __init__.py
#   ### s3_helper.py
#   ### validation.py
### config/
    ### __init__.py
    ### settings.py
```

main.py (ponto de entrada)

O script principal que orquestra as operações:

```
from utils.s3_helper import S3Operations
from utils.validation import validate_bucket_name
from config.settings import get_default_settings

def cleanup_s3_bucket(events, context):
    """Clean up S3 bucket based on specified criteria."""
    try:
        bucket_name = events.get('bucketName')
        max_age_days = events.get('maxAgeDays', 30)

        # Validate inputs
        if not validate_bucket_name(bucket_name):
            raise ValueError(f"Invalid bucket name: {bucket_name}")

        # Initialize S3 operations
        s3_ops = S3Operations()
```

```
        settings = get_default_settings()

        # Perform cleanup
        deleted_objects = s3_ops.delete_old_objects(
            bucket_name,
            max_age_days,
            settings['dry_run']
        )

        result = {
            'bucketName': bucket_name,
            'deletedCount': len(deleted_objects),
            'deletedObjects': deleted_objects[:10],  # Return first 10 for brevity
            'dryRun': settings['dry_run']
        }

        print(f"Cleanup completed for bucket {bucket_name}")
        return result

    except Exception as e:
        print(f"Error during S3 cleanup: {str(e)}")
        raise
```

Solução de problemas de anexos aws:executeScript

Use as diretrizes a seguir para resolver problemas comuns com anexos `aws:executeScript`:

Erros de importação de módulos

Se você receber erros de importação ao usar pacotes com vários módulos:

- Certifique-se de ter incluído um arquivo `__init__.py` vazio em cada diretório contendo módulos Python.
- Verifique se suas instruções de importação correspondem à estrutura real de arquivos e diretórios em seu pacote zip.
- Use importações relativas (por exemplo, `from .utils import helper`) ou importações absolutas (por exemplo, `from utils import helper`) de forma consistente.

Erros de anexo não encontrado

Se sua automação não conseguir encontrar o anexo:

- Verifique se o valor do parâmetro `Attachment` corresponde exatamente à chave em sua seção `files`.
- Verifique se o caminho e o nome do arquivo do bucket do S3 estão corretos na seção `files`.
- Certifique-se de que seu perfil de automação tenha permissão `s3:GetObject` para a localização do anexo do S3.
- Verifique se a soma de verificação em seu runbook corresponde à soma de verificação real do arquivo.

Erros de função do manipulador

Se você receber erros relacionados ao manipulador:

- Em Python: use o formato `filename.function_name` no parâmetro `Handler` (por exemplo, `main.process_data`).
- Certifique-se de que sua função de manipulador aceite exatamente dois parâmetros: `events` e `context`.
- Em PowerShell: não especifique um parâmetro `Handler`; o script é executado diretamente.

Falhas de execução de script

Se o script falhar durante a execução:

- Verifique o histórico de execução da automação para mensagens de erro detalhadas e rastreamentos de pilha.
- Use instruções `print()` (Python) ou `Write-Information` (PowerShell) para adicionar a saída de depuração.
- Verifique se todas as permissões AWS necessárias foram concedidas à sua função de automação.
- Teste a lógica do seu script localmente antes de empacotá-lo como um anexo.

Tratamento de erros e códigos de saída

Para lidar adequadamente com erros e códigos de saída de retorno:

- Em Python: use `raise Exception("error message")` para indicar falha no script.
- Em PowerShell: use `throw "error message"` ou `Write-Error` para indicar falha.

- Retorne dados estruturados de suas funções para fornecer informações detalhadas de êxito/falha.
- Use blocos try-catch para lidar com exceções de forma elegante e fornecer mensagens de erro significativas.

`aws:executeStateMachine` – Executa uma máquina de estado do AWS Step Functions

Executa uma máquina de estado do AWS Step Functions.

Entrada

Essa ação oferece suporte para a maioria dos parâmetros da operação da API StartExecution do Step Functions.

Permissões obrigatórias do AWS Identity and Access Management (IAM)

- `states:DescribeExecution`
- `states:StartExecution`
- `states:StopExecution`

YAML

```
name: executeTheStateMachine
action: aws:executeStateMachine
inputs:
  stateMachineArn: StateMachine_ARN
  input: '{"parameters":"values"}'
  name: name
```

JSON

```
{
    "name": "executeTheStateMachine",
    "action": "aws:executeStateMachine",
    "inputs": {
        "stateMachineArn": "StateMachine_ARN",
        "input": "{\"parameters\":\"values\"}",
        "name": "name"
    }
}
```

stateMachineArn

O nome do recurso da Amazon (ARN) da máquina de estado do Step Functions.

Tipo: string

Obrigatório: Sim

name

O nome da execução.

Tipo: string

Obrigatório: Não

input

Uma string que contém os dados de entrada JSON da execução.

Tipo: string

Obrigatório: Não

Outputs

As saídas a seguir são predefinidas para essa ação.

executionArn

O ARN da execução.

Tipo: string

input

A string que contém os dados de entrada JSON da execução. As restrições de comprimento se aplicam ao tamanho da carga útil e são expressas como bytes na codificação UTF-8.

Tipo: string

name

O nome da execução.

Tipo: string

output

Os dados de saída JSON da execução. As restrições de comprimento se aplicam ao tamanho da carga útil e são expressas como bytes na codificação UTF-8.

Tipo: string

startDate

A data em que a execução é iniciada.

Tipo: string

stateMachineArn

O ARN da máquina de estado executada.

Tipo: string

status

O status atual da execução.

Tipo: string

stopDate

Se a execução já tiver terminado, a data em que a execução foi interrompida.

Tipo: string

`aws:invokeWebhook` — Invoque uma integração de webhook do Automation

Invoca a integração de webhook do Automation especificada. Para obter informações sobre como criar integrações do Automation, consulte Criação de integrações de webhooks para o Automation.

> **Note**
>
> Para usar a ação `aws:invokeWebhook`, seu usuário ou perfil de serviço deve permitir as ações a seguir:
>
> - ssm:GetParameter
> - kms:Decrypt

A permissão para a operação AWS Key Management Service (AWS KMS) `Decrypt` só é necessária se você usar uma chave gerenciada pelo cliente para criptografar o parâmetro para sua integração.

Entrada

Forneça as informações para a integração do Automation que você deseja invocar.

YAML

```
action: "aws:invokeWebhook"
inputs:
 IntegrationName: "exampleIntegration"
 Body: "Request body"
```

JSON

```
{
    "action": "aws:invokeWebhook",
    "inputs": {
        "IntegrationName": "exampleIntegration",
        "Body": "Request body"
    }
}
```

IntegrationName

O nome da integração do Automation. Por exemplo, `exampleIntegration`. A integração que você especificar já deve existir.

Tipo: string

Obrigatório: Sim

Corpo

A carga útil que você deseja enviar quando a integração do webhook for invocada.

Tipo: string

Obrigatório: Não

Saída

Resposta

O texto recebido da resposta do provedor de webhook.

ResponseCode

O código de status de HTTP recebido da resposta do provedor de webhook.

`aws:invokeLambdaFunction` – Invoca uma função do AWS Lambda

Chama a função do AWS Lambda especificada.

> **Note**
>
> Cada ação `aws:invokeLambdaFunction` pode ser executada por uma duração máxima de cinco minutos (300 segundos). Você pode limitar o tempo limite especificando o parâmetro `timeoutSeconds` para uma etapa `aws:invokeLambdaFunction`.

Entrada

Essa ação oferece suporte para a maioria dos parâmetros invocados do serviço Lambda. Para obter mais informações, consulte Invoke.

YAML

```
name: invokeMyLambdaFunction
action: aws:invokeLambdaFunction
maxAttempts: 3
timeoutSeconds: 120
onFailure: Abort
inputs:
  FunctionName: MyLambdaFunction
```

JSON

```
{
```

```
    "name": "invokeMyLambdaFunction",
    "action": "aws:invokeLambdaFunction",
    "maxAttempts": 3,
    "timeoutSeconds": 120,
    "onFailure": "Abort",
    "inputs": {
        "FunctionName": "MyLambdaFunction"
    }
}
```

FunctionName

O nome da função do Lambda. Essa função deve existir.

Tipo: string

Obrigatório: Sim

Qualificador

A versão da função ou nome do alias.

Tipo: String

Obrigatório: Não

InvocationType

O tipo de invocação. O valor padrão é `RequestResponse`.

Tipo: string

Valores válidos: `Event` | `RequestResponse` | `DryRun`

Obrigatório: Não

LogType

Se o valor padrão for `Tail`, o tipo de invocação deverá ser `RequestResponse`. O Lambda retorna os últimos 4 KB de dados de log produzidos pela sua função Lambda, codificados em base64.

Tipo: string

Valores válidos: `None` | `Tail`

Obrigatório: Não

ClientContext

As informações específicas do cliente.

Obrigatório: Não

InputPayload

Um objeto JSON ou YAML que será transmitido para o primeiro parâmetro do manipulador. É possível usar essa entrada a fim de transmitir dados para a função. Essa entrada fornece mais flexibilidade e suporte do que a entrada `Payload` legada. Se você definir `InputPayload` e `Payload` para a ação, `InputPayload` tem precedência e o valor `Payload` não será usado.

Tipo: StringMap

Obrigatório: Não

Carga útil

Uma string JSON que é transmitida para o primeiro parâmetro do manipulador. É possível usá-la a fim de transmitir dados de entrada para a função. Recomendamos usar a entrada `InputPayload` para funcionalidade adicional.

Tipo: String

Obrigatório: Não

Saída

StatusCode

Código de status do HTTP

FunctionError

Se presente, ele indica que ocorreu um erro durante a execução da função. Os detalhes do erro estão incluídos na carga útil da resposta.

LogResult

Os logs codificados pelo base64 para a invocação da função Lambda. Os logs estarão presentes apenas se o tipo de invocação for `RequestResponse` e se tiverem sido solicitados.

Carga útil

A representação JSON do objeto retornado pela função Lambda. A carga útil estará presente apenas se o tipo de invocação for `RequestResponse`.

A seguir é mostrada uma parte do runbook `AWS-PatchInstanceWithRollback` que demonstra como fazer referência a saídas da ação `aws:invokeLambdaFunction`.

YAML

```
- name: IdentifyRootVolume
  action: aws:invokeLambdaFunction
  inputs:
    FunctionName: "IdentifyRootVolumeLambda-{{automation:EXECUTION_ID}}"
    Payload: '{"InstanceId": "{{InstanceId}}"}'
- name: PrePatchSnapshot
  action: aws:executeAutomation
  inputs:
    DocumentName: "AWS-CreateSnapshot"
    RuntimeParameters:
      VolumeId: "{{IdentifyRootVolume.Payload}}"
      Description: "ApplyPatchBaseline restoration case contingency"
```

JSON

```
{
    "name": "IdentifyRootVolume",
    "action": "aws:invokeLambdaFunction",
    "inputs": {
      "FunctionName": "IdentifyRootVolumeLambda-{{automation:EXECUTION_ID}}",
      "Payload": "{\"InstanceId\": \"{{InstanceId}}\"}"
    }
  },
  {
    "name": "PrePatchSnapshot",
    "action": "aws:executeAutomation",
    "inputs": {
      "DocumentName": "AWS-CreateSnapshot",
      "RuntimeParameters": {
        "VolumeId": "{{IdentifyRootVolume.Payload}}",
        "Description": "ApplyPatchBaseline restoration case contingency"
      }
```

```
    }
  }
```

`aws:loop`: itera nas etapas de uma automação

Essa ação itera em um subconjunto de etapas em um runbook de automação. Você pode escolher um loop de estilo do `while` ou `for each`. Para construir um loop do `while`, use o parâmetro de entrada `LoopCondition`. Para construir um loop `for each`, use os parâmetros de entrada `Iterators` e `IteratorDataType`. Ao usar uma ação `aws:loop`, especifique somente o parâmetro de entrada `Iterators` ou `LoopCondition`. O número máximo de iterações é 100.

A propriedade `onCancel` só pode ser usada para etapas definidas em um loop. A propriedade `onCancel` não é compatível com a ação `aws:loop`. A propriedade `onFailure` pode ser usada para uma ação `aws:loop`, mas ela só será usada se ocorrer um erro inesperado causando a falha da etapa. Se você definir propriedades `onFailure` para as etapas em um loop, a ação `aws:loop` herdará essas propriedades e reagirá adequadamente quando ocorrer uma falha.

Exemplos

Veja a seguir exemplos de como criar os diferentes tipos de ações de loop.

do while

```
name: RepeatMyLambdaFunctionUntilOutputIsReturned
action: aws:loop
inputs:
    Steps:
    - name: invokeMyLambda
        action: aws:invokeLambdaFunction
        inputs:
        FunctionName: LambdaFunctionName
        outputs:
        - Name: ShouldRetry
            Selector: $.Retry
            Type: Boolean
    LoopCondition:
        Variable: "{{ invokeMyLambda.ShouldRetry }}"
        BooleanEquals: true
    MaxIterations: 3
```

for each

```
name: stopAllInstancesWithWaitTime
action: aws:loop
inputs:
    Iterators: "{{ DescribeInstancesStep.InstanceIds }}"
    IteratorDataType: "String"
    Steps:
    - name: stopOneInstance
        action: aws:changeInstanceState
        inputs:
        InstanceIds:
            - "{{stopAllInstancesWithWaitTime.CurrentIteratorValue}}"
        CheckStateOnly: false
        DesiredState: stopped
    - name: wait10Seconds
        action: aws:sleep
        inputs:
        Duration: PT10S
```

Entrada

Veja a entrada a seguir.

Iteradores

A lista de itens sobre os quais as etapas devem ser iteradas. O número máximo de iteradores é 100.

Tipo: StringList

Obrigatório: Não

IteratorDataType

Um parâmetro opcional para especificar o tipo de dados dos `Iterators`. Um valor para esse parâmetro pode ser fornecido junto com o parâmetro de entrada `Iterators`. Se você não especificar um valor para esse parâmetro e `Iterators`, deverá especificar um valor para o parâmetro `LoopCondition`.

Tipo: string

Valores válidos: Boolean | Integer | String | StringMap

Padrão: String

Obrigatório: Não

LoopCondition

Consiste em uma `Variable` e uma condição do operador a ser avaliada. Se você não especificar um valor para esse parâmetro, deverá especificar um valor para os parâmetros `Iterators` e `IteratorDataType`. Você pode usar avaliações complexas de operadores usando uma combinação dos operadores `And`, `Not` e `Or`. A condição é avaliada após a conclusão das etapas do loop. Se a condição for `true` e o valor `MaxIterations` não tiver sido atingido, as etapas do loop serão executadas novamente. As condições do operador são as seguintes:

Operações de string

- StringEquals
- EqualsIgnoreCase
- StartsWith
- EndsWith
- Contém

Operações numéricas

- NumericEquals
- NumericGreater
- NumericLesser
- NumericGreaterOrEquals
- NumericLesser
- NumericLesserOrEquals

Operação booleana

- BooleanEquals

Tipo: StringMap

Obrigatório: Não

MaxIterations

O número máximo de vezes que as etapas do loop são executadas. Quando o valor especificado para essa entrada é atingido, o loop para de ser executado mesmo se `LoopCondition` ainda for `true` ou se houver objetos restantes no parâmetro `Iterators`.

Tipo: número inteiro

Valores válidos: 1 a 100

Obrigatório: Não

Etapas

A lista de etapas a serem executadas no loop. Elas funcionam como um runbook aninhado. Nessas etapas, você pode acessar o valor atual do iterador de um loop `for each` usando a sintaxe `{{loopStepName.CurrentIteratorValue}}`. Você também pode acessar um valor inteiro da iteração atual para os dois tipos de loop usando a sintaxe `{{loopStepName.CurrentIteration}}`.

Type: Lista de etapas

Obrigatório: Sim

Saída

CurrentIteration

A iteração atual do loop como um número inteiro. Os valores de iteração começam em 1.

Tipo: número inteiro

CurrentIteratorValue

O valor do iterador atual como uma string. Essa saída só está presente em loops `for each`.

Tipo: string

`aws:pause` – Pausa uma automação

Essa ação pausa a automação. Depois de pausada, o status da automação é Waiting (Em espera). Para continuar a automação, use a operação da API SendAutomationSignal com o tipo de sinal

`Resume`. Recomendamos usar a ação `aws:sleep` ou `aws:approve` para um controle mais granular de seus fluxos de trabalho.

> **Note**
>
> O tempo limite padrão para essa ação é de 7 dias (604.800 segundos) e o valor máximo é de 30 dias (2.592.000 segundos). Você pode limitar ou prolongar o tempo limite especificando o parâmetro `timeoutSeconds` para uma etapa `aws:pause`.

Entrada

Veja a entrada a seguir.

YAML

```
name: pauseThis
action: aws:pause
timeoutSeconds: 1209600
inputs: {}
```

JSON

```
{
    "name": "pauseThis",
    "action": "aws:pause",
    "timeoutSeconds": "1209600",
    "inputs": {}
}
```

Saída

Nenhum

`aws:runCommand` – Executa um comando em uma instância gerenciada

Executa os comandos especificados.

> **Note**
>
> A automação comporta apenas a saída de uma ação do AWS Systems Manager Run Command. Um runbook pode incluir várias ações do Run Command, mas apenas uma ação de cada vez pode gerar uma saída.

Entrada

Essa ação oferece suporte para a maioria dos parâmetros de comando de envio. Para obter mais informações, consulte SendCommand.

YAML

```
- name: checkMembership
  action: 'aws:runCommand'
  inputs:
    DocumentName: AWS-RunPowerShellScript
    InstanceIds:
      - '{{InstanceIds}}'
    Parameters:
      commands:
        - (Get-WmiObject -Class Win32_ComputerSystem).PartOfDomain
```

JSON

```
{
    "name": "checkMembership",
    "action": "aws:runCommand",
    "inputs": {
        "DocumentName": "AWS-RunPowerShellScript",
        "InstanceIds": [
            "{{InstanceIds}}"
        ],
        "Parameters": {
            "commands": [
                "(Get-WmiObject -Class Win32_ComputerSystem).PartOfDomain"
            ]
        }
    }
}
```

DocumentName

Se o documento do tipo Comando for de sua propriedade ou da AWS, especifique o nome do documento. Se você estiver usando um documento compartilhado com você por uma Conta da AWSdiferente, especifique o nome do recurso da Amazon (ARN) do documento. Para obter mais informações sobre o uso de documentos compartilhados, consulte Usar documentos compartilhados do .

Tipo: string

Obrigatório: Sim

InstanceIds

Os IDs de instâncias em que deseja que o comando seja executado. Você pode especificar um máximo de 50 IDs.

Você também pode usar o pseudoparâmetro `{{RESOURCE_ID}}` no lugar de IDs de instância para executar o comando em todas as instâncias no grupo de destino. Para obter mais informações sobre pseudoparâmetros, consulte Usar pseudoparâmetros ao registrar tarefas da janela de manutenção.

Outra alternativa é enviar comandos para uma frota de instâncias usando o parâmetro `Targets`.. O parâmetro `Targets` aceita tags do Amazon Elastic Compute Cloud (Amazon EC2). Para obter mais informações sobre como usar o parâmetro `Targets`, consulte Execução de comandos em escala.

Tipo: StringList

Obrigatório: não (Se não especificar InstanceIds ou usar o pseudoparâmetro `{{RESOURCE_ID}}`, você deverá especificar o parâmetro `Targets`.)

Destinos

Vários critérios de pesquisa que apontam para instâncias usando uma combinação de chave-valor especificada. As `Targets` serão necessárias se você não fornecer um ou mais IDs de instância na chamada. Para obter mais informações sobre como usar o parâmetro `Targets`, consulte Execução de comandos em escala.

Tipo: MapList (o esquema do mapa na lista deve corresponder ao objeto). Para obter informações, consulte Target na Referência de API do AWS Systems Manager.

Obrigatório: não (Se não especificar `Targets`, você deverá especificar os IDs da instância ou usar o pseudoparâmetro `{{RESOURCE_ID}}`.)

Veja um exemplo a seguir.

YAML

```
- name: checkMembership
  action: aws:runCommand
  inputs:
    DocumentName: AWS-RunPowerShellScript
    Targets:
      - Key: tag:Stage
        Values:
          - Gamma
          - Beta
      - Key: tag-key
        Values:
          - Suite
    Parameters:
      commands:
        - (Get-WmiObject -Class Win32_ComputerSystem).PartOfDomain
```

JSON

```
{
    "name": "checkMembership",
    "action": "aws:runCommand",
    "inputs": {
        "DocumentName": "AWS-RunPowerShellScript",
        "Targets": [
            {
                "Key": "tag:Stage",
                "Values": [
                    "Gamma", "Beta"
                ]
            },
            {
                "Key": "tag:Application",
                "Values": [
                    "Suite"
                ]
            }
        ],
```

```
        "Parameters": {
            "commands": [
                "(Get-WmiObject -Class Win32_ComputerSystem).PartOfDomain"
            ]
        }
    }
}
```

Parâmetros

Os parâmetros necessários e opcionais especificados no documento.

Tipo: mapa

Obrigatório: Não

CloudWatchOutputConfig

Opções de configuração para enviar a saída do comando para o Amazon CloudWatch Logs. Para obter mais informações sobre o envio de saída de comando para o CloudWatch Logs, consulte Configurar o Amazon CloudWatch Logs para Run Command.

Tipo: StringMap (O esquema do mapa deve corresponder ao objeto. Para obter mais informações, consulte CloudWatchOutputConfig na Referência de API do AWS Systems Manager).

Obrigatório: Não

Veja um exemplo a seguir.

YAML

```
- name: checkMembership
  action: aws:runCommand
  inputs:
    DocumentName: AWS-RunPowerShellScript
    InstanceIds:
      - "{{InstanceIds}}"
    Parameters:
      commands:
        - "(Get-WmiObject -Class Win32_ComputerSystem).PartOfDomain"
    CloudWatchOutputConfig:
      CloudWatchLogGroupName: CloudWatchGroupForSSMAutomationService
```

```
      CloudWatchOutputEnabled: true
```

JSON

```
{
    "name": "checkMembership",
    "action": "aws:runCommand",
    "inputs": {
        "DocumentName": "AWS-RunPowerShellScript",
        "InstanceIds": [
            "{{InstanceIds}}"
        ],
        "Parameters": {
            "commands": [
                "(Get-WmiObject -Class Win32_ComputerSystem).PartOfDomain"
            ]
        },
        "CloudWatchOutputConfig" : {
                "CloudWatchLogGroupName":
 "CloudWatchGroupForSSMAutomationService",
                "CloudWatchOutputEnabled": true
        }
    }
}
```

Comentário

Informações definidas pelo usuário sobre o comando.

Tipo: String

Obrigatório: Não

DocumentHash

O hash do documento.

Tipo: String

Obrigatório: Não

DocumentHashType

O tipo de hash.

Tipo: String

Valores válidos: `Sha256` | `Sha1`

Obrigatório: Não

NotificationConfig

As configurações para enviar notificações.

Obrigatório: Não

OutputS3BucketName

O nome do bucket do S3 para as respostas de saída de comandos. Seu nó gerenciado deve ter permissões para o bucket do S3 para registrar a saída com êxito.

Tipo: String

Obrigatório: Não

OutputS3KeyPrefix

O prefixo .

Tipo: String

Obrigatório: Não

ServiceRoleArn

O ARN da função do AWS Identity and Access Management (IAM).

Tipo: String

Obrigatório: Não

TimeoutSeconds

O tempo, em segundos, para esperar que um comando seja entregue ao AWS Systems Manager SSM Agent em uma instância. Se o comando não for recebido pelo SSM Agent na instância antes que o valor especificado seja atingido, o status do comando mudará para `Delivery Timed Out`.

Tipo: inteiro

Obrigatório: Não

Valores válidos: 30 a 2592000

Saída

CommandId

O ID do comando.

Status

O status do comando.

ResponseCode

O código de resposta do comando. Se o documento executado tiver mais de uma etapa, nenhum valor será retornado para essa saída.

Saída

A saída do comando. Se você segmentar uma tag ou várias instâncias com seu comando, nenhum valor de saída será retornado. Você pode usar as operações `GetCommandInvocation` e `ListCommandInvocations` da API para recuperar a saída de instâncias individuais.

`aws:runInstances` – Executa uma instância do Amazon EC2

Inicia uma nova instância do Amazon Elastic Compute Cloud (Amazon EC2).

Entrada

A ação oferece suporte para a maioria dos parâmetros de API. Para obter mais informações, consulte a documentação da API RunInstances.

YAML

```
name: launchInstance
action: aws:runInstances
maxAttempts: 3
timeoutSeconds: 1200
onFailure: Abort
inputs:
  ImageId: ami-12345678
```

```
  InstanceType: t2.micro
  MinInstanceCount: 1
  MaxInstanceCount: 1
  IamInstanceProfileName: myRunCmdRole
  TagSpecifications:
  - ResourceType: instance
    Tags:
    - Key: LaunchedBy
      Value: SSMAutomation
    - Key: Category
      Value: HighAvailabilityFleetHost
```

JSON

```
{
   "name":"launchInstance",
   "action":"aws:runInstances",
   "maxAttempts":3,
   "timeoutSeconds":1200,
   "onFailure":"Abort",
   "inputs":{
      "ImageId":"ami-12345678",
      "InstanceType":"t2.micro",
      "MinInstanceCount":1,
      "MaxInstanceCount":1,
      "IamInstanceProfileName":"myRunCmdRole",
      "TagSpecifications":[
         {
            "ResourceType":"instance",
            "Tags":[
               {
                  "Key":"LaunchedBy",
                  "Value":"SSMAutomation"
               },
               {
                  "Key":"Category",
                  "Value":"HighAvailabilityFleetHost"
               }
            ]
         }
      ]
   }
}
```

AdditionalInfo

Reservado.

Tipo: string

Obrigatório: Não

BlockDeviceMappings

Os dispositivos de bloco para a instância.

Tipo: MapList

Obrigatório: Não

ClientToken

O identificador para garantir a idempotência da solicitação.

Tipo: string

Obrigatório: Não

DisableApiTermination

Ativa ou desativa o encerramento da API da instância.

Tipo: booliano

Obrigatório: Não

EbsOptimized

Ativa ou desativa a otimização do Amazon Elastic Block Store (Amazon EBS).

Tipo: booliano

Obrigatório: Não

IamInstanceProfileArn

O nome do recurso da Amazon (ARN) do perfil da instância do AWS Identity and Access Management (IAM) a ser associado às instâncias.

Tipo: string

Obrigatório: Não

IamInstanceProfileName

O nome do perfil de instância do IAM para a instância.

Tipo: string

Obrigatório: Não

ImageId

O ID da Amazon Machine Image (AMI).

Tipo: string

Obrigatório: Sim

InstanceInitiatedShutdownBehavior

Indica se a instância é interrompida ou encerrada no desligamento do sistema.

Tipo: string

Obrigatório: Não

InstanceType

O tipo de instância.

> **Note**
>
> Se o valor de um tipo de instância não for fornecido, o tipo de instância m1.small será usado.

Tipo: string

Obrigatório: Não

KernelId

O ID do kernel.

Tipo: string

Obrigatório: Não

KeyName

O nome do par de chaves.

Tipo: string

Obrigatório: Não

MaxInstanceCount

O número máximo de instâncias a serem executadas.

Tipo: string

Obrigatório: Não

MetadataOptions

As opções de metadados da instância. Para obter mais informações, consulte InstanceMetadataOptionsRequest.

Tipo: StringMap

Obrigatório: Não

MinInstanceCount

O número mínimo de instâncias a serem executadas.

Tipo: string

Obrigatório: Não

Monitorar

Ativa ou desativa o monitoramento detalhado.

Tipo: booliano

Obrigatório: Não

NetworkInterfaces

As interfaces de rede.

Tipo: MapList

Obrigatório: Não

Posicionamento

O posicionamento da instância.

Tipo: StringMap

Obrigatório: Não

PrivateIpAddress

O endereço IPv4 primário.

Tipo: string

Obrigatório: Não

RamdiskId

O ID do disco RAM.

Tipo: string

Obrigatório: Não

SecurityGroupIds

Os IDs dos security groups da instância.

Tipo: StringList

Obrigatório: Não

SecurityGroups

Os nomes dos security groups da instância.

Tipo: StringList

Obrigatório: Não

SubnetId

O ID da sub-rede.

Tipo: string

Obrigatório: Não

TagSpecifications

As tags a serem aplicadas aos recursos durante a execução. Você só pode marcar instâncias e volumes na inicialização. As tags especificadas são aplicadas a todas as instâncias ou volumes que são criados durante a execução. Para marcar uma instância depois que ela tiver sido iniciada, use a ação `aws:createTags`: cria tags para os recursos da AWS.

Tipo: MapList (Para obter mais informações, consulte TagSpecification.)

Obrigatório: Não

UserData

Um script fornecido como um valor literal de string. Se um valor literal for inserido, ele deverá ser codificado em Base64.

Tipo: string

Obrigatório: Não

Saída

InstanceIds

Os IDs das instâncias.

InstanceStates

O estado atual da instância.

`aws:sleep`: atrasa uma automação

Atrasa a execução da Automação por um período de tempo especificado. Essa ação usa o formato de data e hora da Organização Internacional de Normalização (ISO) 8601. Para obter mais informações sobre esse formato de data e hora, consulte ISO 8601.

Entrada

É possível atrasar uma automação para uma duração especificada.

YAML

```
name: sleep
action: aws:sleep
inputs:
  Duration: PT10M
```

JSON

```
{
   "name":"sleep",
   "action":"aws:sleep",
   "inputs":{
      "Duration":"PT10M"
   }
}
```

Você também pode atrasar a automação até uma data e hora especificadas. Se a data e a hora especificadas tiverem passado, a ação prosseguirá imediatamente.

YAML

```
name: sleep
action: aws:sleep
inputs:
  Timestamp: '2020-01-01T01:00:00Z'
```

JSON

```
{
    "name": "sleep",
    "action": "aws:sleep",
    "inputs": {
        "Timestamp": "2020-01-01T01:00:00Z"
    }
}
```

> **Note**
>
> A automação permite um atraso máximo de 604.799 segundos (7 dias).

Duração

Uma duração ISO 8601. Não é possível especificar uma duração negativa.

Tipo: string

Obrigatório: Não

Timestamp

Um timestamp ISO 8601. Se você não especificar um valor para esse parâmetro, deverá especificar um valor para o parâmetro `Duration`.

Tipo: string

Obrigatório: Não

Saída

Nenhum

`aws:updateVariable`: atualiza um valor para uma variável do runbook

Essa ação atualiza um valor para uma variável do runbook. O tipo de dados do valor deve corresponder ao tipo de dados da variável que você deseja atualizar. Não há suporte a conversões de tipo de dados. A propriedade `onCancel` não é compatível com a ação `aws:updateVariable`.

Entrada

Veja a entrada a seguir.

YAML

```
name: updateStringList
action: aws:updateVariable
```

```
inputs:
    Name: variable:variable name
    Value:
    - "1"
    - "2"
```

JSON

```
{
    "name": "updateStringList",
    "action": "aws:updateVariable",
    "inputs": {
        "Name": "variable:variable name",
        "Value": ["1","2"]
    }
}
```

Nome

O nome da variável cujo valor você deseja atualizar. É necessário usar o formato `variable:variable name`

Tipo: string

Obrigatório: Sim

Valor

O novo valor a ser atribuído à variável. O valor deve corresponder ao tipo de dados da variável. Não há suporte a conversões de tipo de dados.

Tipo: Boolean | Integer | MapList | String | StringList | StringMap

Obrigatório: Sim

Restrições:

- O MapList pode conter um número máximo de 200 itens.
- Os comprimentos de chave podem ter um comprimento mínimo de 1 e um máximo de 50.
- StringList pode ter um número mínimo de 0 itens e um número máximo de 50 itens.
- Os comprimentos de string podem ter um comprimento mínimo de 1 e um máximo de 512.

Saída

Nenhum

`aws:waitForAwsResourceProperty`: aguarde uma propriedade de recurso da AWS

A ação `aws:waitForAwsResourceProperty` permite que a automação espere por um estado específico do recurso ou um estado do evento antes de continuar a automação. Para obter mais exemplos de como usar essa ação, consulte Exemplos adicionais de runbook.

> **Note**
>
> O valor de tempo limite padrão para esta ação é 3600 segundos (uma hora). Você pode limitar ou prolongar o tempo limite especificando o parâmetro `timeoutSeconds` para uma etapa `aws:waitForAwsResourceProperty`. Para obter mais informações e exemplos de como usar essa ação, consulte Gerenciar tempos limite em runbooks.

Entrada

As entradas são definidas pela operação de API que você escolher.

YAML

```
action: aws:waitForAwsResourceProperty
inputs:
  Service: The official namespace of the service
  Api: The API operation or method name
  API operation inputs or parameters: A value
  PropertySelector: Response object
  DesiredValues:
  - Desired property value
```

JSON

```
{
  "action": "aws:waitForAwsResourceProperty",
  "inputs": {
    "Service":"The official namespace of the service",
    "Api":"The API operation or method name",
```

```
    "API operation inputs or parameters":"A value",
    "PropertySelector": "Response object",
    "DesiredValues": [
      "Desired property value"
    ]
  }
}
```

Serviço

O namespace do AWS service (Serviço da AWS) que contém a operação de API que você deseja executar. Por exemplo, o namespace para AWS Systems Manager é `ssm`. O namespace do Amazon Elastic Compute Cloud (Amazon EC2) é `ec2`. Você pode visualizar uma lista de namespaces de AWS service (Serviço da AWS) compatíveis na seção Available Services (Serviços disponíveis) da Referência de comandos da AWS CLI.

Tipo: string

Obrigatório: Sim

API

O nome da operação de API que você deseja executar. Você pode visualizar as operações de API (também chamadas de métodos), escolhendo um serviço na navegação à esquerda na seguinte página de Referência de serviços: Escolha um método na seção Client (Cliente) para o serviço que você deseja invocar. Por exemplo, todas as operações de API (métodos) do Amazon Relational Database Service (Amazon RDS) estão listadas na seguinte página: Amazon RDS methods (Métodos do Amazon RDS).

Tipo: string

Obrigatório: Sim

Entradas de operação da API

Uma ou mais entradas de operação da API. Você pode visualizar as entradas disponíveis (também chamadas de parâmetros), escolhendo um serviço na navegação à esquerda na seguinte página de Referência de serviços. Escolha um método na seção Client (Cliente) para o serviço que você deseja invocar. Por exemplo, todos os métodos de API estão listados na página a seguir: Métodos do Amazon RDS. Escolha o método describe_db_instances e role para baixo para ver os parâmetros disponíveis, como DBInstanceIdentifier, Name (Nome) e Values (Valores).

YAML

```
inputs:
  Service: The official namespace of the service
  Api: The API operation name
  API input 1: A value
  API Input 2: A value
  API Input 3: A value
```

JSON

```
"inputs":{
      "Service":"The official namespace of the service",
      "Api":"The API operation name",
      "API input 1":"A value",
      "API Input 2":"A value",
      "API Input 3":"A value"
}
```

Tipo: determinado pela ação de API escolhida

Obrigatório: Sim

PropertySelector

O JSONPath para um determinado atributo no objeto de resposta. Você pode visualizar os objetos de resposta escolhendo um serviço na navegação à esquerda na seguinte página de Referência de serviços. Escolha um método na seção Client (Cliente) para o serviço que você deseja invocar. Por exemplo, todos os métodos de API estão listados na página a seguir: Métodos do Amazon RDS. Escolha o método describe_db_instances e role para baixo até a seção Response Structure (Estrutura de resposta). DBInstances é listado como um objeto de resposta.

Tipo: string

Obrigatório: Sim

DesiredValues

O status ou estado esperado no qual a automação deve continuar.

Tipo: MapList, StringList

Obrigatório: Sim

Variáveis de sistema de automação

Os runbooks do AWS Systems Manager Automation usam as variáveis a seguir. Para obter um exemplo de como essas variáveis são usadas, visualize a origem JSON do runbook `AWS-UpdateWindowsAmi`.

Para visualizar a origem JSON do runbook **`AWS-UpdateWindowsAmi`**

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Documents.
3. Na lista de documentos, use a barra de pesquisa ou os números à direita da barra de pesquisa para escolher o runbook **`AWS-UpdateWindowsAmi`**.
4. Escolha a guia Conteúdo.

Variáveis do sistema

Runbooks do Automation oferecem suporte para as seguintes variáveis de sistema:

Variável	Detalhes
`global:ACCOUNT_ID`	O ID da Conta da AWS do usuário ou do perfil no qual o Automation é executado.
`global:DATE`	A data (no runtime), no formato aaaa-MM-dd.
`global:DATE_TIME`	A data e a hora (no runtime), no formato aaaa-MM-dd_HH.mm.ss.
`global:AWS_PARTITION`	A partição na qual o recurso está. Para Regiões da AWS padrão, a partição é `aws`. Para obter recursos em outras partições, a partição é `aws-`*`partitionname`* . Por exemplo, a partição para recursos na região da AWS GovCloud (Oeste dos EUA) é `aws-us-gov` .

Variável	Detalhes
`global:REGION`	A região em que o runbook é executado. Por exemplo, us-east-2.

Variáveis de automação

Runbooks do Automation oferecem suporte para as seguintes variáveis de automação:

Variável	Detalhes
`automation:EXECUTION_ID`	O identificador exclusivo atribuído à automação atual. Por exemplo, `1a2b3c-1a2b3c-1a2b3c-1a2b3c1a2b3c1a2b3c` .

Tópicos

- Terminologia
- Cenários compatíveis
- Cenários não compatíveis

Terminologia

Os termos a seguir descrevem como as variáveis e os parâmetros são resolvidos.

Prazo	Definição	Exemplo
ARN constante	Um nome do recurso da Amazon (ARN) válido sem variáveis.	`arn:aws:iam::123456789012:role/roleName`
Parâmetro do runbook	Um parâmetro definido no nível do runbook (por exemplo, `instanceId`). O parâmetro é usado em uma substituição de string básica.	`{ "description": "Create Image Demo", "version": "0.3",`

Prazo	Definição	Exemplo
	Seu valor é fornecido na ocasião de início da execução.	<pre>"assumeRole": "*Your_Automation_Assume_Role_ARN* ", "parameters":{ **"instanceId": { "type": "String", "description": "Instance to create image from"** } }</pre>

Prazo	Definição	Exemplo
Variável de sistema	Uma variável geral substituída no runbook quando uma parte dele é avaliada.	`"activities": [{ "id": "copyImage", "activityType": "AWS-CopyImage", "maxAttempts": 1, "onFailure": "Continue", "inputs": { "ImageName": "{{imageName}}", "SourceImageId": "{{sourceImageId}}", "SourceRegion": "{{sourceRegion}}", "Encrypted": true, "ImageDescription": "Test CopyImage Description created on **{{global:DATE}}** " } }]`

Prazo	Definição	Exemplo
Variável de automação	Uma variável relativa à execução de automação substituída no runbook quando uma parte dele for avaliada.	`{ "name": "runFixedCmds", "action": "aws:runCommand", "maxAttempts": 1, "onFailure": "Continue", "inputs": { "DocumentName": "AWS-RunPowerShellScript", "InstanceIds": ["{{LaunchInstance.InstanceIds}}"], "Parameters": { "commands": ["dir", "date", "“{{outputFormat}}” -f “left”,”right”,”{{global:DATE}}”,” **{{automation:EXECUTION_ID}}** ”] } } }`

Prazo	Definição	Exemplo
Parâmetro do Systems Manager	Uma variável definida no AWS Systems Manager Parameter Store. Ele não pode ser referenciado diretamente na entrada da etapa. Podem ser necessárias permissões para acessar o parâmetro.	`description: Launch new Windows test instance schemaVersion: '0.3' assumeRole: '{{AutomationAssumeRole}}' parameters: AutomationAssumeRole: type: String default: '' description: >- (Required) The ARN of the role that allows Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses your IAM permissions to run this runbook. LatestAmi: type: String default: >- {{ssm:/aws/service/ami-windows-latest/Windows_Server-2016-English-Full-Base}} description: The latest Windows Server 2016 AMI queried from the public parameter. mainSteps: - name: launchInstance action: 'aws:runInstances' maxAttempts: 3`

Prazo	Definição	Exemplo
		timeoutSeconds: 1200 onFailure: Abort inputs: ImageId: '{{LatestAmi}}' ...

Cenários compatíveis

Cenário	Comentários	Exemplo
`assumeRole` de ARN constante na criação.	Uma verificação de autorização é realizada para verificar se o usuário de chamada tem permissão para transmitir a `assumeRole` especificada.	{ "description": "Test all Automation resolvable parameters", "schemaVersion": "0.3", "assumeRole": ***"arn:aws:iam::123456789012:role/roleName"*** , "parameters": { ...
Parâmetro do runbook fornecido para o `AssumeRole` quando a automação é iniciada.	Devem ser definidos na lista de parâmetros do runbook.	{ "description": "Test all Automation resolvable parameters", "schemaVersion": "0.3", "assumeRole": **"{{dynamicARN}}"** , "parameters": { ...

Cenário	Comentários	Exemplo

Valor fornecido para o parâmetro do runbook no início.	O cliente fornece o valor a ser usado para um parâmetro. Quaisquer entradas fornecidas na inicialização precisam ser definidas na lista de parâmetros do runbook.	`...`
		`"parameters": {`
		`"amiId": {`
		`"type": "String",`
		`"default": "ami-12345678 ",`
		`"description": "list of commands to run as part of first step"`
		`},`
		`...`
		
		As entradas para iniciar a execução da automação incluem: `{"amiId" : ["ami-12345678 "] }`

Cenário	Comentários	Exemplo

Parâmetro do Systems Manager referenciado no conteúdo do runbook.	A variável existe na conta do cliente ou é um parâmetro acessível publicamente, e o `AssumeRole` do runbook tem acesso à variável. Uma verificação é realizada na ocasião da criação para confirmar se `AssumeRole` tem acesso. O parâmetro não pode ser referenciado diretamente na entrada da etapa.	```
		...
		parameters:
		    LatestAmi:
		    type: String
		    default: >-
		      {{ssm:/aws/
		service/ami-wind
		ows-latest/Windows
		_Server-2016-English-
		Full-Base}}
		    description: The
		 latest Windows Server
		 2016 AMI queried from
		 the public parameter.
		mainSteps:
		  - name: launchIns
		tance
		    action: 'aws:runI
		nstances'
		    maxAttempts: 3
		    timeoutSeconds:
		 1200
		    onFailure: Abort
		    inputs:
		      ImageId: '{{Latest
		Ami}}'
		...
		``

Cenário	Comentários	Exemplo

Variável de sistema referenciada na definição de etapa	Uma variável de sistema é substituída no runbook quando a automação é iniciada. O valor injetado no runbook é relativo a quando ocorre a substituição. Ou seja, o valor de uma variável de tempo injetado na etapa 1 é diferente do valor injetado na etapa 3 devido ao tempo necessário para executar as etapas intermediárias. As variáveis do sistema não precisam ser definidas na lista de parâmetros do runbook.	`...`
		`  "mainSteps": [`
		`    {`
		`      "name": "RunSomeCommands",`
		`      "action": "aws:runCommand",`
		`      "maxAttempts": 1,`
		`      "onFailure": "Continue",`
		`      "inputs": {`
		`        "DocumentName": "AWS:RunPowerShell",`
		`        "InstanceIds": ["{{LaunchInstance.InstanceIds}}"],`
		`        "Parameters": {`
		`            "commands" : [`
		`                "echo {The time is now {{global:DATE_TIME}}}"`
		`            ]`
		`        }`
		`    }`
		`}, ...`

Cenário	Comentários	Exemplo
Variável de automação referenciada na definição de etapa.	As variáveis de automação não precisam ser definidas na lista de parâmetros do runbook. A única variável de Automação com suporte é automation:EXECUTION_ID.	... "mainSteps": [{ "name": "invokeLambdaFunction", "action": "aws:invokeLambdaFunction", "maxAttempts": 1, "onFailure": "Continue", "inputs": { "FunctionName": "Hello-World-LambdaFunction", "Payload" : "{ "executionId" : "{{automation:EXECUTION_ID}}" }" } } ...

Cenário	Comentários	Exemplo

Consulte a saída da etapa anterior na próxima definição de etapa.	Este é o redirecionamento de parâmetros. A saída de uma etapa anterior é referenciada usando a sintaxe `{{stepName.OutputName}}`. Essa sintaxe não pode ser usada pelo cliente para parâmetros do runbook. Isso é resolvido quando a etapa de referência é executada. O parâmetro não está listado nos parâmetros do runbook.	`...`
		`"mainSteps": [`
		`{`
		`"name": "LaunchInstance",`
		`"action": "aws:runInstances",`
		`"maxAttempts": 1,`
		`"onFailure": "Continue",`
		`"inputs": {`
		`"ImageId": "{{amiId}}",`
		`"MinInstanceCount": 1,`
		`"MaxInstanceCount": 2`
		`}`
		`},`
		`{`
		`"name":"changeState",`
		`"action": "aws:changeInstanceState",`
		`"maxAttempts": 1,`
		`"onFailure": "Continue",`
		`"inputs": {`
		`"InstanceIds": ["{{LaunchInstance.InstanceIds}}"],`
		`"DesiredState": "terminated"`
		`}`
		`}`
		
		`...`

Cenários não compatíveis

Cenário	Comentário	Exemplo

Parâmetro do Systems Manager fornecido para o `assumeRole` na criação	Sem suporte.	`...`
		
		`{`
		`  "description":`
		` "Test all Automation`
		` resolvable parameter`
		`s",`
		`  "schemaVersion":`
		` "0.3",`
		`  "assumeRole":`
		` "{{ssm:administrato`
		`rRoleARN}}  ",`
		`  "parameters": {`
		
		`...`

Parâmetro do Systems Manager referenciado diretamente na entrada da etapa.	Retorna a exceção `InvalidDocumentCon tent` no momento da criação.	`...`
		`mainSteps:`
		`  - name: launchIns`
		`tance`
		`    action: 'aws:runI`
		`nstances'`
		`    maxAttempts: 3`
		`    timeoutSeconds:`
		` 1200`
		`    onFailure: Abort`
		`    inputs:`
		`      ImageId: '{{ssm:/`
		`aws/service/ami-win`
		`dows-latest/Window`
		`s_Server-2016-Engl`
		`ish-Full-Base}}'`
		`...`

Cenário	Comentário	Exemplo
Definição de etapa variável	A definição de uma etapa no runbook é construída por variáveis.	... "mainSteps": [{ "name": "LaunchInstance", "action": "aws:runInstances", "**{{attemptModel}}** ": 1, "onFailure": "Continue", "inputs": { "ImageId": "*ami-12345678* ", "MinInstanceCount": 1, "MaxInstanceCount": 2 } ... User supplies input : { "attemptModel" : "**minAttempts** " }

Cenário	Comentário	Exemplo
Referência cruzada de parâmetros do runbook	O usuário fornece um parâmetro de entrada na hora de início, que é uma referência a outro parâmetro no runbook.	... "parameters": { "amiId": { "type": "String", "default": "ami-7f2e6015 ", "description": "list of commands to run as part of first step" }, "alternateAmiId": { "type": "String", "description": "The alternate AMI to try if this first fails". "default" : "{{amiId}}" }, ...

Cenário	Comentário	Exemplo
Expansão de vários níveis	O runbook define uma variável que é avaliada como o nome de uma variável. Ela fica dentro dos delimitadores de variáveis (ou seja, {{ }}) e é expandida para o valor dessa variável/parâmetro.	... "parameters": { "*firstParameter* ": { "type": "String", "default": "param2", "description": "The parameter to reference" }, "*secondParameter* ": { "type": "String", "default" : "echo {Hello world}", "description": "What to run" } }, "mainSteps": [{ "name": "runFixedCmds", "action": "aws:runCommand", "maxAttempts": 1, "onFailure": "Continue", "inputs": { "DocumentName": "AWS-RunPowerShellScript", "InstanceIds" : "{{LaunchInstance.InstanceIds}}", "Parameters": { "commands": ["{{ ***{{firstParameter}}*** }}"] }

Cenário	Comentário	Exemplo
		``` ... Note: The customer intention here would be to run a command of "echo {Hello world}" ```

Cenário	Comentário	Exemplo

Referenciar a saída de uma etapa do runbook que é um tipo de variável diferente	O usuário faz referência à saída de uma etapa do runbook anterior em uma etapa subsequente. A saída é um tipo de variável que não atende aos requisitos da ação na etapa subsequente.	```
		...
		mainSteps:
		- name: getImageId
		  action: aws:execu
		teAwsApi
		  inputs:
		    Service: ec2
		    Api: DescribeImages
		    Filters:
		    - Name: "name"
		      Values:
		      - "{{ImageName}}"
		  outputs:
		  - Name: ImageIdList
		    Selector: "$.Images
		"
		    Type: "StringList"
		- name: copyMyImages
		  action: aws:copyImage
		  maxAttempts: 3
		  onFailure: Abort
		  inputs:
		    SourceImageId:
		 {{getImageId.Image
		IdList}}
		    SourceRegion: ap-
		northeast-2
		    ImageName:
		 Encrypted Copies of
		 LAMP base AMI in ap-
		northeast-2
		    Encrypted: true
		...
		Note: You must provide
		 the type required
		 by the Automation
		 action.
		In this case, aws:copyI
		mage requires a
		 "String" type variable
		 but the preceding step
		``

Cenário	Comentário	Exemplo
		`outputs a "StringList" type variable.`

Criação dos seus próprios runbooks

Um runbook do Automation define as ações que o Systems Manager realizará nas instâncias gerenciadas e outros recursos da AWS quando uma automação for executada. O Automation é uma ferramenta do AWS Systems Manager. Um runbook contém uma ou mais etapas que são executadas em ordem sequencial. Cada etapa se baseia em uma única ação. A saída de uma etapa pode ser usada como entrada de uma etapa posterior.

O processo de execução dessas ações e suas etapas é chamado de automação.

Os tipos de ação compatíveis com runbooks permitem automatizar uma ampla variedade de operações em seu ambiente da AWS. Por exemplo, usando o tipo de ação `executeScript`, você pode incorporar um script do Python ou PowerShell diretamente em seu runbook. (Ao criar um runbook personalizado, você pode adicionar seu script em linha ou anexá-lo de um bucket do S3 ou de sua máquina local.) Você pode automatizar o gerenciamento de seus recursos do AWS CloudFormation usando os tipos de ação `createStack` e `deleteStack`. Além disso, usando o tipo de ação `executeAwsApi`, uma etapa pode executar qualquer operação de API em qualquer AWS service (Serviço da AWS), incluindo criar ou excluir recursos da AWS, iniciar outros processos, acionar notificações e muito mais.

Para obter uma lista de todos os 20 tipos de ação compatíveis com o Automation, consulte Referência de ações do Systems Manager Automation.

O AWS Systems Manager Automation inclui vários runbooks com etapas predefinidas que você pode usar para executar tarefas comuns, como reiniciar uma ou mais instâncias do Amazon Elastic Compute Cloud (Amazon EC2) ou criar uma Amazon Machine Image (AMI). Você também pode criar seus próprios runbooks de automação e compartilhá-los com outras contas da Contas da AWS ou torná-los públicos para todos os usuários do Automation.

Os runbooks são escritos em YAML ou JSON. No entanto, usando o Document Builder no console do Systems Manager Automation, você pode criar um runbook sem precisar criar em JSON ou YAML nativo.

> **Important**
>
> Se você executar um fluxo de trabalho de automação que chama outros serviços usando uma função de serviço do AWS Identity and Access Management (IAM), esteja ciente de que esta função deve ser configurada com permissão para chamar esses serviços. Esse requisito aplica-se a todos os runbooks do Automation da AWS (runbooks da `AWS-*`), como os runbooks `AWS-ConfigureS3BucketLogging`, `AWS-CreateDynamoDBBackup` e `AWS-RestartEC2Instance`, entre outros. Esse requisito também se aplica a todos os runbooks personalizados do Automation criados que invoquem outros Serviços da AWS, usando ações que chamam outros serviços. Por exemplo, se você usar as ações `aws:executeAwsApi`, `aws:createStack` ou `aws:copyImage`, configure a função de serviço com permissão para invocar esses serviços. É possível habilitar permissões para outros Serviços da AWS, adicionando uma política em linha do IAM à função. Para obter mais informações, consulte (Opcional) Adicione uma política em linha ou uma política gerenciada pelo cliente para invocar outros Serviços da AWS.

Para obter informações sobre as ações que podem ser especificadas em um runbook, consulte Referência de ações do Systems Manager Automation.

Para obter informações sobre como usar o AWS Toolkit for Visual Studio Code para criar runbooks, consulteTrabalhar com documentos do Systems Manager Automation no Manual do usuário do AWS Toolkit for Visual Studio Code.

Para obter informações sobre como usar o designer visual para criar um runbook personalizado, consulte Experiência de design visual para runbooks de automação.

Sumário

- Experiência de design visual para runbooks de automação
  - Antes de começar
  - Visão geral da interface de experiência de design visual
    - Navegador Ações
    - Canvas
    - Formulário
    - Atalhos de teclado
  - Usar a experiência de design visual

        - Criar um runbook
        - Desenvolver um runbook
        - Atualizar seu runbook
        - Exportar seu runbook
    - Configurar entradas e saídas para suas ações
        - Forneça dados de entrada para uma ação
        - Definir dados de saída para uma ação
    - Tratamento de erros com a experiência de design visual
        - Repetir a ação em caso de erro
        - Tempo limite
        - Ações com falha
        - Ações canceladas
        - Ações críticas
        - Enceramento de ações
    - Tutorial: criar um runbook usando a experiência de design visual
        - Etapa 1: navegue até a experiência de design visual
        - Etapa 2: criar um fluxo de trabalho
        - Etapa 3: analisar o código gerado automaticamente
        - Etapa 4: executar seu novo runbook
        - Etapa 5: limpar
- Criar runbooks do Automation
    - Identifique seu caso de uso
    - Configuração do ambiente de desenvolvimento
    - Desenvolva conteúdo do runbook
    - Exemplo 1: criação de runbooks pai-filho
        - Crie o runbook filho
        - Crie o runbook pai
    - Exemplo 2: Runbook com script
    - Exemplos adicionais de runbook
        - Implantar a arquitetura da VPC e os controles de domínio do Microsoft Active Directory

        - Restaurar um volume raiz do snapshot mais recente
        - Crie uma AMI e uma cópia entre regiões
- Criar parâmetros de entrada que preenchem os recursos da AWS
- Uso do Document Builder para criar runbooks
    - Criação de um runbook usando o Document Builder
    - Crie um runbook que execute scripts
- Uso de scripts em runbooks
    - Permissões para usar runbooks
    - Adicionar scripts a runbooks
    - Restrições de script para runbooks
- Uso de instruções condicionais em runbooks
    - Trabalhar com a ação aws:branch
        - Criar uma etapa aws:branch em um runbook
            - Sobre a criação de variáveis de saída
        - Runbooks aws:branch de exemplo
        - Criar automações de ramificação complexas com operadores
    - Exemplos de como usar opções condicionais
- Uso de saídas de ações como entradas
    - Uso de JSONPath em runbooks
- Criação de integrações de webhooks para o Automation
    - Criação de integrações (console)
    - Criação de integrações (linha de comando)
    - Criação de webhooks para integrações
- Gerenciar tempos limite em runbooks

Experiência de design visual para runbooks de automação

O AWS Systems Manager Automation fornece uma experiência de design visual de baixo código que ajuda você a criar runbooks de automação. A experiência de design visual fornece uma interface simples de arrastar e soltar com a opção de adição do seu próprio código para que você possa criar e editar runbooks mais facilmente. Com a experiência de design visual, você pode fazer o seguinte:

- Controlar instruções condicionais.
- Controlar como a entrada e a saída são filtradas ou transformadas para cada ação.
- Configurar o tratamento de erros.
- Criar protótipos de novos runbooks.
- Usar seus protótipos de runbooks como ponto de partida para o desenvolvimento local com o AWS Toolkit for Visual Studio Code.

Ao criar ou editar um runbook, você pode acessar a experiência de design visual no console do Automation. À medida que você cria um runbook, a experiência de design visual valida seu trabalho e gera código automaticamente. Você pode revisar o código gerado ou exportá-lo para desenvolvimento local. Ao terminar, você pode salvar seu runbook, executá-lo e examinar os resultados no console do Systems Manager Automation.

Antes de começar

Para usar a experiência de design visual, você precisa de uma Conta da AWS e credenciais que forneçam as permissões corretas para qualquer recurso que você queira usar.

Na experiência de design visual, a automação se integra ao Amazon CodeGuru Security para ajudar a detectar violações de políticas de segurança e vulnerabilidades em seus scripts Python. Para usar esse recurso para ações `aws:executeScript`, sua política do AWS Identity and Access Management (IAM) deve incluir as seguintes permissões:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "codeguru-security:CreateUploadUrl",
                "codeguru-security:CreateScan",
                "codeguru-security:GetScan",
                "codeguru-security:GetFindings"
            ],
            "Resource": "*"
        }
```

```
    ]
}
```

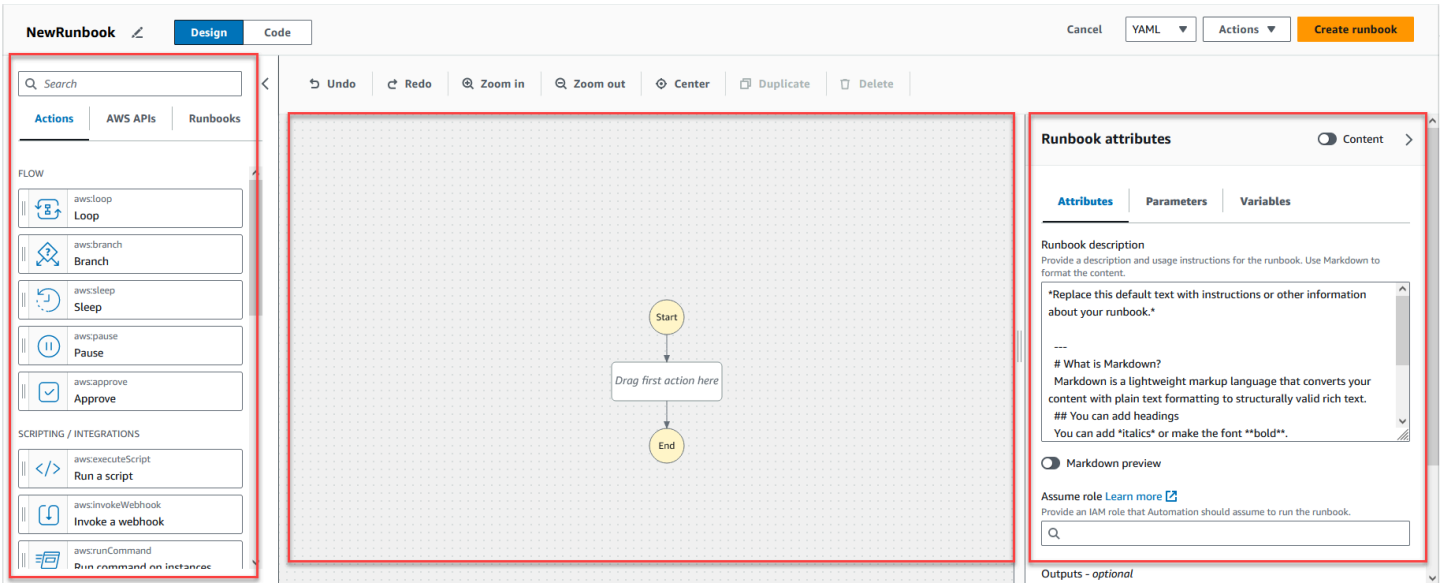
Tópicos

- Visão geral da interface de experiência de design visual
- Usar a experiência de design visual
- Configurar entradas e saídas para suas ações
- Tratamento de erros com a experiência de design visual
- Tutorial: criar um runbook usando a experiência de design visual

Visão geral da interface de experiência de design visual

A experiência de design visual do Systems Manager Automation é um designer de fluxo de trabalho visual de baixo código que ajuda você a criar runbooks de automação.

Conheça a experiência de design visual com uma visão geral dos componentes da interface:



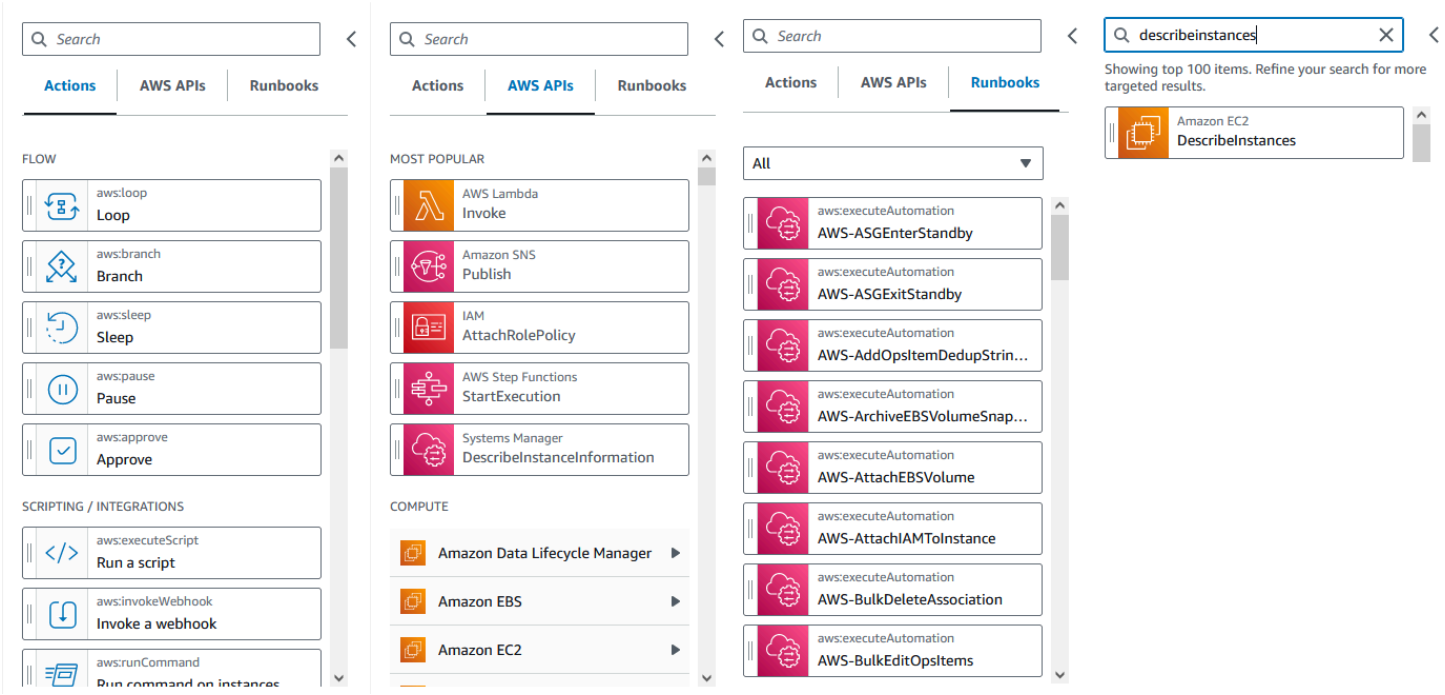
- O navegador Ações contém as guias Ações, APIs da AWS e Runbooks.
- A tela é onde você arrasta e solta ações no gráfico do fluxo de trabalho, altera a ordem das ações e seleciona ações para configurar ou visualizar.
- O painel Formulário é onde você pode visualizar e editar as propriedades de qualquer ação selecionada na tela. Selecione a opção Conteúdo para visualizar o YAML ou o JSON do seu runbook, com a ação atualmente selecionada destacada.

Os links de Informações abrem um painel com informações contextuais quando você precisa de ajuda. Esses painéis também incluem links para tópicos relacionados na documentação do Systems Manager Automation.

Navegador Ações

No navegador Ações, é possível selecionar ações para arrastar e soltar no gráfico do fluxo de trabalho. Você pode pesquisar todas as ações usando o campo de pesquisa na parte superior do navegador Ações. O navegador Ações contém as seguintes guias:

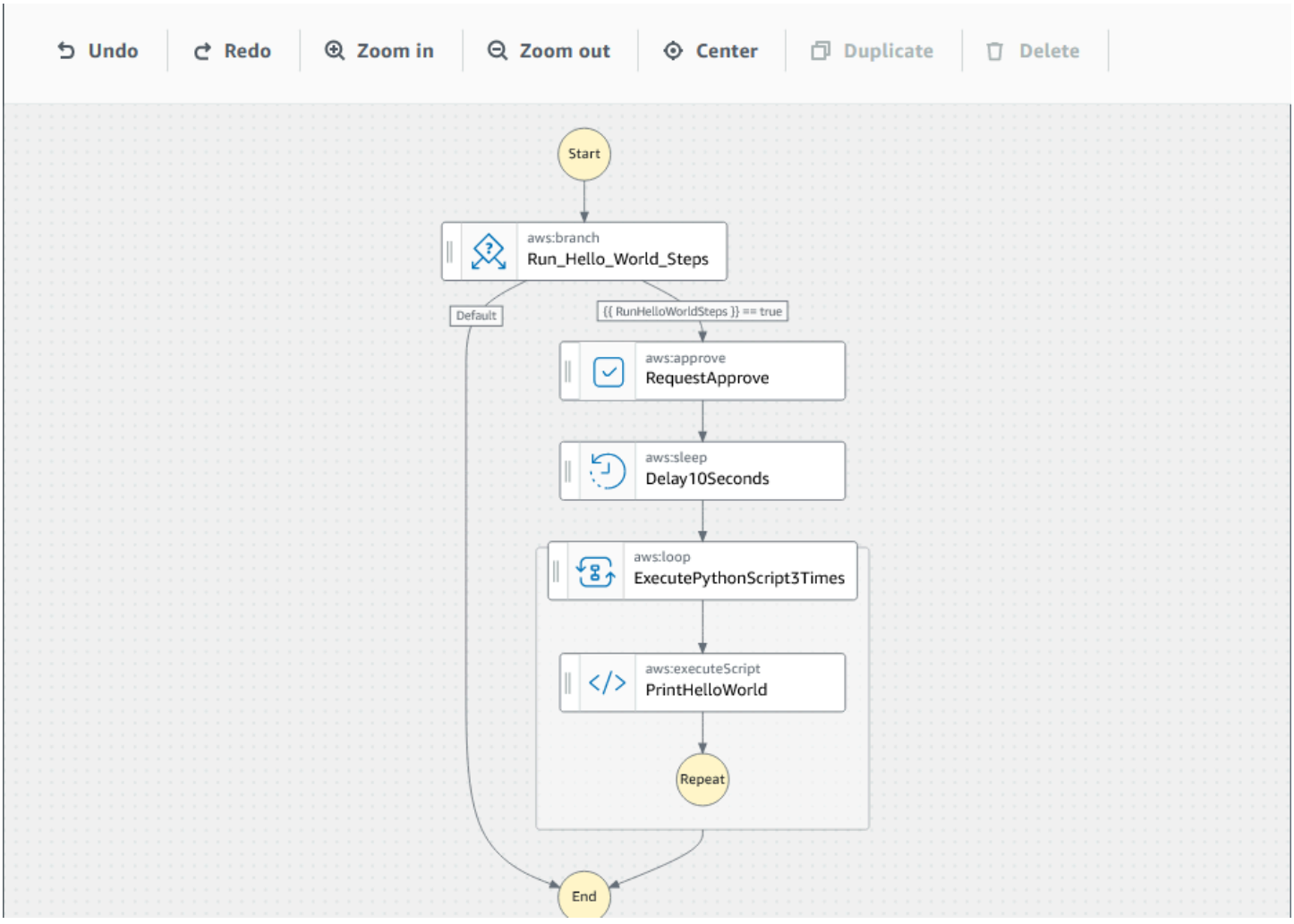
- A guia Ações fornece uma lista de ações de automação que você pode arrastar e soltar no gráfico do fluxo de trabalho do seu runbook na tela.
- A guia APIs da AWS fornece uma lista de APIs da AWS que você pode arrastar e soltar no gráfico do fluxo de trabalho do seu runbook na tela.
- A guia Runbooks fornece vários runbooks prontos para uso e reutilizáveis como blocos de construção que você pode usar em uma variedade de casos de uso. Por exemplo, você pode usar runbooks para realizar tarefas comuns de remediação em instâncias do Amazon EC2 em seu fluxo de trabalho sem precisar recriar as mesmas ações.



Canvas

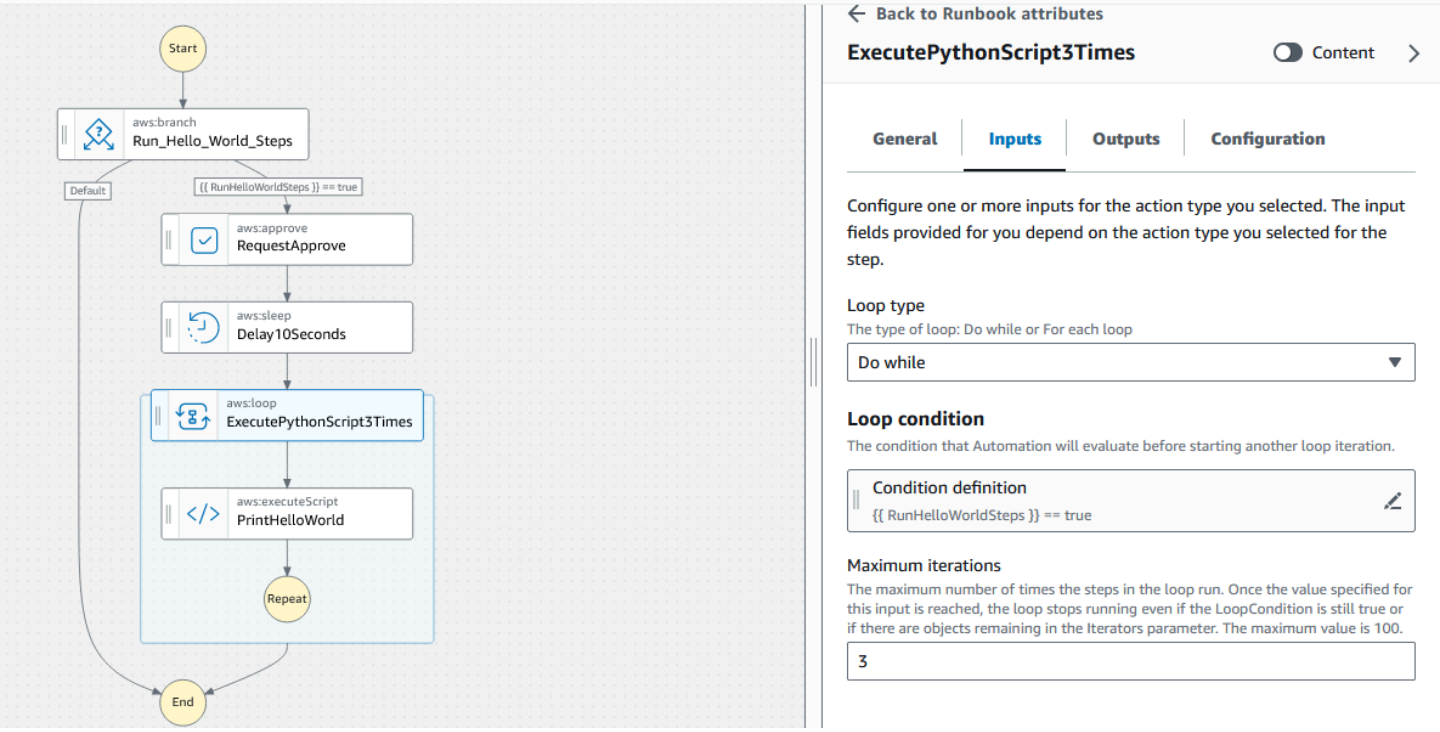
Depois de escolher uma ação para adicionar à sua automação, arraste-a para a tela e solte-a no gráfico do fluxo de trabalho. Você também pode arrastar e soltar ações para movê-las para lugares diferentes no fluxo de trabalho do seu runbook. Se o seu fluxo de trabalho for complexo, talvez não seja possível visualizar tudo no painel da tela. Use os controles na parte superior da tela para aumentar ou diminuir o zoom. Para visualizar diferentes partes de um fluxo de trabalho, é possível arrastar o gráfico do fluxo de trabalho na tela.

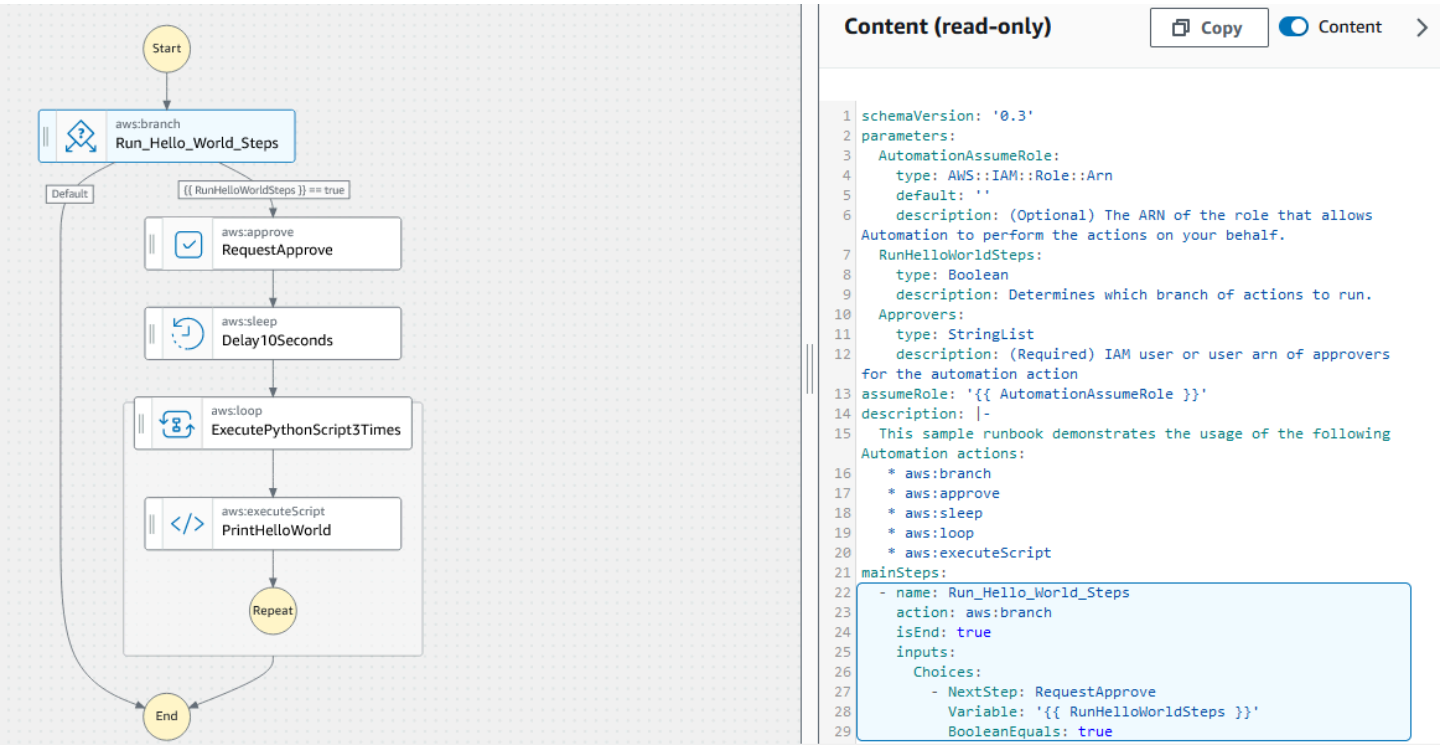
Arraste uma ação do navegador Ações e solte-a no gráfico do fluxo de trabalho do seu runbook. Uma linha mostra onde ela será colocada no fluxo de trabalho. Para alterar a ordem de uma ação, é possível arrastá-la para um local diferente no seu fluxo de trabalho. A nova ação foi adicionada ao seu fluxo de trabalho e seu código é gerado automaticamente.



Formulário

Após adicionar uma ação ao seu fluxo de trabalho do runbook, você pode configurá-la para atender ao seu caso de uso. Escolha a ação que você deseja configurar e você verá seus parâmetros e opções no painel Formulário. Você também pode ver o código YAML ou JSON escolhendo a opção Conteúdo. O código associado à ação que você selecionou é destacado.





Atalhos de teclado

A experiência de design visual é compatível com os atalhos de teclado mostrados na tabela a seguir.

Ação	Atalho de teclado
Desfazer a última operação mais recente.	Ctrl + Z
Refazer a última operação mais recente.	Ctrl + Shift + Z

Atalho de teclado	Ação
Alt + C	Centraliza o fluxo de trabalho na tela.
Backspace	Remover todos os estados selecionados.
Delete	Remover todos os estados selecionados.
Ctrl + D	Duplicar o estado selecionado.

Usar a experiência de design visual

Aprenda a criar, editar e executar fluxos de trabalho do runbook usando a experiência de design visual. Depois que seu fluxo de trabalho estiver pronto, você poderá salvá-lo ou exportá-lo. Você também pode usar a experiência de design visual para prototipagem rápida.

Criar um runbook

1. Faça login no console do Systems Manager Automation.
2. Escolha Criar runbook.
3. Na caixa Nome, insira um nome para o runbook, por exemplo, *MyNewRunbook*.
4. Ao lado do botão Design e Código, selecione o ícone de lápis e insira um nome para o seu runbook.

Agora você pode criar um fluxo de trabalho para seu novo runbook.

Desenvolver um runbook

Para criar um fluxo de trabalho de runbook usando a experiência de design visual, arraste uma ação de automação do navegador Ações para a tela, colocando-a onde desejar no fluxo de trabalho do seu runbook. Você também pode reordenar as ações em seu fluxo de trabalho arrastando-as para um local diferente. Quando você arrasta uma ação para a tela, uma linha aparece nos locais em que é possível soltar a ação no fluxo de trabalho. Após uma ação ser colocada na tela, seu código é gerado automaticamente e adicionado ao conteúdo do seu runbook.

Se você souber o nome da ação que deseja adicionar, use a caixa de pesquisa na parte superior do navegador Ações para encontrá-la.

Depois de soltar uma ação na tela, configure-a usando o painel Formulário à direita. Esse painel contém as guias Geral, Entradas, Saídas e Configuração para cada ação de automação ou ação de API que você coloca na tela. Por exemplo, a guia Geral consiste nas seguintes seções:

- O Nome da etapa identifica a etapa. Especifique um valor exclusivo para o nome da etapa.
- A Descrição ajuda você a descrever o que a ação está fazendo no fluxo de trabalho do seu runbook.

A guia Entradas contém campos que variam com base na ação. Por exemplo, a ação de automação `aws:executeScript` consiste nas seguintes seções:

- O Runtime é a linguagem que será usada para executar o script fornecido.
- O Manipulador é o nome da sua função. É necessário garantir que a função definida no manipulador tenha dois parâmetros: `events` e `context`. O runtime do PowerShell não é compatível com este parâmetro.
- O Script é um script incorporado que você deseja executar durante o fluxo de trabalho.
- (Opcional) O Anexo é para scripts autônomos ou arquivos.zip que podem ser invocados pela ação. Esse parâmetro é obrigatório para runbooks JSON.

A guia Saídas ajuda a especificar os valores que você deseja gerar de uma ação. Você pode referenciar valores de saída em ações posteriores do seu fluxo de trabalho ou gerar resultados de ações para fins de registro. Nem todas as ações terão uma guia Saídas porque nem todas as ações oferecem suporte a saídas. Por exemplo, a ação `aws:pause` não aceita saídas. Para ações que oferecem suporte a saídas, a guia Saídas consiste nas seguintes seções:

- O Nome é o nome a ser usado para o valor de saída. Você pode referenciar saídas em ações posteriores do seu fluxo de trabalho.
- O Seletor é uma string de expressão JSONPath que começa com `"$."` e é usada para selecionar um ou mais componentes em um elemento JSON.
- O Tipo é o tipo de dados para o valor de saída. Por exemplo, o tipo de dados `String` ou `Integer`.

A guia Configuração contém propriedades e opções que todas as ações de automação podem usar. A ação consiste nas seguintes seções:

- A propriedade Máximo de tentativas é o número tentativas de executar uma ação em caso de falha.
- A propriedade Tempo limite em segundos especifica o valor do tempo limite para uma ação.
- A propriedade É crítica determina se a falha na ação interrompe toda a automação.
- A propriedade Próxima etapa determina qual ação a automação executará em seguida no runbook.
- A propriedade Em falha determina qual ação a automação executará em seguida no runbook se a ação falhar.
- A propriedade Em cancelamento determina qual ação a automação executará em seguida no runbook se a ação for cancelada por um usuário.

Para excluir uma ação, você pode usar o backspace, a barra de ferramentas acima da tela ou clicar com o botão direito do mouse e escolher Excluir ação.

À medida que seu fluxo de trabalho cresce, ele pode não caber na tela. Para ajudar a ajustar o fluxo de trabalho à tela, tente uma das opções a seguir:

- Use os controles nos painéis laterais para redimensionar ou fechar os painéis.
- Use a barra de ferramentas na parte superior da tela para ampliar ou reduzir o gráfico do fluxo de trabalho.

Atualizar seu runbook

Você pode atualizar um fluxo de trabalho de runbook existente criando uma nova versão do seu runbook. As atualizações em seus runbooks podem ser feitas usando a experiência de design visual ou editando o código diretamente. Para atualizar um runbook existente, use o procedimento a seguir:

1. Faça login no console do Systems Manager Automation.
2. Escolha o runbook que deseja atualizar.
3. Escolha Create new version (Criar nova versão).
4. A experiência de design visual tem dois painéis: um painel de código e um painel de fluxo de trabalho visual. Escolha Design no painel de fluxo de trabalho visual para editar seu fluxo de trabalho com a experiência de design visual. Ao concluir, escolha Criar nova versão para salvar as alterações e sair.
5. (Opcional) Use o painel de código para editar o conteúdo do runbook em YAML ou JSON.

Exportar seu runbook

Para exportar o código YAML ou JSON do fluxo de trabalho do seu runbook e também um gráfico do seu fluxo de trabalho, use o procedimento a seguir:

1. Escolha seu runbook no console Documentos.
2. Escolha Create new version (Criar nova versão).
3. No menu suspenso Ações, escolha se você deseja exportar o gráfico ou o runbook, além do formato preferido.

Configurar entradas e saídas para suas ações

Cada ação de automação responde com base na entrada recebida. Na maioria dos casos, você envia a saída para as ações subsequentes. Na experiência de design visual, é possível configurar os dados de entrada e saída de uma ação nas guias Entradas e Saídas do painel Formulário.

Para obter informações detalhadas sobre como definir e usar a saída para ações de automação, consulte Uso de saídas de ações como entradas.

Forneça dados de entrada para uma ação

Cada ação de automação tem uma ou mais entradas para as quais você deve fornecer um valor. O valor que você fornece para a entrada de uma ação é determinado pelo tipo e formato de dados aceitos pela ação. Por exemplo, as ações `aws:sleep` exigem um valor de string formatado em ISO 8601 para a entrada `Duration`.

Geralmente, você usa ações no fluxo de trabalho do seu runbook que retornam a saída que você deseja usar em ações subsequentes. É importante garantir que seus valores de entrada estejam corretos para evitar erros no fluxo de trabalho do seu runbook. Os valores de entrada também são importantes porque determinam se a ação retorna a saída esperada. Por exemplo, ao usar a ação `aws:executeAwsApi`, você quer ter certeza de que está fornecendo o valor certo para a operação da API.

Definir dados de saída para uma ação

Algumas ações de automação retornam a saída após realizar suas operações definidas. As ações que retornam uma saída têm saídas predefinidas ou permitem que você mesmo as defina. Por exemplo, a ação `aws:createImage` tem saídas predefinidas que retornam um `ImageId` e `ImageState`. Comparativamente, com a ação `aws:executeAwsApi`, você pode definir as saídas que deseja da operação de API especificada. Como resultado, você pode retornar um ou mais valores de uma única operação de API para usar em ações subsequentes.

Definir suas próprias saídas para uma ação de automação exige que você especifique um nome da saída, o tipo de dados e o valor da saída. Para continuar usando a ação `aws:executeAwsApi` como exemplo, digamos que você esteja chamando a operação da API `DescribeInstances` do Amazon EC2. Neste exemplo, você deseja retornar, ou gerar como saída, o `State` de uma instância do Amazon EC2 e ramificar o fluxo de trabalho do seu runbook com base na saída. Você escolhe nomear a saída **`InstanceState`** e usar o tipo de dados **`String`**.

O processo para definir o valor real da saída é diferente dependendo da ação. Por exemplo, se você estiver usando a ação `aws:executeScript`, deverá usar instruções `return` em

suas funções para fornecer dados às saídas. Com outras ações como `aws:executeAwsApi`, `aws:waitForAwsResourceProperty` e `aws:assertAwsResourceProperty`, um `Selector` é necessário. O `Selector`, ou `PropertySelector` como algumas ações se referem a ele, é uma string JSONPath usada para processar a resposta JSON de uma operação de API. É importante entender como o objeto de resposta JSON de uma operação de API é estruturado para que você possa selecionar o valor correto para sua saída. Usando a operação de API `DescribeInstances` mencionada anteriormente, veja o exemplo de resposta JSON a seguir:


```
{
  "reservationSet": {
    "item": {
      "reservationId": "r-1234567890abcdef0",
      "ownerId": 123456789012,
      "groupSet": "",
      "instancesSet": {
        "item": {
          "instanceId": "i-1234567890abcdef0",
          "imageId": "ami-bff32ccc",
          "instanceState": {
            "code": 16,
            "name": "running"
          },
          "privateDnsName": "ip-192-168-1-88.eu-west-1.compute.internal",
          "dnsName": "ec2-54-194-252-215.eu-west-1.compute.amazonaws.com",
          "reason": "",
          "keyName": "my_keypair",
          "amiLaunchIndex": 0,
          "productCodes": "",
          "instanceType": "t2.micro",
          "launchTime": "2018-05-08T16:46:19.000Z",
          "placement": {
            "availabilityZone": "eu-west-1c",
            "groupName": "",
            "tenancy": "default"
          },
          "monitoring": {
            "state": "disabled"
          },
          "subnetId": "subnet-56f5f000",
          "vpcId": "vpc-11112222",
          "privateIpAddress": "192.168.1.88",
          "ipAddress": "54.194.252.215",
```

```
          "sourceDestCheck": true,
          "groupSet": {
            "item": {
              "groupId": "sg-e4076000",
              "groupName": "SecurityGroup1"
            }
          },
          "architecture": "x86_64",
          "rootDeviceType": "ebs",
          "rootDeviceName": "/dev/xvda",
          "blockDeviceMapping": {
            "item": {
              "deviceName": "/dev/xvda",
              "ebs": {
                "volumeId": "vol-1234567890abcdef0",
                "status": "attached",
                "attachTime": "2015-12-22T10:44:09.000Z",
                "deleteOnTermination": true
              }
            }
          },
          "virtualizationType": "hvm",
          "clientToken": "xMcwG14507example",
          "tagSet": {
            "item": {
              "key": "Name",
              "value": "Server_1"
            }
          },
          "hypervisor": "xen",
          "networkInterfaceSet": {
            "item": {
              "networkInterfaceId": "eni-551ba000",
              "subnetId": "subnet-56f5f000",
              "vpcId": "vpc-11112222",
              "description": "Primary network interface",
              "ownerId": 123456789012,
              "status": "in-use",
              "macAddress": "02:dd:2c:5e:01:69",
              "privateIpAddress": "192.168.1.88",
              "privateDnsName": "ip-192-168-1-88.eu-west-1.compute.internal",
              "sourceDestCheck": true,
              "groupSet": {
                "item": {
```

```
                "groupId": "sg-e4076000",
                "groupName": "SecurityGroup1"
              }
            },
            "attachment": {
              "attachmentId": "eni-attach-39697adc",
              "deviceIndex": 0,
              "status": "attached",
              "attachTime": "2018-05-08T16:46:19.000Z",
              "deleteOnTermination": true
            },
            "association": {
              "publicIp": "54.194.252.215",
              "publicDnsName": "ec2-54-194-252-215.eu-west-1.compute.amazonaws.com",
              "ipOwnerId": "amazon"
            },
            "privateIpAddressesSet": {
              "item": {
                "privateIpAddress": "192.168.1.88",
                "privateDnsName": "ip-192-168-1-88.eu-west-1.compute.internal",
                "primary": true,
                "association": {
                  "publicIp": "54.194.252.215",
                  "publicDnsName": "ec2-54-194-252-215.eu-
west-1.compute.amazonaws.com",
                  "ipOwnerId": "amazon"
                }
              }
            },
            "ipv6AddressesSet": {
              "item": {
                "ipv6Address": "2001:db8:1234:1a2b::123"
              }
            }
          }
        },
        "iamInstanceProfile": {
          "arn": "arn:aws:iam::123456789012:instance-profile/AdminRole",
          "id": "ABCAJEDNCAA64SSD123AB"
        },
        "ebsOptimized": false,
        "cpuOptions": {
          "coreCount": 1,
          "threadsPerCore": 1
```

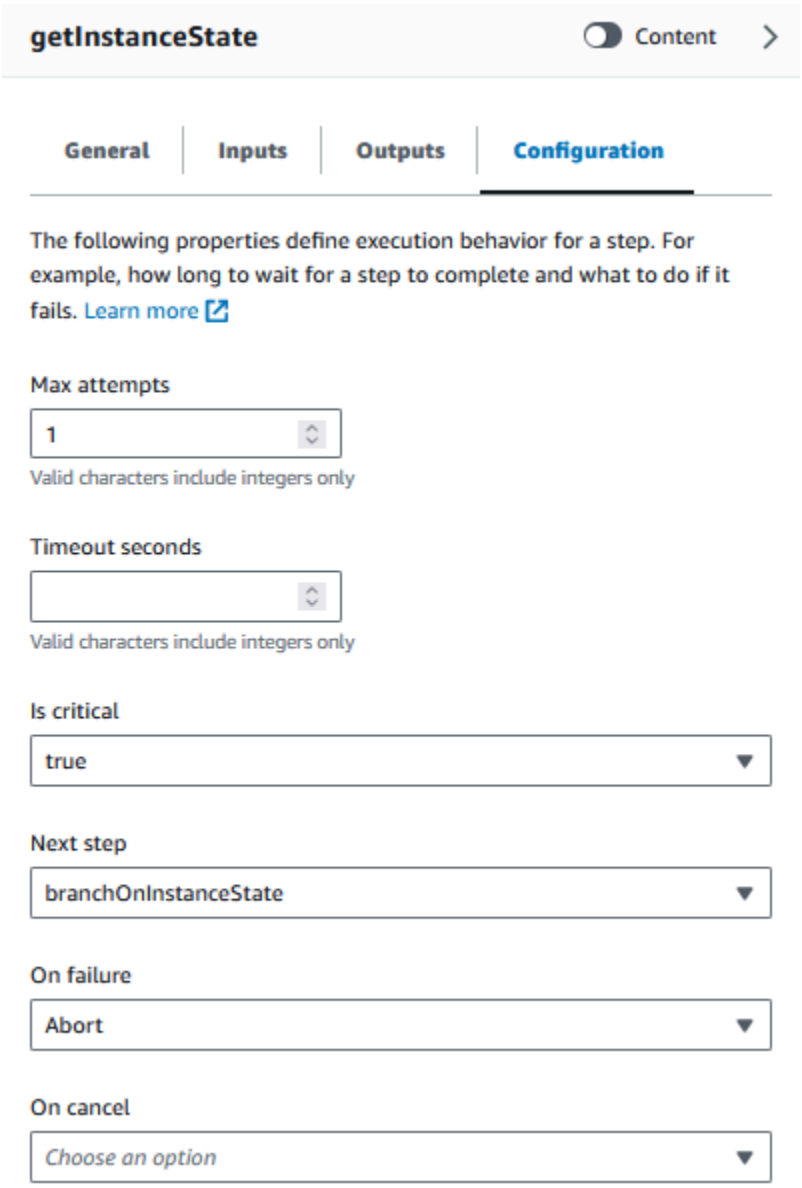
```
            }
          }
        }
      }
    }
}
```

No objeto de resposta JSON, a instância `State` está aninhada em um objeto `Instances` que está aninhado no objeto `Reservations`. Para retornar o valor da instância `State`, use a seguinte string para `Selector` para que o valor possa ser usado em nossa saída: **`$.Reservations[0].Instances[0].State.Name`**.

Para referenciar um valor de saída em ações subsequentes do fluxo de trabalho do seu runbook, o seguinte formato é usado: `{{ StepName.NameOfOutput }}`. Por exemplo, **`{{ GetInstanceState.InstanceState }}`**. Na experiência de design visual, é possível escolher valores de saída para usar em ações subsequentes utilizando o menu suspenso para a entrada. Ao usar saídas em ações subsequentes, o tipo de dados da saída deve corresponder ao tipo de dados da entrada. Neste exemplo, a saída `InstanceState` é uma `String`. Portanto, para usar o valor na entrada de uma ação subsequente, a entrada deve aceitar uma `String`.

Tratamento de erros com a experiência de design visual

Por padrão, quando uma ação relata um erro, o Automation interrompe totalmente o fluxo de trabalho do runbook. Isso ocorre porque o valor padrão da propriedade `onFailure` em todas as ações é `Abort`. É possível configurar a forma como a automação lida com erros no fluxo de trabalho do seu runbook. Mesmo que você tenha configurado o tratamento de erros, alguns erros ainda podem causar uma falha na automação. Para ter mais informações, consulte Solução de problemas do Systems Manager Automation. Na experiência de design visual, você configura o tratamento de erros no painel Configuração.



Repetir a ação em caso de erro

Para repetir uma ação em caso de erro, especifique um valor para a propriedade Máximo de tentativas. O valor padrão é 1. Se o valor for maior que 1, a etapa não será considerada como em falha até que todas as novas tentativas tenham falhado.

Tempo limite

Você pode configurar um tempo limite para as ações para definir o número máximo de segundos que sua ação pode ser executada antes que ela falhe. Para configurar um tempo limite, insira o número de segundos que sua ação deve esperar antes que a ação falhe na propriedade Tempo limite em

segundos. Se o tempo limite for atingido e o valor de `Max attempts` for maior que 1, a etapa não será considerada expirada até que todas as novas tentativas tenham sido feitas.

Ações com falha

Por padrão, quando uma ação falha, o Automation interrompe totalmente o fluxo de trabalho do runbook. Você pode modificar esse comportamento especificando um valor alternativo para a propriedade Em falha das ações em seu runbook. Se desejar que o fluxo de trabalho continue na próxima etapa do runbook, escolha Continuar. Se desejar que o fluxo de trabalho pule para outra etapa subsequente no runbook, escolha Etapa e insira o nome da etapa.

Ações canceladas

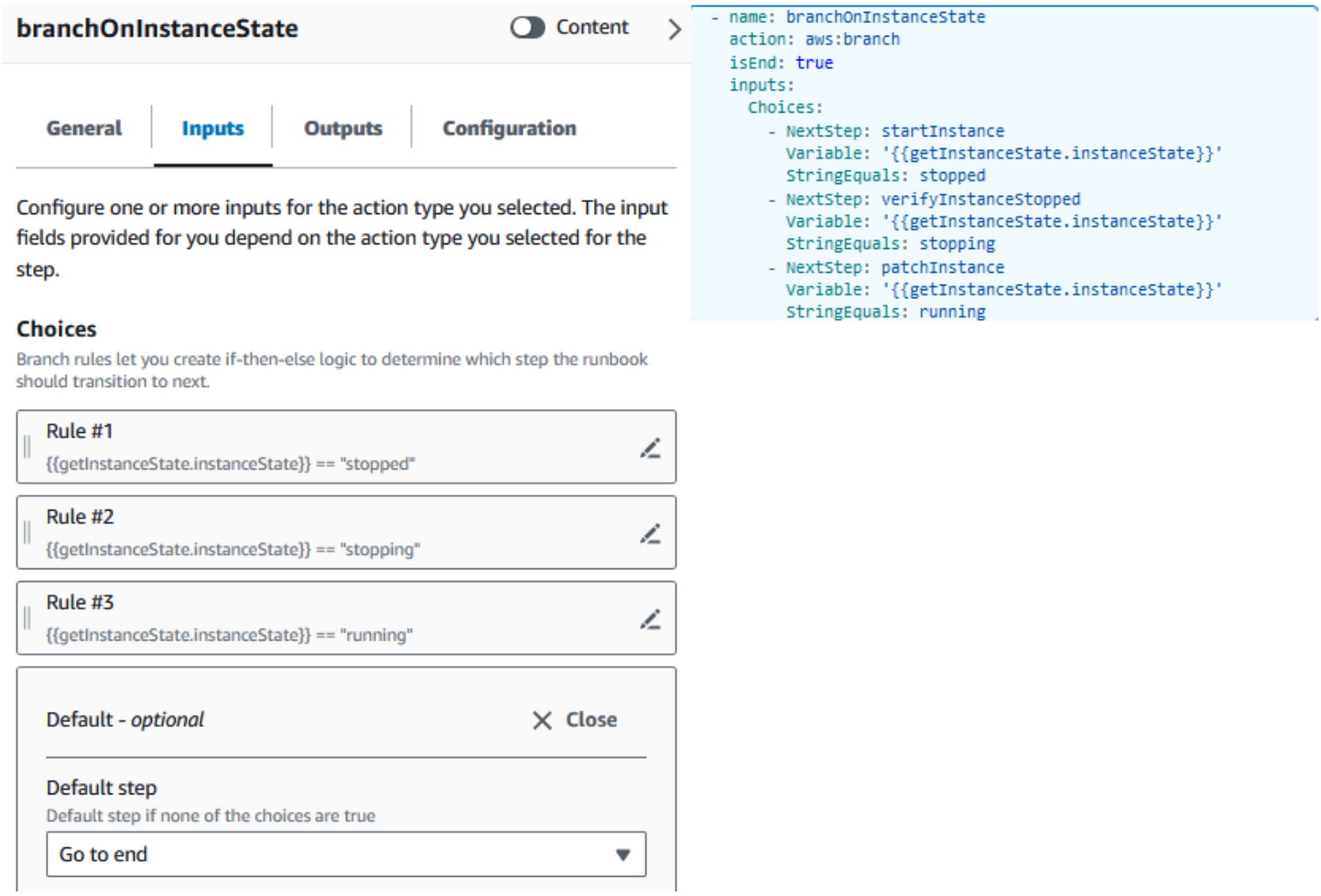
Por padrão, quando uma ação é cancelada por um usuário, o Automation interrompe totalmente o fluxo de trabalho do runbook. Você pode modificar esse comportamento especificando um valor alternativo para a propriedade Em cancelamento das ações em seu runbook. Se desejar que o fluxo de trabalho pule para outra etapa subsequente no runbook, escolha Etapa e insira o nome da etapa.

Ações críticas

Você pode designar uma ação como crítica, o que significa que ela determina o status geral dos relatórios de sua automação. Se uma etapa com essa designação falhar, o Automation relatará o status final como `Failed`, independentemente do sucesso de outras ações. Para configurar uma ação como crítica, mantenha o valor padrão como Verdadeiro para a propriedade É crítica.

Enceramento de ações

A propriedade É o fim interrompe automação no final de determinada ação. O valor padrão da propriedade é `false`. Se você configurar essa propriedade para uma ação, a automação será interrompida se a ação for bem-sucedida ou falhar. Essa propriedade é mais frequentemente usada com ações `aws:branch` para lidar com valores de entrada inesperados ou indefinidos. O exemplo a seguir mostra um runbook que espera um estado de instância de `running`, `stopping` ou `stopped`. Se uma instância estiver em um estado diferente, a automação será encerrada.



Tutorial: criar um runbook usando a experiência de design visual

Neste tutorial, você aprenderá o básico de como trabalhar com a experiência de design visual fornecida pelo Systems Manager Automation. Na experiência de design visual, é possível criar um runbook que usa várias ações. Você usará o recurso de arrastar e soltar para organizar as ações na tela. Você também pesquisará, selecionará e configurará essas ações. Em seguida, você poderá visualizar o código YAML gerado automaticamente para o fluxo de trabalho do seu runbook, sair da experiência de design visual, executar o runbook e revisar os detalhes da execução.

Este tutorial também mostra como atualizar o runbook e visualizar a nova versão. No final do tutorial, você executa uma etapa de limpeza e exclui seu runbook.

Após concluir este tutorial, você saberá como usar a experiência de design visual para criar um runbook. Você também aprenderá a atualizar, executar e excluir seu runbook.

> **Note**
>
> Antes de começar este tutorial, certifique-se de concluir a Configurar a automação.

Tópicos

- Etapa 1: navegue até a experiência de design visual
- Etapa 2: criar um fluxo de trabalho
- Etapa 3: analisar o código gerado automaticamente
- Etapa 4: executar seu novo runbook
- Etapa 5: limpar

Etapa 1: navegue até a experiência de design visual

1. Faça login no console do Systems Manager Automation.
2. Escolha Criar runbook de automação.

Etapa 2: criar um fluxo de trabalho

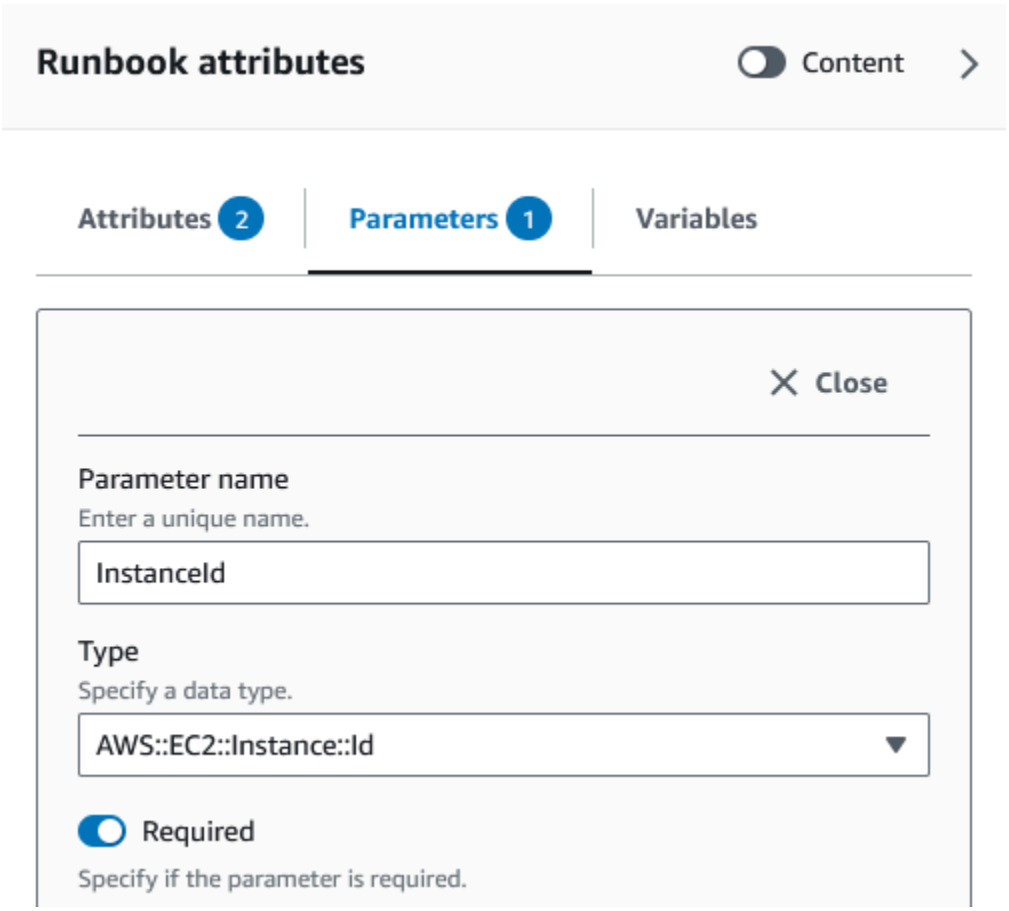
Na experiência de design visual, um fluxo de trabalho é uma representação gráfica do seu runbook na tela. É possível usar a experiência de design visual para definir, configurar e examinar as ações individuais do seu runbook.

Para criar um fluxo de trabalho

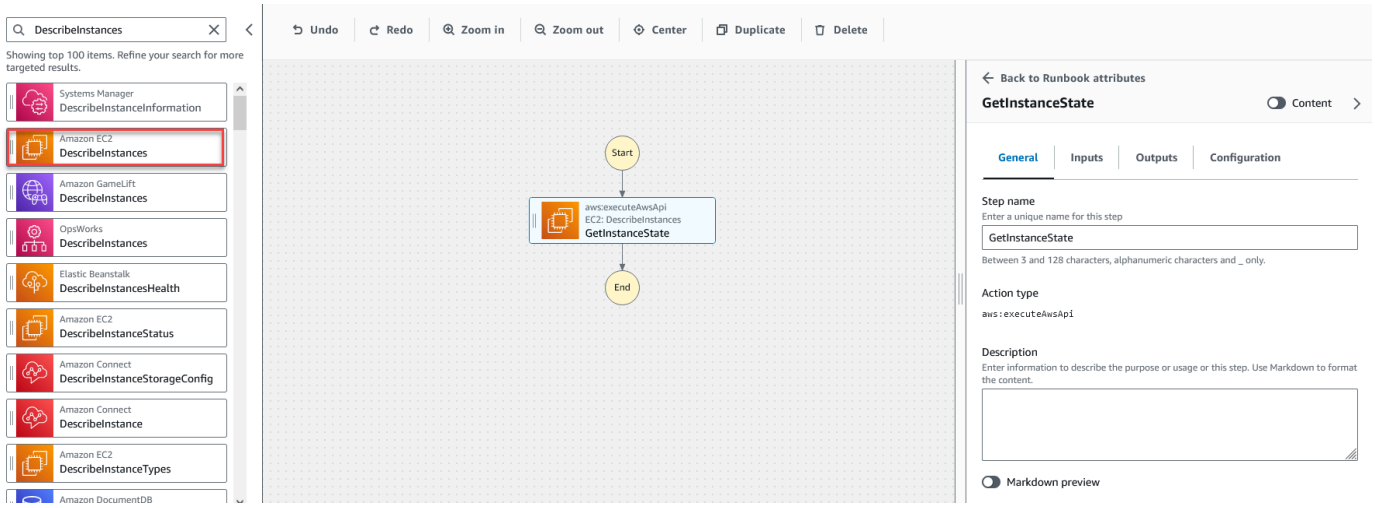
1. Ao lado do botão Design e Código, selecione o ícone de lápis e insira um nome para o seu runbook. Para este tutorial, insira **`VisualDesignExperienceTutorial`**.



2. Na seção Atributos do documento do painel Formulário, expanda a lista suspensa Parâmetros de entrada e selecione Adicionar um parâmetro.
   a. No campo Nome do parâmetro, insira **`InstanceId`**.
   b. No menu suspenso Tipo, escolha AWS::EC2::Instance.
   c. Selecione o botão Obrigatório.



3. No navegador APIs da AWS, insira **`DescribeInstances`** na barra de pesquisa.
4. Arraste uma ação Amazon EC2 – DescribeInstances para a tela vazia.
5. Em Nome da etapa, insira um valor. Para este tutorial, use o nome **`GetInstanceState`**.



a. Expanda a lista suspensa Entradas adicionais e, no campo Nome da entrada, insira **`InstanceIds`**.

b. Escolha a guia Entradas.

c. No campo Valor de entrada, escolha a entrada do documento **`InstanceId`**. Isso faz referência ao valor do parâmetro de entrada criado por você no início do procedimento. Como a entrada InstanceIDs da ação `DescribeInstances` aceita valores `StringList`, é necessário colocar a entrada InstanceID entre colchetes. O YAML para o Valor de entrada deve corresponder ao seguinte: **`['{{ InstanceId }}']`**.

d. Na guia Saídas, selecione Adicionar uma saída e insira **`InstanceState`** no campo Nome.

e. No campo Seletor, insira **`$.Reservations[0].Instances[0].State.Name`**.

f. No menu suspenso Tipo, escolha String.

6. Arraste uma ação de Ramificação do navegador Ações e solte-a abaixo da etapa **`GetInstanceState`**.

7. Em Nome da etapa, insira um valor. Neste tutorial, use o nome **`BranchOnInstanceState`**.

   Para definir a lógica de ramificação, faça o seguinte:

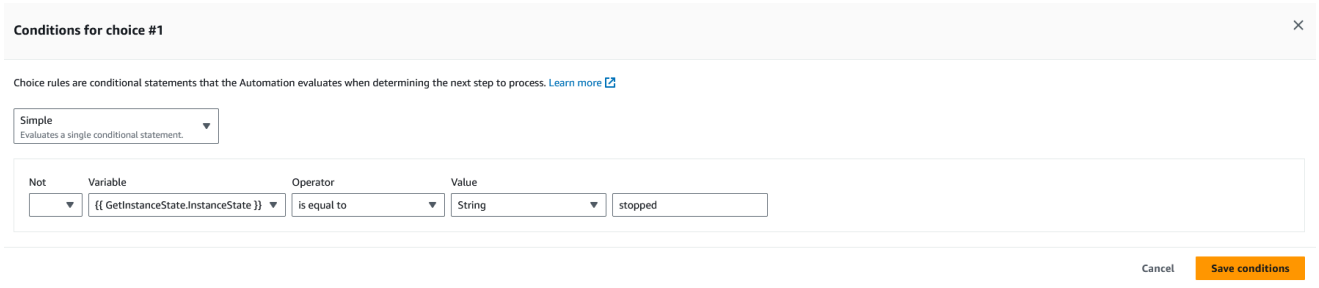
   a. Escolha o estado **`Branch`** na tela. Em seguida, em Entradas e Escolhas, selecione o ícone de lápis para editar a Regra #1.

   b. Escolha Adicionar condições.

   c. Na caixa de diálogo Condições para a regra #1, escolha a saída da etapa **`GetInstanceState.InstanceState`** no menu suspenso Variável.

   d. Em Operador, escolha é igual a.

   e. Em Valor, escolha String na lista suspensa. Insira **`stopped`**.



   f. Selecione Salvar condições.

   g. Escolha Adicionar nova regra de escolha.

   h. Escolha Adicionar condições para a Regra #2.

i. Na caixa de diálogo Condições para a regra #2, escolha a saída da etapa **`GetInstanceState.InstanceState`** no menu suspenso Variável.

j. Em Operador, escolha é igual a.

k. Em Valor, escolha String na lista suspensa. Insira **`stopping`**.

l. Selecione Salvar condições.

m. Escolha Adicionar nova regra de escolha.

n. Para a Regra #3, escolha Adicionar condições.

o. Na caixa de diálogo Condições para a regra #3, escolha a saída da etapa **`GetInstanceState.InstanceState`** no menu suspenso Variável.

p. Em Operador, escolha é igual a.

q. Em Valor, escolha String na lista suspensa. Insira **`running`**.

r. Selecione Salvar condições.

s. Na Regra padrão, escolha Ir para o final em Etapa padrão.

8. Arraste a ação Alterar um estado de instância para a caixa vazia Arraste a ação aqui sob a condição {{GetInstanceState.InstanceState}} == "stopped".

   a. Em Nome da etapa, insira **`StartInstance`**.

   b. Na guia Entradas, em IDs de instância, escolha o valor de entrada do documento InstanceId no menu suspenso.

   c. Para Estado desejado, especifique **`running`**.

9. Arraste a ação Aguardar recurso da AWS para a caixa vazia Arraste a ação aqui sob a condição {{ GetInstanceState.InstanceState }} == "stopping".

10. Em Nome da etapa, insira um valor. Neste tutorial, use o nome **`WaitForInstanceStop`**.

    a. No campo Serviço, escolha Amazon EC2.

    b. No campo API, escolha DescribeInstances.

    c. No campo Seletor de propriedades, insira **`$.Reservations[0].Instances[0].State.Name`**.

    d. Para o parâmetro Valores desejados, insira **`["stopped"]`**.

    e. Na guia Configuração da ação WaitForInstanceStop, escolha StartInstance no menu suspenso Próxima etapa.

11. Arraste a ação Executar comando em instâncias para a caixa vazia Arraste a ação aqui sob a condição {{GetInstanceState.InstanceState}} == "running".

12. Em Nome da etapa, insira **`SayHello`**.

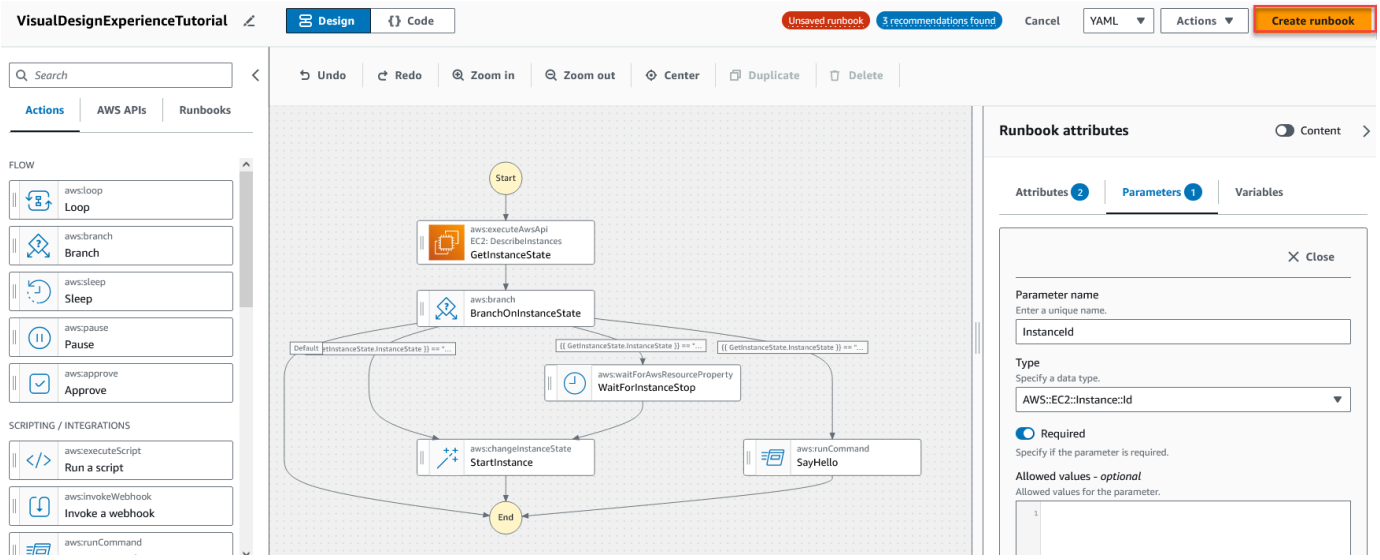
    a. Na guia Entradas, insira **`AWS-RunShellScript`** para o parâmetro Nome do documento.

    b. Para InstanceIds, escolha o valor de entrada do documento InstanceId no menu suspenso.

    c. Expanda o menu suspenso Entradas adicionais e, no menu suspenso Nome da entrada, escolha Parâmetros.

    d. No campo Valore de entrada, insira **`{"commands": "echo 'Hello World'"}`**.

13. Revise o runbook concluído na tela e selecione Criar runbook para salvar o runbook do tutorial.



Etapa 3: analisar o código gerado automaticamente

Conforme você arrasta e solta ações do navegador Ações na tela, a experiência de design visual compõe automaticamente o conteúdo YAML ou JSON do seu runbook em tempo real. Você pode visualizar e editar esse código. Para visualizar o código gerado automaticamente, selecione Código na opção Design e Código.

Etapa 4: executar seu novo runbook

Após criar seu runbook, você poderá executar a automação.

Para executar seu novo runbook de automação

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, selecione Automation e Execute automation (Executar automação).

3. Na lista Automation document (Documento do Automation), escolha um runbook. Escolha uma ou mais opções no painel Document categories (Categorias de documentos) para filtrar documentos SSM de acordo com sua finalidade. Para visualizar um runbook que você tenha, escolha a guia Owned by me (De minha propriedade). Para visualizar um runbook compartilhado com sua conta, escolha a guia Shared with me (Compartilhado comigo). Para visualizar todos os runbooks, escolha a guia All documents (Todos os documentos).

   > **Note**
   >
   > Você pode visualizar informações sobre um runbook, selecionando o nome dele.

4. Na seção Document details (Detalhes do documento), verifique se Document version (Versão do documento) está definida como a versão que você quer executar. O sistema inclui as seguintes opções de versão:

   - Versão padrão no runtime: escolha essa opção se o runbook do Automation for atualizado periodicamente e uma nova versão padrão for atribuída.
   - Versão mais recente no runtime: escolha essa opção se o runbook do Automation for atualizado periodicamente e se você quiser executar a versão mais recente.
   - 1 (padrão): escolha esta opção para executar a primeira versão do documento, que é a versão padrão.

5. Escolha Próximo.
6. Na página Executar runbook de automação, escolha Execução simples.
7. Na seção Input parameters (Parâmetros de entrada), especifique as entradas necessárias. Opcionalmente, você pode escolher uma função de serviço do IAM na lista AutomationAssumeRole.
8. (Opcional) Escolha um alarme do Amazon CloudWatch a fim de aplicar à sua automação para monitoramento. Para anexar um alarme do CloudWatch à sua automação, a entidade principal do IAM que inicia a automação deve ter permissão para a ação `iam:createServiceLinkedRole`. Para obter mais informações sobre alarmes do CloudWatch, consulte Usar alarmes do Amazon CloudWatch. A automação será interrompida se o alarme for ativado. Se você usar o AWS CloudTrail, você verá a chamada de API em sua trilha.
9. Clique em Executar.

Etapa 5: limpar

Para excluir seu runbook

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Documents.
3. Selecione a guia Pertencente a mim.
4. Localize o runbook VisualDesignExperienceTutorial.
5. Selecione o botão na página do cartão do documento e escolha Excluir documento no menu suspenso Ações.

Criar runbooks do Automation

Cada runbook no Automation, uma ferramenta do AWS Systems Manager, define uma automação. Os runbooks do Automation definem as ações que são executadas durante uma automação. No conteúdo do runbook, você define os parâmetros de entrada, saídas e ações que o Systems Manager realiza nas suas instâncias gerenciadas e os recursos da AWS.

A Automação inclui vários runbooks predefinidos que você pode usar para executar tarefas comuns, como reiniciar uma ou mais instâncias do Amazon Elastic Compute Cloud (Amazon EC2) ou criar uma Amazon Machine Image (AMI). No entanto, seus casos de uso podem se estender além dos recursos dos runbooks predefinidos. Se esse for o caso, você pode criar seus próprios runbooks e modificá-los de acordo com suas necessidades.

Um runbook consiste em ações de automação, parâmetros para essas ações e parâmetros de entrada especificados. O conteúdo de um runbook é escrito em YAML ou JSON. Se você não estiver familiarizado com YAML ou JSON, recomendamos usar o designer visual ou aprender mais sobre qualquer linguagem de marcação antes de tentar criar seu próprio runbook. Para obter mais informações sobre o designer visual, consulte Experiência de design visual para runbooks de automação.

As seções a seguir ajudarão você a criar seu primeiro runbook.

Identifique seu caso de uso

A primeira etapa na criação de um runbook é identificar o caso de uso. Por exemplo, você agendou o runbook `AWS-CreateImage` para ser executado diariamente em todas as instâncias do Amazon EC2 de produção. No final do mês, você percebe que tem mais imagens do que são necessárias para obter pontos de recuperação. No futuro, você precisará excluir automaticamente a AMI mais

antiga de uma instância do Amazon EC2, quando uma nova AMI for criada. Para fazer isso, você cria um novo runbook que faz o seguinte:

1. Executa a ação `aws:createImage` e especifica o ID da instância na descrição da imagem.
2. Executa a ação `aws:waitForAwsResourceProperty` para sondar o estado da imagem até que ela esteja `available`.
3. Depois que o estado da imagem estiver `available`, o `aws:executeScript` executa um script Python personalizado que reúne os IDs de todas as imagens associadas à sua instância do Amazon EC2. O script faz isso filtrando, usando o ID da instância na descrição da imagem que você especificou na criação. Em seguida, o script classifica a lista de IDs de imagem com base no `creationDate` da imagem e emite o ID da AMI mais antiga.
4. Por último, a ação `aws:deleteImage` é executada para excluir a AMI mais antiga, usando o ID do resultado da etapa anterior.

Nesse cenário, você já estava usando o runbook `AWS-CreateImage`, mas descobriu que seu caso de uso exigia maior flexibilidade. Esta é uma situação comum porque pode haver sobreposição entre runbooks e ações de automação. Como resultado, talvez seja necessário ajustar quais runbooks ou ações você deve usar para resolver seu caso de uso.

Por exemplo, as ações `aws:executeScript` e `aws:invokeLambdaFunction` permitem que você execute scripts personalizados como parte de sua automação. Para escolher entre eles, você pode preferir o `aws:invokeLambdaFunction` devido às linguagens do runtime adicionais suportadas. No entanto, você pode preferir o `aws:executeScript`, porque permite criar conteúdo de script diretamente em runbooks YAML e fornecer conteúdo de script como anexos para runbooks JSON. Você também pode considerar o `aws:executeScript` para ser mais simples em termos de configuração do AWS Identity and Access Management (IAM). Como ele usa as permissões fornecidas no `AutomationAssumeRole`, o `aws:executeScript` não requer uma função do AWS Lambda de execução da função.

Em qualquer cenário específico, uma ação pode fornecer mais flexibilidade, ou funcionalidade adicional, sobre outra. Portanto, recomendamos que você revise os parâmetros de entrada disponíveis para o runbook ou ação que quiser usar, para determinar qual melhor se adapta ao seu caso de uso e preferências.

Configuração do ambiente de desenvolvimento

Depois de identificar seu caso de uso e os runbooks predefinidos ou ações de automação que deseja usar no runbook, é hora de configurar seu ambiente de desenvolvimento para o conteúdo do

runbook. Para desenvolver o conteúdo do runbook, recomendamos usar o AWS Toolkit for Visual Studio Code em vez do console de documentos do Systems Manager.

O Toolkit for VS Code é uma extensão de código aberto para o Visual Studio Code (VS Code) que oferece mais recursos do que o console Systems Manager Documents. Os recursos úteis incluem validação de esquema para YAML e JSON, trechos para tipos de ação de automação e suporte de preenchimento automático para várias opções em YAML e JSON.

Para obter mais informações sobre como instalar o Toolkit for VS Code, consulte Instalar o AWS Toolkit for Visual Studio Code. Para obter mais informações sobre como usar o Toolkit for VS Code para desenvolver runbooks, consulte Trabalhar com documentos do Systems Manager Automation no Manual do usuário do AWS Toolkit for Visual Studio Code.

Desenvolva conteúdo do runbook

Com o caso de uso identificado e o ambiente configurado, você está pronto para desenvolver o conteúdo do runbook. Seu caso de uso e suas preferências ditarão em grande parte as ações de automação ou os runbooks que você usa no conteúdo do runbook. Algumas ações suportam apenas um subconjunto de parâmetros de entrada quando comparadas a outra ação que permite realizar uma tarefa semelhante. Outras ações têm resultados específicos, como `aws:createImage`, onde algumas ações permitem que você defina suas próprias saídas, como `aws:executeAwsApi`.

Se você não tiver certeza de como usar uma ação específica em seu runbook, recomendamos analisar a entrada correspondente para a ação na seção Referência de ações do Systems Manager Automation. Recomendamos também examinar o conteúdo de runbooks predefinidos para ver exemplos reais de como essas ações são usadas. Para obter mais exemplos de aplicações reais de runbooks, consulte Exemplos adicionais de runbook.

Para demonstrar as diferenças de simplicidade e flexibilidade que o conteúdo do runbook fornece, os tutoriais a seguir fornecem um exemplo de como aplicar patches a grupos de instâncias do Amazon EC2 em etapas:

- the section called “Exemplo 1: criação de runbooks pai-filho”: neste exemplo, dois runbooks são usados em um relacionamento pai-filho. O runbook pai inicia uma automação de controle de taxa do runbook filho.
- the section called “Exemplo 2: Runbook com script”: este exemplo demonstra como você pode realizar as mesmas tarefas do Exemplo 1 condensando o conteúdo em um único runbook e usando scripts em seu runbook.

Exemplo 1: criação de runbooks pai-filho

O exemplo a seguir demonstra como criar dois runbooks que corrigem grupos marcados de instâncias do Amazon Elastic Compute Cloud (Amazon EC2) em estágios. Esses runbooks são usados em um relacionamento pai-filho com o runbook pai usado para iniciar uma automação de controle de taxa do runbook filho. Para obter mais informações sobre automações de controle de taxas, consulte Executar operações automatizadas em escala. Para obter mais informações sobre as ações de automação usadas neste exemplo, consulte a Referência de ações do Systems Manager Automation.

Crie o runbook filho

Este exemplo de runbook aborda o seguinte cenário. Emily é engenheira de sistemas na AnyCompany Consultants, LLC. Ela precisa configurar patches para grupos de instâncias do Amazon Elastic Compute Cloud (Amazon EC2) que hospedam bancos de dados primários e secundários. As aplicações acessam esses bancos de dados 24 horas por dia, portanto, uma das instâncias do banco de dados deve estar sempre disponível.

Ela decide que a aplicação de patches nas instâncias em etapas é a melhor abordagem. O grupo principal de instâncias de banco de dados receberão o patch primeiro, seguido pelo grupo secundário de instâncias de banco de dados. Além disso, para evitar custos adicionais, deixando instâncias em execução que foram interrompidas anteriormente, Emily quer que as instâncias corrigidas sejam retornadas ao seu estado original, antes da aplicação de patches.

Emily identifica os grupos primário e secundário de instâncias de banco de dados pelas tags associadas às instâncias. Ela decide criar um runbook pai que inicia uma automação de controle de taxa de um runbook filho. Ao fazer isso, ela pode direcionar as tags associadas aos grupos primário e secundário de instâncias do banco de dados e gerenciar a simultaneidade das automações filho. Depois de analisar os documentos do Systems Manager (SSM) disponíveis para aplicação de patches, ela escolhe o documento `AWS-RunPatchBaseline`. Usando este documento do SSM, seus colegas podem revisar as informações de conformidade de patch associadas após a conclusão da operação de patch.

Para começar a criar o conteúdo do runbook, Emily analisa as ações de automação disponíveis e começa a criar o conteúdo para o runbook filho da seguinte maneira:

1. Primeiro, ela fornece valores para o esquema e a descrição do runbook e define os parâmetros de entrada para o runbook filho.

Utilizando o parâmetro `AutomationAssumeRole`, Emily e seus colegas podem usar uma função do IAM existente que permite que o Automation realize ações no runbook em seu nome. Emily usa o parâmetro `InstanceId` para determinar a instância que deve ser corrigida. Opcionalmente, os parâmetros `Operation`, `RebootOption` e `SnapshotId` podem ser usados para fornecer valores para documentar os parâmetros para `AWS-RunPatchBaseline`. Para evitar que valores inválidos sejam fornecidos a esses parâmetros do documento, ela define a propriedade `allowedValues` conforme necessário.

YAML

```
schemaVersion: '0.3'
description: 'An example of an Automation runbook that patches groups of Amazon
 EC2 instances in stages.'
assumeRole: '{{AutomationAssumeRole}}'
parameters:
  AutomationAssumeRole:
    type: String
    description: >-
      '(Optional) The Amazon Resource Name (ARN) of the IAM role that allows
 Automation to perform the
      actions on your behalf. If no role is specified, Systems Manager
      Automation uses your IAM permissions to operate this runbook.'
    default: ''
  InstanceId:
    type: String
    description: >-
      '(Required) The instance you want to patch.'
  SnapshotId:
    type: String
    description: '(Optional) The snapshot ID to use to retrieve a patch baseline
 snapshot.'
    default: ''
  RebootOption:
    type: String
    description: '(Optional) Reboot behavior after a patch Install operation. If
 you choose NoReboot and patches are installed, the instance is marked as non-
compliant until a subsequent reboot and scan.'
    allowedValues:
      - NoReboot
      - RebootIfNeeded
    default: RebootIfNeeded
  Operation:
```

```
   type: String
   description: '(Optional) The update or configuration to perform on the
instance. The system checks if patches specified in the patch baseline are
installed on the instance. The install operation installs patches missing from
the baseline.'
   allowedValues:
     - Install
     - Scan
   default: Install
```

JSON

```
{
  "schemaVersion":"0.3",
  "description":"An example of an Automation runbook that patches groups of
Amazon EC2 instances in stages.",
  "assumeRole":"{{AutomationAssumeRole}}",
  "parameters":{
     "AutomationAssumeRole":{
        "type":"String",
        "description":"(Optional) The Amazon Resource Name (ARN) of the IAM role
that allows Automation to perform the actions on your behalf. If no role is
specified, Systems Manager Automation uses your IAM permissions to operate this
runbook.",
        "default":""
     },
     "InstanceId":{
        "type":"String",
        "description":"(Required) The instance you want to patch."
     },
     "SnapshotId":{
        "type":"String",
        "description":"(Optional) The snapshot ID to use to retrieve a patch
baseline snapshot.",
        "default":""
     },
     "RebootOption":{
        "type":"String",
        "description":"(Optional) Reboot behavior after a patch Install
operation. If you choose NoReboot and patches are installed, the instance is
marked as non-compliant until a subsequent reboot and scan.",
        "allowedValues":[
           "NoReboot",
```

```
            "RebootIfNeeded"
         ],
         "default":"RebootIfNeeded"
      },
      "Operation":{
         "type":"String",
         "description":"(Optional) The update or configuration to perform on
 the instance. The system checks if patches specified in the patch baseline are
 installed on the instance. The install operation installs patches missing from
 the baseline.",
         "allowedValues":[
            "Install",
            "Scan"
         ],
         "default":"Install"
      }
   }
},
```

2. Com os elementos de nível superior definidos, Emily prossegue com a criação das ações que compõem as `mainSteps` do runbook. A primeira etapa gera o estado atual da instância de destino especificada no parâmetro de entrada `InstanceId` usando a ação `aws:executeAwsApi`. A saída desta ação é usada em ações posteriores.

   YAML

```
mainSteps:
  - name: getInstanceState
    action: 'aws:executeAwsApi'
    onFailure: Abort
    inputs:
      inputs:
      Service: ec2
      Api: DescribeInstances
      InstanceIds:
        - '{{InstanceId}}'
    outputs:
      - Name: instanceState
        Selector: '$.Reservations[0].Instances[0].State.Name'
        Type: String
    nextStep: branchOnInstanceState
```

JSON

```
"mainSteps":[
      {
         "name":"getInstanceState",
         "action":"aws:executeAwsApi",
         "onFailure":"Abort",
         "inputs":{
            "inputs":null,
            "Service":"ec2",
            "Api":"DescribeInstances",
            "InstanceIds":[
               "{{InstanceId}}"
            ]
         },
         "outputs":[
            {
               "Name":"instanceState",
               "Selector":"$.Reservations[0].Instances[0].State.Name",
               "Type":"String"
            }
         ],
         "nextStep":"branchOnInstanceState"
      },
```

3. Em vez de iniciar manualmente e manter o controle do estado original de cada instância que precisa ser corrigida, Emily usa a saída da ação anterior para ramificar a automação com base no estado da instância de destino. Isso permite que a automação execute etapas diferentes dependendo das condições definidas na ação `aws:branch` e melhora a eficiência geral da automação sem intervenção manual.

   Se o estado da instância já for `running`, a automação prossegue com o patch da instância com o documento `AWS-RunPatchBaseline` usando a ação `aws:runCommand`.

   Se o estado da instância for `stopping`, as pesquisas de automação sondam para a instância alcançar o estado `stopped`, usando a `aws:waitForAwsResourceProperty`, inicia a instância usando a ação `executeAwsApi`, e sonda a instância para alcançar um estado `running` antes de aplicar os patches nela.

Se o estado da instância for `stopped`, as pesquisas de automação iniciam a instância e a sonda para alcançar o estado `running`, antes de aplicar os patches na instância, usando as mesmas ações.

YAML

```
- name: branchOnInstanceState
    action: 'aws:branch'
    onFailure: Abort
    inputs:
      Choices:
        - NextStep: startInstance
          Variable: '{{getInstanceState.instanceState}}'
          StringEquals: stopped
         - NextStep: verifyInstanceStopped
          Variable: '{{getInstanceState.instanceState}}'
          StringEquals: stopping
         - NextStep: patchInstance
          Variable: '{{getInstanceState.instanceState}}'
          StringEquals: running
    isEnd: true
  - name: startInstance
    action: 'aws:executeAwsApi'
    onFailure: Abort
    inputs:
      Service: ec2
      Api: StartInstances
      InstanceIds:
        - '{{InstanceId}}'
    nextStep: verifyInstanceRunning
  - name: verifyInstanceRunning
    action: 'aws:waitForAwsResourceProperty'
    timeoutSeconds: 120
    inputs:
      Service: ec2
      Api: DescribeInstances
      InstanceIds:
        - '{{InstanceId}}'
      PropertySelector: '$.Reservations[0].Instances[0].State.Name'
      DesiredValues:
        - running
    nextStep: patchInstance
  - name: verifyInstanceStopped
```

```
    action: 'aws:waitForAwsResourceProperty'
    timeoutSeconds: 120
    inputs:
      Service: ec2
      Api: DescribeInstances
      InstanceIds:
        - '{{InstanceId}}'
      PropertySelector: '$.Reservations[0].Instances[0].State.Name'
      DesiredValues:
        - stopped
      nextStep: startInstance
  - name: patchInstance
    action: 'aws:runCommand'
    onFailure: Abort
    timeoutSeconds: 5400
    inputs:
      DocumentName: 'AWS-RunPatchBaseline'
      InstanceIds:
      - '{{InstanceId}}'
      Parameters:
        SnapshotId: '{{SnapshotId}}'
        RebootOption: '{{RebootOption}}'
        Operation: '{{Operation}}'
```

JSON

```
{
        "name":"branchOnInstanceState",
        "action":"aws:branch",
        "onFailure":"Abort",
        "inputs":{
           "Choices":[
              {
                 "NextStep":"startInstance",
                 "Variable":"{{getInstanceState.instanceState}}",
                 "StringEquals":"stopped"
              },
              {
                 "Or":[
                    {
                       "Variable":"{{getInstanceState.instanceState}}",
                       "StringEquals":"stopping"
                    }
```

```
                ],
                "NextStep":"verifyInstanceStopped"
             },
             {
                "NextStep":"patchInstance",
                "Variable":"{{getInstanceState.instanceState}}",
                "StringEquals":"running"
             }
          ]
       },
       "isEnd":true
    },
    {
       "name":"startInstance",
       "action":"aws:executeAwsApi",
       "onFailure":"Abort",
       "inputs":{
          "Service":"ec2",
          "Api":"StartInstances",
          "InstanceIds":[
             "{{InstanceId}}"
          ]
       },
       "nextStep":"verifyInstanceRunning"
    },
    {
       "name":"verifyInstanceRunning",
       "action":"aws:waitForAwsResourceProperty",
       "timeoutSeconds":120,
       "inputs":{
          "Service":"ec2",
          "Api":"DescribeInstances",
          "InstanceIds":[
             "{{InstanceId}}"
          ],
          "PropertySelector":"$.Reservations[0].Instances[0].State.Name",
          "DesiredValues":[
             "running"
          ]
       },
       "nextStep":"patchInstance"
    },
    {
       "name":"verifyInstanceStopped",
```

```
         "action":"aws:waitForAwsResourceProperty",
         "timeoutSeconds":120,
         "inputs":{
            "Service":"ec2",
            "Api":"DescribeInstances",
            "InstanceIds":[
               "{{InstanceId}}"
            ],
            "PropertySelector":"$.Reservations[0].Instances[0].State.Name",
            "DesiredValues":[
               "stopped"
            ],
            "nextStep":"startInstance"
         }
      },
      {
         "name":"patchInstance",
         "action":"aws:runCommand",
         "onFailure":"Abort",
         "timeoutSeconds":5400,
         "inputs":{
            "DocumentName":"AWS-RunPatchBaseline",
            "InstanceIds":[
               "{{InstanceId}}"
            ],
            "Parameters":{
               "SnapshotId":"{{SnapshotId}}",
               "RebootOption":"{{RebootOption}}",
               "Operation":"{{Operation}}"
            }
         }
      },
```

4. Após a conclusão da operação de patch, Emily deseja que a automação retorne a instância de destino ao mesmo estado em que estava antes da automação ser iniciada. Ela faz isso usando novamente a saída da primeira ação. As ramificações de automação com base no estado original da instância de destino usando a ação `aws:branch`. Se a instância estava anteriormente em outro estado diferente de `running`, ela será interrompida. Caso contrário, se o estado da instância for `running`, a automação é encerrada.

   YAML

```
- name: branchOnOriginalInstanceState
```

```
    action: 'aws:branch'
    onFailure: Abort
    inputs:
      Choices:
        - NextStep: stopInstance
          Not:
            Variable: '{{getInstanceState.instanceState}}'
            StringEquals: running
    isEnd: true
  - name: stopInstance
    action: 'aws:executeAwsApi'
    onFailure: Abort
    inputs:
      Service: ec2
      Api: StopInstances
      InstanceIds:
        - '{{InstanceId}}'
```

JSON

```
{
        "name":"branchOnOriginalInstanceState",
        "action":"aws:branch",
        "onFailure":"Abort",
        "inputs":{
           "Choices":[
              {
                 "NextStep":"stopInstance",
                 "Not":{
                    "Variable":"{{getInstanceState.instanceState}}",
                    "StringEquals":"running"
                 }
              }
           ]
        },
        "isEnd":true
     },
     {
        "name":"stopInstance",
        "action":"aws:executeAwsApi",
        "onFailure":"Abort",
        "inputs":{
           "Service":"ec2",
```

```
            "Api":"StopInstances",
            "InstanceIds":[
               "{{InstanceId}}"
            ]
         }
      }
   ]
}
```

5. Emily analisa o conteúdo do runbook filho concluído e cria o runbook na mesma Conta da AWS e Região da AWS das instâncias de destino. Agora ela está pronta para continuar com a criação do conteúdo do runbook pai. Abaixo está o conteúdo do runbook filho concluído.

   YAML

```
schemaVersion: '0.3'
description: 'An example of an Automation runbook that patches groups of Amazon
 EC2 instances in stages.'
assumeRole: '{{AutomationAssumeRole}}'
parameters:
  AutomationAssumeRole:
    type: String
    description: >-
      '(Optional) The Amazon Resource Name (ARN) of the IAM role that allows
 Automation to perform the
      actions on your behalf. If no role is specified, Systems Manager
      Automation uses your IAM permissions to operate this runbook.'
    default: ''
  InstanceId:
    type: String
    description: >-
      '(Required) The instance you want to patch.'
  SnapshotId:
    type: String
    description: '(Optional) The snapshot ID to use to retrieve a patch baseline
 snapshot.'
    default: ''
  RebootOption:
    type: String
    description: '(Optional) Reboot behavior after a patch Install operation. If
 you choose NoReboot and patches are installed, the instance is marked as non-
compliant until a subsequent reboot and scan.'
    allowedValues:
      - NoReboot
```

```
      - RebootIfNeeded
    default: RebootIfNeeded
  Operation:
    type: String
    description: '(Optional) The update or configuration to perform on the
 instance. The system checks if patches specified in the patch baseline are
 installed on the instance. The install operation installs patches missing from
 the baseline.'
    allowedValues:
      - Install
      - Scan
    default: Install
mainSteps:
  - name: getInstanceState
    action: 'aws:executeAwsApi'
    onFailure: Abort
    inputs:
      inputs:
      Service: ec2
      Api: DescribeInstances
      InstanceIds:
        - '{{InstanceId}}'
    outputs:
      - Name: instanceState
        Selector: '$.Reservations[0].Instances[0].State.Name'
        Type: String
    nextStep: branchOnInstanceState
  - name: branchOnInstanceState
    action: 'aws:branch'
    onFailure: Abort
    inputs:
      Choices:
        - NextStep: startInstance
          Variable: '{{getInstanceState.instanceState}}'
          StringEquals: stopped
        - Or:
            - Variable: '{{getInstanceState.instanceState}}'
              StringEquals: stopping
          NextStep: verifyInstanceStopped
        - NextStep: patchInstance
          Variable: '{{getInstanceState.instanceState}}'
          StringEquals: running
    isEnd: true
  - name: startInstance
```

```
    action: 'aws:executeAwsApi'
    onFailure: Abort
    inputs:
      Service: ec2
      Api: StartInstances
      InstanceIds:
        - '{{InstanceId}}'
    nextStep: verifyInstanceRunning
  - name: verifyInstanceRunning
    action: 'aws:waitForAwsResourceProperty'
    timeoutSeconds: 120
    inputs:
      Service: ec2
      Api: DescribeInstances
      InstanceIds:
        - '{{InstanceId}}'
      PropertySelector: '$.Reservations[0].Instances[0].State.Name'
      DesiredValues:
        - running
    nextStep: patchInstance
  - name: verifyInstanceStopped
    action: 'aws:waitForAwsResourceProperty'
    timeoutSeconds: 120
    inputs:
      Service: ec2
      Api: DescribeInstances
      InstanceIds:
        - '{{InstanceId}}'
      PropertySelector: '$.Reservations[0].Instances[0].State.Name'
      DesiredValues:
        - stopped
      nextStep: startInstance
  - name: patchInstance
    action: 'aws:runCommand'
    onFailure: Abort
    timeoutSeconds: 5400
    inputs:
      DocumentName: 'AWS-RunPatchBaseline'
      InstanceIds:
      - '{{InstanceId}}'
      Parameters:
        SnapshotId: '{{SnapshotId}}'
        RebootOption: '{{RebootOption}}'
        Operation: '{{Operation}}'
```

```
  - name: branchOnOriginalInstanceState
    action: 'aws:branch'
    onFailure: Abort
    inputs:
      Choices:
        - NextStep: stopInstance
          Not:
            Variable: '{{getInstanceState.instanceState}}'
            StringEquals: running
    isEnd: true
  - name: stopInstance
    action: 'aws:executeAwsApi'
    onFailure: Abort
    inputs:
      Service: ec2
      Api: StopInstances
      InstanceIds:
        - '{{InstanceId}}'
```

JSON

```
{
   "schemaVersion":"0.3",
   "description":"An example of an Automation runbook that patches groups of
 Amazon EC2 instances in stages.",
   "assumeRole":"{{AutomationAssumeRole}}",
   "parameters":{
      "AutomationAssumeRole":{
         "type":"String",
         "description":"'(Optional) The Amazon Resource Name (ARN) of the IAM
 role that allows Automation to perform the actions on your behalf. If no role is
 specified, Systems Manager Automation uses your IAM permissions to operate this
 runbook.'",
         "default":""
      },
      "InstanceId":{
         "type":"String",
         "description":"'(Required) The instance you want to patch.'"
      },
      "SnapshotId":{
         "type":"String",
         "description":"(Optional) The snapshot ID to use to retrieve a patch
 baseline snapshot.",
```

```
        "default":""
     },
     "RebootOption":{
        "type":"String",
        "description":"(Optional) Reboot behavior after a patch Install
operation. If you choose NoReboot and patches are installed, the instance is
marked as non-compliant until a subsequent reboot and scan.",
        "allowedValues":[
           "NoReboot",
           "RebootIfNeeded"
        ],
        "default":"RebootIfNeeded"
     },
     "Operation":{
        "type":"String",
        "description":"(Optional) The update or configuration to perform on
the instance. The system checks if patches specified in the patch baseline are
installed on the instance. The install operation installs patches missing from
the baseline.",
        "allowedValues":[
           "Install",
           "Scan"
        ],
        "default":"Install"
     }
  },
  "mainSteps":[
     {
        "name":"getInstanceState",
        "action":"aws:executeAwsApi",
        "onFailure":"Abort",
        "inputs":{
           "inputs":null,
           "Service":"ec2",
           "Api":"DescribeInstances",
           "InstanceIds":[
              "{{InstanceId}}"
           ]
        },
        "outputs":[
           {
              "Name":"instanceState",
              "Selector":"$.Reservations[0].Instances[0].State.Name",
              "Type":"String"
```

```
            }
         ],
         "nextStep":"branchOnInstanceState"
      },
      {
         "name":"branchOnInstanceState",
         "action":"aws:branch",
         "onFailure":"Abort",
         "inputs":{
            "Choices":[
               {
                  "NextStep":"startInstance",
                  "Variable":"{{getInstanceState.instanceState}}",
                  "StringEquals":"stopped"
               },
               {
                  "Or":[
                     {
                        "Variable":"{{getInstanceState.instanceState}}",
                        "StringEquals":"stopping"
                     }
                  ],
                  "NextStep":"verifyInstanceStopped"
               },
               {
                  "NextStep":"patchInstance",
                  "Variable":"{{getInstanceState.instanceState}}",
                  "StringEquals":"running"
               }
            ]
         },
         "isEnd":true
      },
      {
         "name":"startInstance",
         "action":"aws:executeAwsApi",
         "onFailure":"Abort",
         "inputs":{
            "Service":"ec2",
            "Api":"StartInstances",
            "InstanceIds":[
               "{{InstanceId}}"
            ]
         },
```

```
            "nextStep":"verifyInstanceRunning"
         },
         {
            "name":"verifyInstanceRunning",
            "action":"aws:waitForAwsResourceProperty",
            "timeoutSeconds":120,
            "inputs":{
               "Service":"ec2",
               "Api":"DescribeInstances",
               "InstanceIds":[
                  "{{InstanceId}}"
               ],
               "PropertySelector":"$.Reservations[0].Instances[0].State.Name",
               "DesiredValues":[
                  "running"
               ]
            },
            "nextStep":"patchInstance"
         },
         {
            "name":"verifyInstanceStopped",
            "action":"aws:waitForAwsResourceProperty",
            "timeoutSeconds":120,
            "inputs":{
               "Service":"ec2",
               "Api":"DescribeInstances",
               "InstanceIds":[
                  "{{InstanceId}}"
               ],
               "PropertySelector":"$.Reservations[0].Instances[0].State.Name",
               "DesiredValues":[
                  "stopped"
               ],
               "nextStep":"startInstance"
            }
         },
         {
            "name":"patchInstance",
            "action":"aws:runCommand",
            "onFailure":"Abort",
            "timeoutSeconds":5400,
            "inputs":{
               "DocumentName":"AWS-RunPatchBaseline",
               "InstanceIds":[
```

```
                "{{InstanceId}}"
            ],
            "Parameters":{
               "SnapshotId":"{{SnapshotId}}",
               "RebootOption":"{{RebootOption}}",
               "Operation":"{{Operation}}"
            }
         }
      },
      {
         "name":"branchOnOriginalInstanceState",
         "action":"aws:branch",
         "onFailure":"Abort",
         "inputs":{
            "Choices":[
               {
                  "NextStep":"stopInstance",
                  "Not":{
                     "Variable":"{{getInstanceState.instanceState}}",
                     "StringEquals":"running"
                  }
               }
            ]
         },
         "isEnd":true
      },
      {
         "name":"stopInstance",
         "action":"aws:executeAwsApi",
         "onFailure":"Abort",
         "inputs":{
            "Service":"ec2",
            "Api":"StopInstances",
            "InstanceIds":[
               "{{InstanceId}}"
            ]
         }
      }
   ]
}
```

Para obter mais informações sobre as ações de automação usadas neste exemplo, consulte a Referência de ações do Systems Manager Automation.

Crie o runbook pai

Esse exemplo de runbook continua o cenário descrito na seção anterior. Agora que Emily criou o runbook filho, ela começa a criar o conteúdo para o runbook pai da seguinte maneira:

1. Primeiro, ela fornece valores para o esquema e a descrição do runbook e define os parâmetros de entrada para o runbook pai.

   Utilizando o parâmetro `AutomationAssumeRole`, Emily e seus colegas podem usar uma função do IAM existente que permite que o Automation realize ações no runbook em seu nome. Emily usa os parâmetros `PatchGroupPrimaryKey` e `PatchGroupPrimaryValue` para especificar a tag associada ao grupo primário de instâncias de banco de dados que será corrigido. Ela usa os parâmetros `PatchGroupSecondaryKey` e `PatchGroupSecondaryValue` para especificar as tags associadas ao grupo secundário de instâncias de banco de dados nos quais os patches serão aplicados.

   YAML

```
description: 'An example of an Automation runbook that patches groups of Amazon
 EC2 instances in stages.'
schemaVersion: '0.3'
assumeRole: '{{AutomationAssumeRole}}'
parameters:
  AutomationAssumeRole:
    type: String
    description: '(Optional) The Amazon Resource Name (ARN) of the IAM role that
 allows Automation to perform the actions on your behalf. If no role is specified,
 Systems Manager Automation uses your IAM permissions to operate this runbook.'
    default: ''
  PatchGroupPrimaryKey:
    type: String
    description: '(Required) The key of the tag for the primary group of instances
 you want to patch.''
  PatchGroupPrimaryValue:
    type: String
    description: '(Required) The value of the tag for the primary group of
 instances you want to patch.'
  PatchGroupSecondaryKey:
    type: String
```

```
    description: '(Required) The key of the tag for the secondary group of
 instances you want to patch.'
  PatchGroupSecondaryValue:
    type: String
    description: '(Required) The value of the tag for the secondary group of
 instances you want to patch.'
```

JSON

```
{
   "schemaVersion": "0.3",
   "description": "An example of an Automation runbook that patches groups of
 Amazon EC2 instances in stages.",
   "assumeRole": "{{AutomationAssumeRole}}",
   "parameters": {
      "AutomationAssumeRole": {
         "type": "String",
         "description": "(Optional) The Amazon Resource Name (ARN) of the IAM
 role that allows Automation to perform the actions on your behalf. If no role is
 specified, Systems Manager Automation uses your IAM permissions to operate this
 runbook.",
         "default": ""
      },
      "PatchGroupPrimaryKey": {
         "type": "String",
         "description": "(Required) The key of the tag for the primary group of
 instances you want to patch."
      },
      "PatchGroupPrimaryValue": {
         "type": "String",
         "description": "(Required) The value of the tag for the primary group of
 instances you want to patch."
      },
      "PatchGroupSecondaryKey": {
         "type": "String",
         "description": "(Required) The key of the tag for the secondary group of
 instances you want to patch."
      },
      "PatchGroupSecondaryValue": {
         "type": "String",
         "description": "(Required) The value of the tag for the secondary group
 of instances you want to patch."
      }
```

```
    }
},
```

2. Com os elementos de nível superior definidos, Emily prossegue com a criação das ações que compõem as `mainSteps` do runbook.

   A primeira ação inicia uma automação de controle de taxa usando o runbook filho recém-criado, direcionado a instâncias associadas à tag especificada no `PatchGroupPrimaryKey` e nos parâmetros de entrada `PatchGroupPrimaryValue`. Ela usa os valores fornecidos aos parâmetros de entrada para especificar a chave e o valor da tag associada ao grupo primário de instâncias de banco de dados nos quais deseja aplicar o patch.

   Depois que a primeira automação for concluída, a segunda ação inicia outra automação de controle de taxa usando o runbook filho direcionado a instâncias associadas à tag especificada no `PatchGroupSecondaryKey` e no parâmetros de entrada do `PatchGroupSecondaryValue`. Ela usa os valores fornecidos aos parâmetros de entrada para especificar a chave e o valor da tag associada ao grupo secundário de instâncias de banco de dados nos quais deseja aplicar o patch.

   YAML

```
mainSteps:
  - name: patchPrimaryTargets
    action: 'aws:executeAutomation'
    onFailure: Abort
    timeoutSeconds: 7200
    inputs:
      DocumentName: RunbookTutorialChildAutomation
      Targets:
        - Key: 'tag:{{PatchGroupPrimaryKey}}'
          Values:
            - '{{PatchGroupPrimaryValue}}'
      TargetParameterName: 'InstanceId'
  - name: patchSecondaryTargets
    action: 'aws:executeAutomation'
    onFailure: Abort
    timeoutSeconds: 7200
    inputs:
      DocumentName: RunbookTutorialChildAutomation
      Targets:
        - Key: 'tag:{{PatchGroupSecondaryKey}}'
          Values:
            - '{{PatchGroupSecondaryValue}}'
```

```
      TargetParameterName: 'InstanceId'
```

JSON

```
"mainSteps":[
      {
         "name":"patchPrimaryTargets",
         "action":"aws:executeAutomation",
         "onFailure":"Abort",
         "timeoutSeconds":7200,
         "inputs":{
            "DocumentName":"RunbookTutorialChildAutomation",
            "Targets":[
               {
                  "Key":"tag:{{PatchGroupPrimaryKey}}",
                  "Values":[
                     "{{PatchGroupPrimaryValue}}"
                  ]
               }
            ],
            "TargetParameterName":"InstanceId"
         }
      },
      {
         "name":"patchSecondaryTargets",
         "action":"aws:executeAutomation",
         "onFailure":"Abort",
         "timeoutSeconds":7200,
         "inputs":{
            "DocumentName":"RunbookTutorialChildAutomation",
            "Targets":[
               {
                  "Key":"tag:{{PatchGroupSecondaryKey}}",
                  "Values":[
                     "{{PatchGroupSecondaryValue}}"
                  ]
               }
            ],
            "TargetParameterName":"InstanceId"
         }
      }
   ]
}
```

3. Emily analisa o conteúdo do runbook pai concluído e cria o runbook na mesma Conta da AWS e Região da AWS das instâncias de destino. Agora ela está pronta para testar seus runbooks para garantir que a automação funcione conforme desejado antes de implementá-la em seu ambiente de produção. Abaixo está o conteúdo do runbook pai concluído.

   YAML

```
description: An example of an Automation runbook that patches groups of Amazon EC2
 instances in stages.
schemaVersion: '0.3'
assumeRole: '{{AutomationAssumeRole}}'
parameters:
  AutomationAssumeRole:
    type: String
    description: '(Optional) The Amazon Resource Name (ARN) of the IAM role that
 allows Automation to perform the actions on your behalf. If no role is specified,
 Systems Manager Automation uses your IAM permissions to operate this runbook.'
    default: ''
  PatchGroupPrimaryKey:
    type: String
    description: (Required) The key of the tag for the primary group of instances
 you want to patch.
  PatchGroupPrimaryValue:
    type: String
    description: '(Required) The value of the tag for the primary group of
 instances you want to patch. '
  PatchGroupSecondaryKey:
    type: String
    description: (Required) The key of the tag for the secondary group of
 instances you want to patch.
  PatchGroupSecondaryValue:
    type: String
    description: '(Required) The value of the tag for the secondary group of
 instances you want to patch.  '
mainSteps:
  - name: patchPrimaryTargets
    action: 'aws:executeAutomation'
    onFailure: Abort
    timeoutSeconds: 7200
    inputs:
      DocumentName: RunbookTutorialChildAutomation
      Targets:
        - Key: 'tag:{{PatchGroupPrimaryKey}}'
```

```
          Values:
            - '{{PatchGroupPrimaryValue}}'
      TargetParameterName: 'InstanceId'
  - name: patchSecondaryTargets
    action: 'aws:executeAutomation'
    onFailure: Abort
    timeoutSeconds: 7200
    inputs:
      DocumentName: RunbookTutorialChildAutomation
      Targets:
        - Key: 'tag:{{PatchGroupSecondaryKey}}'
          Values:
            - '{{PatchGroupSecondaryValue}}'
      TargetParameterName: 'InstanceId'
```

JSON

```
{
   "description":"An example of an Automation runbook that patches groups of
 Amazon EC2 instances in stages.",
   "schemaVersion":"0.3",
   "assumeRole":"{{AutomationAssumeRole}}",
   "parameters":{
      "AutomationAssumeRole":{
         "type":"String",
         "description":"(Optional) The Amazon Resource Name (ARN) of the IAM role
 that allows Automation to perform the actions on your behalf. If no role is
 specified, Systems Manager Automation uses your IAM permissions to operate this
 runbook.",
         "default":""
      },
      "PatchGroupPrimaryKey":{
         "type":"String",
         "description":"(Required) The key of the tag for the primary group of
 instances you want to patch."
      },
      "PatchGroupPrimaryValue":{
         "type":"String",
         "description":"(Required) The value of the tag for the primary group of
 instances you want to patch. "
      },
      "PatchGroupSecondaryKey":{
         "type":"String",
```

```
        "description":"(Required) The key of the tag for the secondary group of
instances you want to patch."
     },
     "PatchGroupSecondaryValue":{
        "type":"String",
        "description":"(Required) The value of the tag for the secondary group of
instances you want to patch.  "
     }
  },
  "mainSteps":[
     {
        "name":"patchPrimaryTargets",
        "action":"aws:executeAutomation",
        "onFailure":"Abort",
        "timeoutSeconds":7200,
        "inputs":{
           "DocumentName":"RunbookTutorialChildAutomation",
           "Targets":[
              {
                 "Key":"tag:{{PatchGroupPrimaryKey}}",
                 "Values":[
                    "{{PatchGroupPrimaryValue}}"
                 ]
              }
           ],
           "TargetParameterName":"InstanceId"
        }
     },
     {
        "name":"patchSecondaryTargets",
        "action":"aws:executeAutomation",
        "onFailure":"Abort",
        "timeoutSeconds":7200,
        "inputs":{
           "DocumentName":"RunbookTutorialChildAutomation",
           "Targets":[
              {
                 "Key":"tag:{{PatchGroupSecondaryKey}}",
                 "Values":[
                    "{{PatchGroupSecondaryValue}}"
                 ]
              }
           ],
           "TargetParameterName":"InstanceId"
```

```
            }
        }
    ]
}
```

Para obter mais informações sobre as ações de automação usadas neste exemplo, consulte a Referência de ações do Systems Manager Automation.

Exemplo 2: Runbook com script

Este exemplo de runbook aborda o seguinte cenário. Emily é engenheira de sistemas na AnyCompany Consultants, LLC. Ela criou anteriormente dois runbooks que são usados em um relacionamento pai-filho para grupos de patches de instâncias do Amazon Elastic Compute Cloud (Amazon EC2) que hospedam bancos de dados primários e secundários. As aplicações acessam esses bancos de dados 24 horas por dia, portanto, uma das instâncias do banco de dados deve estar sempre disponível.

Com base nesse requisito, ela criou uma solução que corrige as instâncias em etapas usando o documento do Systems Manager (SSM) `AWS-RunPatchBaseline`. Usando este documento do SSM, seus colegas podem revisar as informações de conformidade de patch associadas após a conclusão da operação de patch.

O grupo principal de instâncias de banco de dados recebem o patch primeiro, seguido pelo grupo secundário de instâncias de banco de dados. Além disso, para evitar custos adicionais, deixando instâncias em execução que foram interrompidas anteriormente, Emily garantiu que a automação retornasse as instâncias corrigidas ao seu estado original antes da aplicação de patches ocorrer. Emily usou tags associadas aos grupos primário e secundário de instâncias de banco de dados para identificar quais instâncias devem ser corrigidas na ordem desejada.

A solução automatizada existente funciona, mas ela quer melhorar a solução, se possível. Para ajudar na manutenção do conteúdo do runbook e facilitar a solução de problemas, ela gostaria de condensar a automação em um único runbook e simplificar o número de parâmetros de entrada. Além disso, ela gostaria de evitar a criação de várias automações filho.

Depois que Emily revisar as ações de automação disponíveis, ela determinará que pode melhorar sua solução usando o `aws:executeScript` para executar seus scripts Python personalizados. Ela agora começa a criar o conteúdo para o runbook da seguinte forma:

1. Primeiro, ela fornece valores para o esquema e a descrição do runbook e define os parâmetros de entrada para o runbook pai.

   Utilizando o parâmetro `AutomationAssumeRole`, Emily e seus colegas podem usar uma função do IAM existente que permite que o Automation realize ações no runbook em seu nome. Ao contrário do Exemplo 1, o parâmetro `AutomationAssumeRole` agora é necessário e não opcional. Como este runbook inclui ações `aws:executeScript`, uma função de serviço (IAM) AWS Identity and Access Management (ou função assumida) será sempre necessária. Este requisito é necessário porque alguns dos scripts Python especificados para as ações chamam as operações de API da AWS.

   Emily usa os parâmetros `PrimaryPatchGroupTag` e `SecondaryPatchGroupTag` para especificar as tags associadas ao grupo secundário de instâncias de banco de dados que será corrigido. Para simplificar os parâmetros de entrada necessários, ela decide usar os parâmetros `StringMap` em vez de usar vários parâmetros `String` como ela usou no runbook de Exemplo 1. Opcionalmente, os parâmetros `Operation`, `RebootOption` e `SnapshotId` podem ser usados para fornecer valores para documentar os parâmetros para `AWS-RunPatchBaseline`. Para evitar que valores inválidos sejam fornecidos a esses parâmetros do documento, ela define a propriedade `allowedValues` conforme necessário.

   YAML

```
description: 'An example of an Automation runbook that patches groups of Amazon
 EC2 instances in stages.'
schemaVersion: '0.3'
assumeRole: '{{AutomationAssumeRole}}'
parameters:
  AutomationAssumeRole:
    type: String
    description: '(Required) The Amazon Resource Name (ARN) of the IAM role that
 allows Automation to perform the actions on your behalf. If no role is specified,
 Systems Manager Automation uses your IAM permissions to operate this runbook.'
  PrimaryPatchGroupTag:
    type: StringMap
    description: '(Required) The tag for the primary group of instances you want
 to patch. Specify a key-value pair. Example: {"key" : "value"}'
  SecondaryPatchGroupTag:
    type: StringMap
    description: '(Required) The tag for the secondary group of instances you want
 to patch. Specify a key-value pair. Example: {"key" : "value"}'
  SnapshotId:
```

```
    type: String
    description: '(Optional) The snapshot ID to use to retrieve a patch baseline
 snapshot.'
    default: ''
  RebootOption:
    type: String
    description: '(Optional) Reboot behavior after a patch Install operation. If
 you choose NoReboot and patches are installed, the instance is marked as non-
compliant until a subsequent reboot and scan.'
    allowedValues:
      - NoReboot
      - RebootIfNeeded
    default: RebootIfNeeded
  Operation:
    type: String
    description: '(Optional) The update or configuration to perform on the
 instance. The system checks if patches specified in the patch baseline are
 installed on the instance. The install operation installs patches missing from
 the baseline.'
    allowedValues:
      - Install
      - Scan
    default: Install
```

JSON

```
{
   "description":"An example of an Automation runbook that patches groups of
 Amazon EC2 instances in stages.",
   "schemaVersion":"0.3",
   "assumeRole":"{{AutomationAssumeRole}}",
   "parameters":{
      "AutomationAssumeRole":{
         "type":"String",
         "description":"(Required) The Amazon Resource Name (ARN) of the IAM role
 that allows Automation to perform the actions on your behalf. If no role is
 specified, Systems Manager Automation uses your IAM permissions to operate this
 runbook."
      },
      "PrimaryPatchGroupTag":{
         "type":"StringMap",
         "description":"(Required) The tag for the primary group of instances you
 want to patch. Specify a key-value pair. Example: {\"key\" : \"value\"}"
```

```
      },
      "SecondaryPatchGroupTag":{
         "type":"StringMap",
         "description":"(Required) The tag for the secondary group of instances
 you want to patch. Specify a key-value pair. Example: {\"key\" : \"value\"}"
      },
      "SnapshotId":{
         "type":"String",
         "description":"(Optional) The snapshot ID to use to retrieve a patch
 baseline snapshot.",
         "default":""
      },
      "RebootOption":{
         "type":"String",
         "description":"(Optional) Reboot behavior after a patch Install
 operation. If you choose NoReboot and patches are installed, the instance is
 marked as non-compliant until a subsequent reboot and scan.",
         "allowedValues":[
            "NoReboot",
            "RebootIfNeeded"
         ],
         "default":"RebootIfNeeded"
      },
      "Operation":{
         "type":"String",
         "description":"(Optional) The update or configuration to perform on
 the instance. The system checks if patches specified in the patch baseline are
 installed on the instance. The install operation installs patches missing from
 the baseline.",
         "allowedValues":[
            "Install",
            "Scan"
         ],
         "default":"Install"
      }
   }
},
```

2. Com os elementos de nível superior definidos, Emily prossegue com a criação das ações que compõem as `mainSteps` do runbook. A primeira etapa reúne os IDs de todas as instâncias associadas à tag especificada no parâmetro `PrimaryPatchGroupTag` e resulta em um parâmetro `StringMap` que contém o ID de instância e o estado atual da instância. A saída desta ação é usada em ações posteriores.

Observe que o parâmetro de entrada do `script` é compatível com runbooks JSON. Os runbooks JSON devem fornecer conteúdo de script usando o parâmetro de entrada do `attachment`.

YAML

```
mainSteps:
  - name: getPrimaryInstanceState
    action: 'aws:executeScript'
    timeoutSeconds: 120
    onFailure: Abort
    inputs:
      Runtime: python3.7
      Handler: getInstanceStates
      InputPayload:
        primaryTag: '{{PrimaryPatchGroupTag}}'
      Script: |-
        def getInstanceStates(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          tag = events['primaryTag']
          tagKey, tagValue = list(tag.items())[0]
          instanceQuery = ec2.describe_instances(
          Filters=[
              {
                  "Name": "tag:" + tagKey,
                  "Values": [tagValue]
              }]
          )
          if not instanceQuery['Reservations']:
              noInstancesForTagString = "No instances found for specified tag."
              return({ 'noInstancesFound' : noInstancesForTagString })
          else:
              queryResponse = instanceQuery['Reservations']
              originalInstanceStates = {}
              for results in queryResponse:
                  instanceSet = results['Instances']
                  for instance in instanceSet:
                      instanceId = instance['InstanceId']
                      originalInstanceStates[instanceId] = instance['State']
['Name']
              return originalInstanceStates
```

```
    outputs:
      - Name: originalInstanceStates
        Selector: $.Payload
        Type: StringMap
    nextStep: verifyPrimaryInstancesRunning
```

JSON

```
"mainSteps":[
      {
         "name":"getPrimaryInstanceState",
         "action":"aws:executeScript",
         "timeoutSeconds":120,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.7",
            "Handler":"getInstanceStates",
            "InputPayload":{
               "primaryTag":"{{PrimaryPatchGroupTag}}"
            },
            "Script":"..."
         },
         "outputs":[
            {
               "Name":"originalInstanceStates",
               "Selector":"$.Payload",
               "Type":"StringMap"
            }
         ],
         "nextStep":"verifyPrimaryInstancesRunning"
      },
```

3. Emily usa o resultado da ação anterior em outra ação do `aws:executeScript` para verificar se todas as ocorrências associadas à tag especificada no parâmetro `PrimaryPatchGroupTag` estão em um estado `running`.

   Se o estado da instância já for `running` ou `shutting-down`, o script continua a percorrer as instâncias restantes.

   Se o estado da instância for `stopping`, o script sonda para que a instância atinja o estado `stopped` e inicie a instância.

Se o estado da instância for `stopped`, o script inicia a instância.

YAML

```
- name: verifyPrimaryInstancesRunning
    action: 'aws:executeScript'
    timeoutSeconds: 600
    onFailure: Abort
    inputs:
      Runtime: python3.7
      Handler: verifyInstancesRunning
      InputPayload:
        targetInstances: '{{getPrimaryInstanceState.originalInstanceStates}}'
      Script: |-
        def verifyInstancesRunning(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          instanceDict = events['targetInstances']
          for instance in instanceDict:
            if instanceDict[instance] == 'stopped':
                print("The target instance " + instance + " is stopped. The
 instance will now be started.")
                ec2.start_instances(
                    InstanceIds=[instance]
                    )
            elif instanceDict[instance] == 'stopping':
                print("The target instance " + instance + " is stopping. Polling
 for instance to reach stopped state.")
                while instanceDict[instance] != 'stopped':
                    poll = ec2.get_waiter('instance_stopped')
                    poll.wait(
                        InstanceIds=[instance]
                    )
                ec2.start_instances(
                    InstanceIds=[instance]
                )
            else:
              pass
    nextStep: waitForPrimaryRunningInstances
```

JSON

```
{
         "name":"verifyPrimaryInstancesRunning",
         "action":"aws:executeScript",
         "timeoutSeconds":600,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.7",
            "Handler":"verifyInstancesRunning",
            "InputPayload":{

 "targetInstances":"{{getPrimaryInstanceState.originalInstanceStates}}"
            },
            "Script":"..."
         },
         "nextStep":"waitForPrimaryRunningInstances"
      },
```

4. Emily verifica se todas as instâncias associadas com a tag especificada no parâmetro `PrimaryPatchGroupTag` foram iniciadas ou já estão em um estado `running`. Ele então usa outro script para verificar se todas as instâncias, incluindo aquelas que foram iniciadas na ação anterior, atingiram o estado `running`.

   YAML

```
- name: waitForPrimaryRunningInstances
    action: 'aws:executeScript'
    timeoutSeconds: 300
    onFailure: Abort
    inputs:
      Runtime: python3.7
      Handler: waitForRunningInstances
      InputPayload:
        targetInstances: '{{getPrimaryInstanceState.originalInstanceStates}}'
      Script: |-
        def waitForRunningInstances(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          instanceDict = events['targetInstances']
          for instance in instanceDict:
```

```
                poll = ec2.get_waiter('instance_running')
                poll.wait(
                    InstanceIds=[instance]
                )
    nextStep: returnPrimaryTagKey
```

JSON

```
{
         "name":"waitForPrimaryRunningInstances",
         "action":"aws:executeScript",
         "timeoutSeconds":300,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.7",
            "Handler":"waitForRunningInstances",
            "InputPayload":{

 "targetInstances":"{{getPrimaryInstanceState.originalInstanceStates}}"
            },
            "Script":"..."
         },
         "nextStep":"returnPrimaryTagKey"
      },
```

5. Emily usa mais dois scripts para retornar valores `String` individuais da chave e valor da tag especificada no parâmetro `PrimaryPatchGroupTag`. Os valores retornados por essas ações permitem que ela forneça valores diretamente ao parâmetro `Targets` do documento `AWS-RunPatchBaseline`. A automação prossegue então com a aplicação do patch na instância com o documento `AWS-RunPatchBaseline` usando a ação `aws:runCommand`.

   YAML

```
- name: returnPrimaryTagKey
    action: 'aws:executeScript'
    timeoutSeconds: 120
    onFailure: Abort
    inputs:
      Runtime: python3.7
      Handler: returnTagValues
      InputPayload:
        primaryTag: '{{PrimaryPatchGroupTag}}'
      Script: |-
```

```
        def returnTagValues(events,context):
          tag = events['primaryTag']
          tagKey = list(tag)[0]
          stringKey = "tag:" + tagKey
          return {'tagKey' : stringKey}
    outputs:
      - Name: Payload
        Selector: $.Payload
        Type: StringMap
      - Name: primaryPatchGroupKey
        Selector: $.Payload.tagKey
        Type: String
    nextStep: returnPrimaryTagValue
  - name: returnPrimaryTagValue
    action: 'aws:executeScript'
    timeoutSeconds: 120
    onFailure: Abort
    inputs:
      Runtime: python3.7
      Handler: returnTagValues
      InputPayload:
        primaryTag: '{{PrimaryPatchGroupTag}}'
      Script: |-
        def returnTagValues(events,context):
          tag = events['primaryTag']
          tagKey = list(tag)[0]
          tagValue = tag[tagKey]
          return {'tagValue' : tagValue}
    outputs:
      - Name: Payload
        Selector: $.Payload
        Type: StringMap
      - Name: primaryPatchGroupValue
        Selector: $.Payload.tagValue
        Type: String
    nextStep: patchPrimaryInstances
  - name: patchPrimaryInstances
    action: 'aws:runCommand'
    onFailure: Abort
    timeoutSeconds: 7200
    inputs:
      DocumentName: AWS-RunPatchBaseline
      Parameters:
        SnapshotId: '{{SnapshotId}}'
```

```
        RebootOption: '{{RebootOption}}'
        Operation: '{{Operation}}'
      Targets:
        - Key: '{{returnPrimaryTagKey.primaryPatchGroupKey}}'
          Values:
            - '{{returnPrimaryTagValue.primaryPatchGroupValue}}'
      MaxConcurrency: 10%
      MaxErrors: 10%
    nextStep: returnPrimaryToOriginalState
```

JSON

```
{
         "name":"returnPrimaryTagKey",
         "action":"aws:executeScript",
         "timeoutSeconds":120,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.7",
            "Handler":"returnTagValues",
            "InputPayload":{
               "primaryTag":"{{PrimaryPatchGroupTag}}"
            },
            "Script":"..."
         },
         "outputs":[
            {
               "Name":"Payload",
               "Selector":"$.Payload",
               "Type":"StringMap"
            },
            {
               "Name":"primaryPatchGroupKey",
               "Selector":"$.Payload.tagKey",
               "Type":"String"
            }
         ],
         "nextStep":"returnPrimaryTagValue"
      },
      {
         "name":"returnPrimaryTagValue",
         "action":"aws:executeScript",
         "timeoutSeconds":120,
```

```
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.7",
            "Handler":"returnTagValues",
            "InputPayload":{
               "primaryTag":"{{PrimaryPatchGroupTag}}"
            },
            "Script":"..."
         },
         "outputs":[
            {
               "Name":"Payload",
               "Selector":"$.Payload",
               "Type":"StringMap"
            },
            {
               "Name":"primaryPatchGroupValue",
               "Selector":"$.Payload.tagValue",
               "Type":"String"
            }
         ],
         "nextStep":"patchPrimaryInstances"
      },
      {
         "name":"patchPrimaryInstances",
         "action":"aws:runCommand",
         "onFailure":"Abort",
         "timeoutSeconds":7200,
         "inputs":{
            "DocumentName":"AWS-RunPatchBaseline",
            "Parameters":{
               "SnapshotId":"{{SnapshotId}}",
               "RebootOption":"{{RebootOption}}",
               "Operation":"{{Operation}}"
            },
            "Targets":[
               {
                  "Key":"{{returnPrimaryTagKey.primaryPatchGroupKey}}",
                  "Values":[
                     "{{returnPrimaryTagValue.primaryPatchGroupValue}}"
                  ]
               }
            ],
            "MaxConcurrency":"10%",
```

```
            "MaxErrors":"10%"
         },
         "nextStep":"returnPrimaryToOriginalState"
      },
```

6. Após a conclusão da operação de patch, Emily deseja que a automação retorne as instâncias de destino associadas à tag especificada no parâmetro `PrimaryPatchGroupTag` para o mesmo estado em que estavam antes da automação ser iniciada. Ela faz isso usando novamente a saída da primeira ação em um script. Com base no estado original da instância de destino, se a instância estava anteriormente em qualquer estado diferente de `running`, ela é interrompida. Caso contrário, se o estado da instância for `running` o script continuará a percorrer as instâncias restantes.

   YAML

```
- name: returnPrimaryToOriginalState
    action: 'aws:executeScript'
    timeoutSeconds: 600
    onFailure: Abort
    inputs:
      Runtime: python3.7
      Handler: returnToOriginalState
      InputPayload:
        targetInstances: '{{getPrimaryInstanceState.originalInstanceStates}}'
      Script: |-
        def returnToOriginalState(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          instanceDict = events['targetInstances']
          for instance in instanceDict:
            if instanceDict[instance] == 'stopped' or instanceDict[instance] ==
 'stopping':
                ec2.stop_instances(
                    InstanceIds=[instance]
                    )
            else:
              pass
    nextStep: getSecondaryInstanceState
```

JSON

```
{
         "name":"returnPrimaryToOriginalState",
         "action":"aws:executeScript",
         "timeoutSeconds":600,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.7",
            "Handler":"returnToOriginalState",
            "InputPayload":{

 "targetInstances":"{{getPrimaryInstanceState.originalInstanceStates}}"
            },
            "Script":"..."
         },
         "nextStep":"getSecondaryInstanceState"
      },
```

7. A operação de patch é concluída para as instâncias associadas à tag especificada no parâmetro `PrimaryPatchGroupTag`. Agora, Emily duplica todas as ações anteriores em seu conteúdo do runbook para direcionar as instâncias associadas à tag especificada no parâmetro `SecondaryPatchGroupTag`.

   YAML

```
- name: getSecondaryInstanceState
    action: 'aws:executeScript'
    timeoutSeconds: 120
    onFailure: Abort
    inputs:
      Runtime: python3.7
      Handler: getInstanceStates
      InputPayload:
        secondaryTag: '{{SecondaryPatchGroupTag}}'
      Script: |-
        def getInstanceStates(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          tag = events['secondaryTag']
          tagKey, tagValue = list(tag.items())[0]
```

```
          instanceQuery = ec2.describe_instances(
          Filters=[
              {
                  "Name": "tag:" + tagKey,
                  "Values": [tagValue]
              }]
          )
          if not instanceQuery['Reservations']:
              noInstancesForTagString = "No instances found for specified tag."
              return({ 'noInstancesFound' : noInstancesForTagString })
          else:
              queryResponse = instanceQuery['Reservations']
              originalInstanceStates = {}
              for results in queryResponse:
                  instanceSet = results['Instances']
                  for instance in instanceSet:
                      instanceId = instance['InstanceId']
                      originalInstanceStates[instanceId] = instance['State']
['Name']
              return originalInstanceStates
    outputs:
      - Name: originalInstanceStates
        Selector: $.Payload
        Type: StringMap
    nextStep: verifySecondaryInstancesRunning
  - name: verifySecondaryInstancesRunning
    action: 'aws:executeScript'
    timeoutSeconds: 600
    onFailure: Abort
    inputs:
      Runtime: python3.7
      Handler: verifyInstancesRunning
      InputPayload:
        targetInstances: '{{getSecondaryInstanceState.originalInstanceStates}}'
      Script: |-
        def verifyInstancesRunning(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          instanceDict = events['targetInstances']
          for instance in instanceDict:
            if instanceDict[instance] == 'stopped':
```

```
                print("The target instance " + instance + " is stopped. The
instance will now be started.")
                ec2.start_instances(
                    InstanceIds=[instance]
                    )
            elif instanceDict[instance] == 'stopping':
                print("The target instance " + instance + " is stopping. Polling
for instance to reach stopped state.")
                while instanceDict[instance] != 'stopped':
                    poll = ec2.get_waiter('instance_stopped')
                    poll.wait(
                        InstanceIds=[instance]
                    )
                ec2.start_instances(
                    InstanceIds=[instance]
                )
            else:
              pass
    nextStep: waitForSecondaryRunningInstances
  - name: waitForSecondaryRunningInstances
    action: 'aws:executeScript'
    timeoutSeconds: 300
    onFailure: Abort
    inputs:
      Runtime: python3.7
      Handler: waitForRunningInstances
      InputPayload:
        targetInstances: '{{getSecondaryInstanceState.originalInstanceStates}}'
      Script: |-
        def waitForRunningInstances(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          instanceDict = events['targetInstances']
          for instance in instanceDict:
              poll = ec2.get_waiter('instance_running')
              poll.wait(
                  InstanceIds=[instance]
              )
    nextStep: returnSecondaryTagKey
  - name: returnSecondaryTagKey
    action: 'aws:executeScript'
    timeoutSeconds: 120
```

```
    onFailure: Abort
    inputs:
      Runtime: python3.7
      Handler: returnTagValues
      InputPayload:
        secondaryTag: '{{SecondaryPatchGroupTag}}'
      Script: |-
        def returnTagValues(events,context):
          tag = events['secondaryTag']
          tagKey = list(tag)[0]
          stringKey = "tag:" + tagKey
          return {'tagKey' : stringKey}
    outputs:
      - Name: Payload
        Selector: $.Payload
        Type: StringMap
      - Name: secondaryPatchGroupKey
        Selector: $.Payload.tagKey
        Type: String
    nextStep: returnSecondaryTagValue
  - name: returnSecondaryTagValue
    action: 'aws:executeScript'
    timeoutSeconds: 120
    onFailure: Abort
    inputs:
      Runtime: python3.7
      Handler: returnTagValues
      InputPayload:
        secondaryTag: '{{SecondaryPatchGroupTag}}'
      Script: |-
        def returnTagValues(events,context):
          tag = events['secondaryTag']
          tagKey = list(tag)[0]
          tagValue = tag[tagKey]
          return {'tagValue' : tagValue}
    outputs:
      - Name: Payload
        Selector: $.Payload
        Type: StringMap
      - Name: secondaryPatchGroupValue
        Selector: $.Payload.tagValue
        Type: String
    nextStep: patchSecondaryInstances
  - name: patchSecondaryInstances
```

```
    action: 'aws:runCommand'
    onFailure: Abort
    timeoutSeconds: 7200
    inputs:
      DocumentName: AWS-RunPatchBaseline
      Parameters:
        SnapshotId: '{{SnapshotId}}'
        RebootOption: '{{RebootOption}}'
        Operation: '{{Operation}}'
      Targets:
        - Key: '{{returnSecondaryTagKey.secondaryPatchGroupKey}}'
          Values:
          - '{{returnSecondaryTagValue.secondaryPatchGroupValue}}'
      MaxConcurrency: 10%
      MaxErrors: 10%
    nextStep: returnSecondaryToOriginalState
  - name: returnSecondaryToOriginalState
    action: 'aws:executeScript'
    timeoutSeconds: 600
    onFailure: Abort
    inputs:
      Runtime: python3.7
      Handler: returnToOriginalState
      InputPayload:
        targetInstances: '{{getSecondaryInstanceState.originalInstanceStates}}'
      Script: |-
        def returnToOriginalState(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          instanceDict = events['targetInstances']
          for instance in instanceDict:
            if instanceDict[instance] == 'stopped' or instanceDict[instance] ==
'stopping':
                ec2.stop_instances(
                    InstanceIds=[instance]
                    )
            else:
              pass
```

JSON

```
{
         "name":"getSecondaryInstanceState",
         "action":"aws:executeScript",
         "timeoutSeconds":120,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.7",
            "Handler":"getInstanceStates",
            "InputPayload":{
               "secondaryTag":"{{SecondaryPatchGroupTag}}"
            },
            "Script":"..."
         },
         "outputs":[
            {
               "Name":"originalInstanceStates",
               "Selector":"$.Payload",
               "Type":"StringMap"
            }
         ],
         "nextStep":"verifySecondaryInstancesRunning"
      },
      {
         "name":"verifySecondaryInstancesRunning",
         "action":"aws:executeScript",
         "timeoutSeconds":600,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.7",
            "Handler":"verifyInstancesRunning",
            "InputPayload":{

 "targetInstances":"{{getSecondaryInstanceState.originalInstanceStates}}"
            },
            "Script":"..."
         },
         "nextStep":"waitForSecondaryRunningInstances"
      },
      {
         "name":"waitForSecondaryRunningInstances",
         "action":"aws:executeScript",
```

```
        "timeoutSeconds":300,
        "onFailure":"Abort",
        "inputs":{
           "Runtime":"python3.7",
           "Handler":"waitForRunningInstances",
           "InputPayload":{

"targetInstances":"{{getSecondaryInstanceState.originalInstanceStates}}"
           },
           "Script":"..."
        },
        "nextStep":"returnSecondaryTagKey"
     },
     {
        "name":"returnSecondaryTagKey",
        "action":"aws:executeScript",
        "timeoutSeconds":120,
        "onFailure":"Abort",
        "inputs":{
           "Runtime":"python3.7",
           "Handler":"returnTagValues",
           "InputPayload":{
              "secondaryTag":"{{SecondaryPatchGroupTag}}"
           },
           "Script":"..."
        },
        "outputs":[
           {
              "Name":"Payload",
              "Selector":"$.Payload",
              "Type":"StringMap"
           },
           {
              "Name":"secondaryPatchGroupKey",
              "Selector":"$.Payload.tagKey",
              "Type":"String"
           }
        ],
        "nextStep":"returnSecondaryTagValue"
     },
     {
        "name":"returnSecondaryTagValue",
        "action":"aws:executeScript",
        "timeoutSeconds":120,
```

```
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.7",
            "Handler":"returnTagValues",
            "InputPayload":{
               "secondaryTag":"{{SecondaryPatchGroupTag}}"
            },
            "Script":"..."
         },
         "outputs":[
            {
               "Name":"Payload",
               "Selector":"$.Payload",
               "Type":"StringMap"
            },
            {
               "Name":"secondaryPatchGroupValue",
               "Selector":"$.Payload.tagValue",
               "Type":"String"
            }
         ],
         "nextStep":"patchSecondaryInstances"
      },
      {
         "name":"patchSecondaryInstances",
         "action":"aws:runCommand",
         "onFailure":"Abort",
         "timeoutSeconds":7200,
         "inputs":{
            "DocumentName":"AWS-RunPatchBaseline",
            "Parameters":{
               "SnapshotId":"{{SnapshotId}}",
               "RebootOption":"{{RebootOption}}",
               "Operation":"{{Operation}}"
            },
            "Targets":[
               {
                  "Key":"{{returnSecondaryTagKey.secondaryPatchGroupKey}}",
                  "Values":[
                     "{{returnSecondaryTagValue.secondaryPatchGroupValue}}"
                  ]
               }
            ],
            "MaxConcurrency":"10%",
```

```
            "MaxErrors":"10%"
         },
         "nextStep":"returnSecondaryToOriginalState"
      },
      {
         "name":"returnSecondaryToOriginalState",
         "action":"aws:executeScript",
         "timeoutSeconds":600,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.7",
            "Handler":"returnToOriginalState",
            "InputPayload":{

 "targetInstances":"{{getSecondaryInstanceState.originalInstanceStates}}"
            },
            "Script":"..."
         }
      }
   ]
}
```

8. Emily analisa o conteúdo do runbook com script concluído e cria o runbook na mesma Conta da AWS e Região da AWS das instâncias de destino. Agora ela está pronta para testar seu runbook para garantir que a automação funcione conforme desejado antes de implementá-la em seu ambiente de produção. Abaixo está o conteúdo do runbook com script concluído.

   YAML

```
description: An example of an Automation runbook that patches groups of Amazon EC2
 instances in stages.
schemaVersion: '0.3'
assumeRole: '{{AutomationAssumeRole}}'
parameters:
  AutomationAssumeRole:
    type: String
    description: '(Required) The Amazon Resource Name (ARN) of the IAM role that
 allows Automation to perform the actions on your behalf. If no role is specified,
 Systems Manager Automation uses your IAM permissions to operate this runbook.'
  PrimaryPatchGroupTag:
    type: StringMap
    description: '(Required) The tag for the primary group of instances you want
 to patch. Specify a key-value pair. Example: {"key" : "value"}'
  SecondaryPatchGroupTag:
```

```
    type: StringMap
    description: '(Required) The tag for the secondary group of instances you want
 to patch. Specify a key-value pair. Example: {"key" : "value"}'
  SnapshotId:
    type: String
    description: '(Optional) The snapshot ID to use to retrieve a patch baseline
 snapshot.'
    default: ''
  RebootOption:
    type: String
    description: '(Optional) Reboot behavior after a patch Install operation. If
 you choose NoReboot and patches are installed, the instance is marked as non-
compliant until a subsequent reboot and scan.'
    allowedValues:
      - NoReboot
      - RebootIfNeeded
    default: RebootIfNeeded
  Operation:
    type: String
    description: '(Optional) The update or configuration to perform on the
 instance. The system checks if patches specified in the patch baseline are
 installed on the instance. The install operation installs patches missing from
 the baseline.'
    allowedValues:
      - Install
      - Scan
    default: Install
mainSteps:
  - name: getPrimaryInstanceState
    action: 'aws:executeScript'
    timeoutSeconds: 120
    onFailure: Abort
    inputs:
      Runtime: python3.7
      Handler: getInstanceStates
      InputPayload:
        primaryTag: '{{PrimaryPatchGroupTag}}'
      Script: |-
        def getInstanceStates(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          tag = events['primaryTag']
```

```
          tagKey, tagValue = list(tag.items())[0]
          instanceQuery = ec2.describe_instances(
          Filters=[
              {
                  "Name": "tag:" + tagKey,
                  "Values": [tagValue]
              }]
          )
          if not instanceQuery['Reservations']:
              noInstancesForTagString = "No instances found for specified tag."
              return({ 'noInstancesFound' : noInstancesForTagString })
          else:
              queryResponse = instanceQuery['Reservations']
              originalInstanceStates = {}
              for results in queryResponse:
                  instanceSet = results['Instances']
                  for instance in instanceSet:
                      instanceId = instance['InstanceId']
                      originalInstanceStates[instanceId] = instance['State']
['Name']
              return originalInstanceStates
    outputs:
      - Name: originalInstanceStates
        Selector: $.Payload
        Type: StringMap
    nextStep: verifyPrimaryInstancesRunning
  - name: verifyPrimaryInstancesRunning
    action: 'aws:executeScript'
    timeoutSeconds: 600
    onFailure: Abort
    inputs:
      Runtime: python3.7
      Handler: verifyInstancesRunning
      InputPayload:
        targetInstances: '{{getPrimaryInstanceState.originalInstanceStates}}'
      Script: |-
        def verifyInstancesRunning(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          instanceDict = events['targetInstances']
          for instance in instanceDict:
            if instanceDict[instance] == 'stopped':
```

```
                print("The target instance " + instance + " is stopped. The
instance will now be started.")
                ec2.start_instances(
                    InstanceIds=[instance]
                    )
            elif instanceDict[instance] == 'stopping':
                print("The target instance " + instance + " is stopping. Polling
for instance to reach stopped state.")
                while instanceDict[instance] != 'stopped':
                    poll = ec2.get_waiter('instance_stopped')
                    poll.wait(
                        InstanceIds=[instance]
                    )
                ec2.start_instances(
                    InstanceIds=[instance]
                )
            else:
              pass
    nextStep: waitForPrimaryRunningInstances
  - name: waitForPrimaryRunningInstances
    action: 'aws:executeScript'
    timeoutSeconds: 300
    onFailure: Abort
    inputs:
      Runtime: python3.7
      Handler: waitForRunningInstances
      InputPayload:
        targetInstances: '{{getPrimaryInstanceState.originalInstanceStates}}'
      Script: |-
        def waitForRunningInstances(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          instanceDict = events['targetInstances']
          for instance in instanceDict:
              poll = ec2.get_waiter('instance_running')
              poll.wait(
                  InstanceIds=[instance]
              )
    nextStep: returnPrimaryTagKey
  - name: returnPrimaryTagKey
    action: 'aws:executeScript'
    timeoutSeconds: 120
```

```
    onFailure: Abort
    inputs:
      Runtime: python3.7
      Handler: returnTagValues
      InputPayload:
        primaryTag: '{{PrimaryPatchGroupTag}}'
      Script: |-
        def returnTagValues(events,context):
          tag = events['primaryTag']
          tagKey = list(tag)[0]
          stringKey = "tag:" + tagKey
          return {'tagKey' : stringKey}
    outputs:
      - Name: Payload
        Selector: $.Payload
        Type: StringMap
      - Name: primaryPatchGroupKey
        Selector: $.Payload.tagKey
        Type: String
    nextStep: returnPrimaryTagValue
  - name: returnPrimaryTagValue
    action: 'aws:executeScript'
    timeoutSeconds: 120
    onFailure: Abort
    inputs:
      Runtime: python3.7
      Handler: returnTagValues
      InputPayload:
        primaryTag: '{{PrimaryPatchGroupTag}}'
      Script: |-
        def returnTagValues(events,context):
          tag = events['primaryTag']
          tagKey = list(tag)[0]
          tagValue = tag[tagKey]
          return {'tagValue' : tagValue}
    outputs:
      - Name: Payload
        Selector: $.Payload
        Type: StringMap
      - Name: primaryPatchGroupValue
        Selector: $.Payload.tagValue
        Type: String
    nextStep: patchPrimaryInstances
  - name: patchPrimaryInstances
```

```
    action: 'aws:runCommand'
    onFailure: Abort
    timeoutSeconds: 7200
    inputs:
      DocumentName: AWS-RunPatchBaseline
      Parameters:
        SnapshotId: '{{SnapshotId}}'
        RebootOption: '{{RebootOption}}'
        Operation: '{{Operation}}'
      Targets:
        - Key: '{{returnPrimaryTagKey.primaryPatchGroupKey}}'
          Values:
            - '{{returnPrimaryTagValue.primaryPatchGroupValue}}'
      MaxConcurrency: 10%
      MaxErrors: 10%
    nextStep: returnPrimaryToOriginalState
  - name: returnPrimaryToOriginalState
    action: 'aws:executeScript'
    timeoutSeconds: 600
    onFailure: Abort
    inputs:
      Runtime: python3.7
      Handler: returnToOriginalState
      InputPayload:
        targetInstances: '{{getPrimaryInstanceState.originalInstanceStates}}'
      Script: |-
        def returnToOriginalState(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          instanceDict = events['targetInstances']
          for instance in instanceDict:
            if instanceDict[instance] == 'stopped' or instanceDict[instance] ==
'stopping':
                ec2.stop_instances(
                    InstanceIds=[instance]
                    )
            else:
              pass
    nextStep: getSecondaryInstanceState
  - name: getSecondaryInstanceState
    action: 'aws:executeScript'
    timeoutSeconds: 120
```

```
    onFailure: Abort
    inputs:
      Runtime: python3.7
      Handler: getInstanceStates
      InputPayload:
        secondaryTag: '{{SecondaryPatchGroupTag}}'
      Script: |-
        def getInstanceStates(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          tag = events['secondaryTag']
          tagKey, tagValue = list(tag.items())[0]
          instanceQuery = ec2.describe_instances(
          Filters=[
              {
                  "Name": "tag:" + tagKey,
                  "Values": [tagValue]
              }]
          )
          if not instanceQuery['Reservations']:
              noInstancesForTagString = "No instances found for specified tag."
              return({ 'noInstancesFound' : noInstancesForTagString })
          else:
              queryResponse = instanceQuery['Reservations']
              originalInstanceStates = {}
              for results in queryResponse:
                  instanceSet = results['Instances']
                  for instance in instanceSet:
                      instanceId = instance['InstanceId']
                      originalInstanceStates[instanceId] = instance['State']
['Name']
              return originalInstanceStates
    outputs:
      - Name: originalInstanceStates
        Selector: $.Payload
        Type: StringMap
    nextStep: verifySecondaryInstancesRunning
  - name: verifySecondaryInstancesRunning
    action: 'aws:executeScript'
    timeoutSeconds: 600
    onFailure: Abort
    inputs:
```

```
      Runtime: python3.7
      Handler: verifyInstancesRunning
      InputPayload:
        targetInstances: '{{getSecondaryInstanceState.originalInstanceStates}}'
      Script: |-
        def verifyInstancesRunning(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          instanceDict = events['targetInstances']
          for instance in instanceDict:
            if instanceDict[instance] == 'stopped':
                print("The target instance " + instance + " is stopped. The
instance will now be started.")
                ec2.start_instances(
                    InstanceIds=[instance]
                    )
            elif instanceDict[instance] == 'stopping':
                print("The target instance " + instance + " is stopping. Polling
for instance to reach stopped state.")
                while instanceDict[instance] != 'stopped':
                    poll = ec2.get_waiter('instance_stopped')
                    poll.wait(
                        InstanceIds=[instance]
                    )
                ec2.start_instances(
                    InstanceIds=[instance]
                )
            else:
              pass
    nextStep: waitForSecondaryRunningInstances
  - name: waitForSecondaryRunningInstances
    action: 'aws:executeScript'
    timeoutSeconds: 300
    onFailure: Abort
    inputs:
      Runtime: python3.7
      Handler: waitForRunningInstances
      InputPayload:
        targetInstances: '{{getSecondaryInstanceState.originalInstanceStates}}'
      Script: |-
        def waitForRunningInstances(events,context):
          import boto3
```

```
        #Initialize client
        ec2 = boto3.client('ec2')
        instanceDict = events['targetInstances']
        for instance in instanceDict:
            poll = ec2.get_waiter('instance_running')
            poll.wait(
                InstanceIds=[instance]
            )
  nextStep: returnSecondaryTagKey
- name: returnSecondaryTagKey
  action: 'aws:executeScript'
  timeoutSeconds: 120
  onFailure: Abort
  inputs:
    Runtime: python3.7
    Handler: returnTagValues
    InputPayload:
      secondaryTag: '{{SecondaryPatchGroupTag}}'
    Script: |-
      def returnTagValues(events,context):
        tag = events['secondaryTag']
        tagKey = list(tag)[0]
        stringKey = "tag:" + tagKey
        return {'tagKey' : stringKey}
  outputs:
    - Name: Payload
      Selector: $.Payload
      Type: StringMap
    - Name: secondaryPatchGroupKey
      Selector: $.Payload.tagKey
      Type: String
  nextStep: returnSecondaryTagValue
- name: returnSecondaryTagValue
  action: 'aws:executeScript'
  timeoutSeconds: 120
  onFailure: Abort
  inputs:
    Runtime: python3.7
    Handler: returnTagValues
    InputPayload:
      secondaryTag: '{{SecondaryPatchGroupTag}}'
    Script: |-
      def returnTagValues(events,context):
```

```
        tag = events['secondaryTag']
        tagKey = list(tag)[0]
        tagValue = tag[tagKey]
        return {'tagValue' : tagValue}
  outputs:
    - Name: Payload
      Selector: $.Payload
      Type: StringMap
    - Name: secondaryPatchGroupValue
      Selector: $.Payload.tagValue
      Type: String
  nextStep: patchSecondaryInstances
- name: patchSecondaryInstances
  action: 'aws:runCommand'
  onFailure: Abort
  timeoutSeconds: 7200
  inputs:
    DocumentName: AWS-RunPatchBaseline
    Parameters:
      SnapshotId: '{{SnapshotId}}'
      RebootOption: '{{RebootOption}}'
      Operation: '{{Operation}}'
    Targets:
      - Key: '{{returnSecondaryTagKey.secondaryPatchGroupKey}}'
        Values:
        - '{{returnSecondaryTagValue.secondaryPatchGroupValue}}'
    MaxConcurrency: 10%
    MaxErrors: 10%
  nextStep: returnSecondaryToOriginalState
- name: returnSecondaryToOriginalState
  action: 'aws:executeScript'
  timeoutSeconds: 600
  onFailure: Abort
  inputs:
    Runtime: python3.7
    Handler: returnToOriginalState
    InputPayload:
      targetInstances: '{{getSecondaryInstanceState.originalInstanceStates}}'
    Script: |-
      def returnToOriginalState(events,context):
        import boto3

        #Initialize client
        ec2 = boto3.client('ec2')
```

```
          instanceDict = events['targetInstances']
          for instance in instanceDict:
            if instanceDict[instance] == 'stopped' or instanceDict[instance] ==
 'stopping':
                ec2.stop_instances(
                    InstanceIds=[instance]
                    )
            else:
              pass
```

JSON

```
{
   "description":"An example of an Automation runbook that patches groups of
 Amazon EC2 instances in stages.",
   "schemaVersion":"0.3",
   "assumeRole":"{{AutomationAssumeRole}}",
   "parameters":{
      "AutomationAssumeRole":{
         "type":"String",
         "description":"(Required) The Amazon Resource Name (ARN) of the IAM role
 that allows Automation to perform the actions on your behalf. If no role is
 specified, Systems Manager Automation uses your IAM permissions to operate this
 runbook."
      },
      "PrimaryPatchGroupTag":{
         "type":"StringMap",
         "description":"(Required) The tag for the primary group of instances you
 want to patch. Specify a key-value pair. Example: {\"key\" : \"value\"}"
      },
      "SecondaryPatchGroupTag":{
         "type":"StringMap",
         "description":"(Required) The tag for the secondary group of instances
 you want to patch. Specify a key-value pair. Example: {\"key\" : \"value\"}"
      },
      "SnapshotId":{
         "type":"String",
         "description":"(Optional) The snapshot ID to use to retrieve a patch
 baseline snapshot.",
         "default":""
      },
      "RebootOption":{
         "type":"String",
```

```
        "description":"(Optional) Reboot behavior after a patch Install
operation. If you choose NoReboot and patches are installed, the instance is
marked as non-compliant until a subsequent reboot and scan.",
        "allowedValues":[
           "NoReboot",
           "RebootIfNeeded"
        ],
        "default":"RebootIfNeeded"
     },
     "Operation":{
        "type":"String",
        "description":"(Optional) The update or configuration to perform on
the instance. The system checks if patches specified in the patch baseline are
installed on the instance. The install operation installs patches missing from
the baseline.",
        "allowedValues":[
           "Install",
           "Scan"
        ],
        "default":"Install"
     }
  },
  "mainSteps":[
     {
        "name":"getPrimaryInstanceState",
        "action":"aws:executeScript",
        "timeoutSeconds":120,
        "onFailure":"Abort",
        "inputs":{
           "Runtime":"python3.7",
           "Handler":"getInstanceStates",
           "InputPayload":{
              "primaryTag":"{{PrimaryPatchGroupTag}}"
           },
           "Script":"..."
        },
        "outputs":[
           {
              "Name":"originalInstanceStates",
              "Selector":"$.Payload",
              "Type":"StringMap"
           }
        ],
        "nextStep":"verifyPrimaryInstancesRunning"
```

```
      },
      {
         "name":"verifyPrimaryInstancesRunning",
         "action":"aws:executeScript",
         "timeoutSeconds":600,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.7",
            "Handler":"verifyInstancesRunning",
            "InputPayload":{

"targetInstances":"{{getPrimaryInstanceState.originalInstanceStates}}"
            },
            "Script":"..."
         },
         "nextStep":"waitForPrimaryRunningInstances"
      },
      {
         "name":"waitForPrimaryRunningInstances",
         "action":"aws:executeScript",
         "timeoutSeconds":300,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.7",
            "Handler":"waitForRunningInstances",
            "InputPayload":{

"targetInstances":"{{getPrimaryInstanceState.originalInstanceStates}}"
            },
            "Script":"..."
         },
         "nextStep":"returnPrimaryTagKey"
      },
      {
         "name":"returnPrimaryTagKey",
         "action":"aws:executeScript",
         "timeoutSeconds":120,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.7",
            "Handler":"returnTagValues",
            "InputPayload":{
               "primaryTag":"{{PrimaryPatchGroupTag}}"
            },
```

```
            "Script":"..."
         },
         "outputs":[
            {
               "Name":"Payload",
               "Selector":"$.Payload",
               "Type":"StringMap"
            },
            {
               "Name":"primaryPatchGroupKey",
               "Selector":"$.Payload.tagKey",
               "Type":"String"
            }
         ],
         "nextStep":"returnPrimaryTagValue"
      },
      {
         "name":"returnPrimaryTagValue",
         "action":"aws:executeScript",
         "timeoutSeconds":120,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.7",
            "Handler":"returnTagValues",
            "InputPayload":{
               "primaryTag":"{{PrimaryPatchGroupTag}}"
            },
            "Script":"..."
         },
         "outputs":[
            {
               "Name":"Payload",
               "Selector":"$.Payload",
               "Type":"StringMap"
            },
            {
               "Name":"primaryPatchGroupValue",
               "Selector":"$.Payload.tagValue",
               "Type":"String"
            }
         ],
         "nextStep":"patchPrimaryInstances"
      },
      {
```

```
         "name":"patchPrimaryInstances",
         "action":"aws:runCommand",
         "onFailure":"Abort",
         "timeoutSeconds":7200,
         "inputs":{
            "DocumentName":"AWS-RunPatchBaseline",
            "Parameters":{
               "SnapshotId":"{{SnapshotId}}",
               "RebootOption":"{{RebootOption}}",
               "Operation":"{{Operation}}"
            },
            "Targets":[
               {
                  "Key":"{{returnPrimaryTagKey.primaryPatchGroupKey}}",
                  "Values":[
                     "{{returnPrimaryTagValue.primaryPatchGroupValue}}"
                  ]
               }
            ],
            "MaxConcurrency":"10%",
            "MaxErrors":"10%"
         },
         "nextStep":"returnPrimaryToOriginalState"
      },
      {
         "name":"returnPrimaryToOriginalState",
         "action":"aws:executeScript",
         "timeoutSeconds":600,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.7",
            "Handler":"returnToOriginalState",
            "InputPayload":{

"targetInstances":"{{getPrimaryInstanceState.originalInstanceStates}}"
            },
            "Script":"..."
         },
         "nextStep":"getSecondaryInstanceState"
      },
      {
         "name":"getSecondaryInstanceState",
         "action":"aws:executeScript",
         "timeoutSeconds":120,
```

```
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.7",
            "Handler":"getInstanceStates",
            "InputPayload":{
               "secondaryTag":"{{SecondaryPatchGroupTag}}"
            },
            "Script":"..."
         },
         "outputs":[
            {
               "Name":"originalInstanceStates",
               "Selector":"$.Payload",
               "Type":"StringMap"
            }
         ],
         "nextStep":"verifySecondaryInstancesRunning"
      },
      {
         "name":"verifySecondaryInstancesRunning",
         "action":"aws:executeScript",
         "timeoutSeconds":600,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.7",
            "Handler":"verifyInstancesRunning",
            "InputPayload":{

"targetInstances":"{{getSecondaryInstanceState.originalInstanceStates}}"
            },
            "Script":"..."
         },
         "nextStep":"waitForSecondaryRunningInstances"
      },
      {
         "name":"waitForSecondaryRunningInstances",
         "action":"aws:executeScript",
         "timeoutSeconds":300,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.7",
            "Handler":"waitForRunningInstances",
            "InputPayload":{
```

```
"targetInstances":"{{getSecondaryInstanceState.originalInstanceStates}}"
            },
            "Script":"..."
         },
         "nextStep":"returnSecondaryTagKey"
      },
      {
         "name":"returnSecondaryTagKey",
         "action":"aws:executeScript",
         "timeoutSeconds":120,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.7",
            "Handler":"returnTagValues",
            "InputPayload":{
               "secondaryTag":"{{SecondaryPatchGroupTag}}"
            },
            "Script":"..."
         },
         "outputs":[
            {
               "Name":"Payload",
               "Selector":"$.Payload",
               "Type":"StringMap"
            },
            {
               "Name":"secondaryPatchGroupKey",
               "Selector":"$.Payload.tagKey",
               "Type":"String"
            }
         ],
         "nextStep":"returnSecondaryTagValue"
      },
      {
         "name":"returnSecondaryTagValue",
         "action":"aws:executeScript",
         "timeoutSeconds":120,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.7",
            "Handler":"returnTagValues",
            "InputPayload":{
               "secondaryTag":"{{SecondaryPatchGroupTag}}"
```

```
            },
            "Script":"..."
         },
         "outputs":[
            {
               "Name":"Payload",
               "Selector":"$.Payload",
               "Type":"StringMap"
            },
            {
               "Name":"secondaryPatchGroupValue",
               "Selector":"$.Payload.tagValue",
               "Type":"String"
            }
         ],
         "nextStep":"patchSecondaryInstances"
      },
      {
         "name":"patchSecondaryInstances",
         "action":"aws:runCommand",
         "onFailure":"Abort",
         "timeoutSeconds":7200,
         "inputs":{
            "DocumentName":"AWS-RunPatchBaseline",
            "Parameters":{
               "SnapshotId":"{{SnapshotId}}",
               "RebootOption":"{{RebootOption}}",
               "Operation":"{{Operation}}"
            },
            "Targets":[
               {
                  "Key":"{{returnSecondaryTagKey.secondaryPatchGroupKey}}",
                  "Values":[
                     "{{returnSecondaryTagValue.secondaryPatchGroupValue}}"
                  ]
               }
            ],
            "MaxConcurrency":"10%",
            "MaxErrors":"10%"
         },
         "nextStep":"returnSecondaryToOriginalState"
      },
      {
         "name":"returnSecondaryToOriginalState",
```

```
            "action":"aws:executeScript",
            "timeoutSeconds":600,
            "onFailure":"Abort",
            "inputs":{
               "Runtime":"python3.7",
               "Handler":"returnToOriginalState",
               "InputPayload":{

 "targetInstances":"{{getSecondaryInstanceState.originalInstanceStates}}"
               },
               "Script":"..."
            }
         }
      ]
}
```

Para obter mais informações sobre as ações de automação usadas neste exemplo, consulte a Referência de ações do Systems Manager Automation.

Exemplos adicionais de runbook

Os runbooks de exemplo a seguir demonstram como é possível usar as ações de automação do AWS Systems Manager para automatizar tarefas comuns de implantação, solução de problemas e manutenção.

> **Note**
>
> Os exemplos de runbooks nesta seção são fornecidos para demonstrar como é possível criar runbooks personalizados para oferecer suporte às suas necessidades operacionais específicas. Esses documentos não se destinam ao uso em ambientes de produção como estão. No entanto, você pode personalizá-los para seu próprio uso.

Exemplos

- Implantar a arquitetura da VPC e os controles de domínio do Microsoft Active Directory
- Restaurar um volume raiz do snapshot mais recente
- Crie uma AMI e uma cópia entre regiões

Implantar a arquitetura da VPC e os controles de domínio do Microsoft Active Directory

Para aumentar a eficiência e padronizar tarefas comuns, é possível optar por automatizar implantações. Isso é útil se você implantar regularmente a mesma arquitetura em várias contas e Regiões da AWS. A automatização de implantações de arquitetura também pode reduzir o potencial de erro humano que pode ocorrer ao implantar a arquitetura manualmente. AWS Systems Manager As ações do Automation podem ajudar você a fazer isso. O Automation é uma ferramenta do AWS Systems Manager.

O runbook AWS Systems Manager do exemplo a seguir realiza estas ações:

- Recupera a Amazon Machine Image (AMI) mais recente do Windows Server 2016 usando o Parameter Store do Systems Manager para usar ao executar as instâncias do EC2 que serão configuradas como controladores de domínio. O Parameter Store é uma ferramenta do AWS Systems Manager.
- Usa a ação de automação `aws:executeAwsApi` para chamar várias operações da API da AWS para criar a arquitetura da VPC. As instâncias do controlador de domínio são executadas em sub-redes privadas e se conectam à Internet usando um gateway NAT. Isso permite que o SSM Agent nas instâncias acessem os endpoints do Systems Manager necessários.
- Usa a ação de automação `aws:waitForAwsResourceProperty` para confirmar se as instâncias executadas pela ação anterior estão `Online` para o AWS Systems Manager.
- Usa a ação de automação `aws:runCommand` para configurar as instâncias executadas como controles de domínio do Microsoft Active Directory.

YAML

```
---
description: Custom Automation Deployment Example
schemaVersion: '0.3'
parameters:
  AutomationAssumeRole:
    type: String
    default: ''
    description: >-
      (Optional) The ARN of the role that allows Automation to perform the
      actions on your behalf. If no role is specified, Systems Manager
      Automation uses your IAM permissions to run this runbook.
mainSteps:
```

```
      - name: getLatestWindowsAmi
        action: aws:executeAwsApi
        onFailure: Abort
        inputs:
          Service: ssm
          Api: GetParameter
          Name: >-
            /aws/service/ami-windows-latest/Windows_Server-2016-English-Full-Base
        outputs:
          - Name: amiId
            Selector: $.Parameter.Value
            Type: String
        nextStep: createSSMInstanceRole
      - name: createSSMInstanceRole
        action: aws:executeAwsApi
        onFailure: Abort
        inputs:
          Service: iam
          Api: CreateRole
          AssumeRolePolicyDocument: >-
            {"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":
{"Service":["ec2.amazonaws.com"]},"Action":["sts:AssumeRole"]}]}
          RoleName: sampleSSMInstanceRole
        nextStep: attachManagedSSMPolicy
      - name: attachManagedSSMPolicy
        action: aws:executeAwsApi
        onFailure: Abort
        inputs:
          Service: iam
          Api: AttachRolePolicy
          PolicyArn: 'arn:aws:iam::aws:policy/service-role/
AmazonSSMManagedInstanceCore'
          RoleName: sampleSSMInstanceRole
        nextStep: createSSMInstanceProfile
      - name: createSSMInstanceProfile
        action: aws:executeAwsApi
        onFailure: Abort
        inputs:
          Service: iam
          Api: CreateInstanceProfile
          InstanceProfileName: sampleSSMInstanceRole
        outputs:
          - Name: instanceProfileArn
            Selector: $.InstanceProfile.Arn
```

```
        Type: String
    nextStep: addSSMInstanceRoleToProfile
  - name: addSSMInstanceRoleToProfile
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
      Service: iam
      Api: AddRoleToInstanceProfile
      InstanceProfileName: sampleSSMInstanceRole
      RoleName: sampleSSMInstanceRole
    nextStep: createVpc
  - name: createVpc
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
      Service: ec2
      Api: CreateVpc
      CidrBlock: 10.0.100.0/22
    outputs:
      - Name: vpcId
        Selector: $.Vpc.VpcId
        Type: String
    nextStep: getMainRtb
  - name: getMainRtb
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
      Service: ec2
      Api: DescribeRouteTables
      Filters:
        - Name: vpc-id
          Values:
            - '{{ createVpc.vpcId }}'
    outputs:
      - Name: mainRtbId
        Selector: '$.RouteTables[0].RouteTableId'
        Type: String
    nextStep: verifyMainRtb
  - name: verifyMainRtb
    action: aws:assertAwsResourceProperty
    onFailure: Abort
    inputs:
      Service: ec2
      Api: DescribeRouteTables
```

```
        RouteTableIds:
          - '{{ getMainRtb.mainRtbId }}'
        PropertySelector: '$.RouteTables[0].Associations[0].Main'
        DesiredValues:
          - 'True'
      nextStep: createPubSubnet
    - name: createPubSubnet
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: CreateSubnet
        CidrBlock: 10.0.103.0/24
        AvailabilityZone: us-west-2c
        VpcId: '{{ createVpc.vpcId }}'
      outputs:
        - Name: pubSubnetId
          Selector: $.Subnet.SubnetId
          Type: String
      nextStep: createPubRtb
    - name: createPubRtb
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: CreateRouteTable
        VpcId: '{{ createVpc.vpcId }}'
      outputs:
        - Name: pubRtbId
          Selector: $.RouteTable.RouteTableId
          Type: String
      nextStep: createIgw
    - name: createIgw
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: CreateInternetGateway
      outputs:
        - Name: igwId
          Selector: $.InternetGateway.InternetGatewayId
          Type: String
      nextStep: attachIgw
    - name: attachIgw
```

```
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: AttachInternetGateway
        InternetGatewayId: '{{ createIgw.igwId }}'
        VpcId: '{{ createVpc.vpcId }}'
      nextStep: allocateEip
    - name: allocateEip
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: AllocateAddress
        Domain: vpc
      outputs:
        - Name: eipAllocationId
          Selector: $.AllocationId
          Type: String
      nextStep: createNatGw
    - name: createNatGw
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: CreateNatGateway
        AllocationId: '{{ allocateEip.eipAllocationId }}'
        SubnetId: '{{ createPubSubnet.pubSubnetId }}'
      outputs:
        - Name: natGwId
          Selector: $.NatGateway.NatGatewayId
          Type: String
      nextStep: verifyNatGwAvailable
    - name: verifyNatGwAvailable
      action: aws:waitForAwsResourceProperty
      timeoutSeconds: 150
      inputs:
        Service: ec2
        Api: DescribeNatGateways
        NatGatewayIds:
          - '{{ createNatGw.natGwId }}'
        PropertySelector: '$.NatGateways[0].State'
        DesiredValues:
          - available
```

```
      nextStep: createNatRoute
    - name: createNatRoute
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: CreateRoute
        DestinationCidrBlock: 0.0.0.0/0
        NatGatewayId: '{{ createNatGw.natGwId }}'
        RouteTableId: '{{ getMainRtb.mainRtbId }}'
      nextStep: createPubRoute
    - name: createPubRoute
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: CreateRoute
        DestinationCidrBlock: 0.0.0.0/0
        GatewayId: '{{ createIgw.igwId }}'
        RouteTableId: '{{ createPubRtb.pubRtbId }}'
      nextStep: setPubSubAssoc
    - name: setPubSubAssoc
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: AssociateRouteTable
        RouteTableId: '{{ createPubRtb.pubRtbId }}'
        SubnetId: '{{ createPubSubnet.pubSubnetId }}'
    - name: createDhcpOptions
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: CreateDhcpOptions
        DhcpConfigurations:
          - Key: domain-name-servers
            Values:
              - '10.0.100.50,10.0.101.50'
          - Key: domain-name
            Values:
              - sample.com
      outputs:
        - Name: dhcpOptionsId
```

```
          Selector: $.DhcpOptions.DhcpOptionsId
          Type: String
    nextStep: createDCSubnet1
  - name: createDCSubnet1
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
      Service: ec2
      Api: CreateSubnet
      CidrBlock: 10.0.100.0/24
      AvailabilityZone: us-west-2a
      VpcId: '{{ createVpc.vpcId }}'
    outputs:
      - Name: firstSubnetId
        Selector: $.Subnet.SubnetId
        Type: String
    nextStep: createDCSubnet2
  - name: createDCSubnet2
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
      Service: ec2
      Api: CreateSubnet
      CidrBlock: 10.0.101.0/24
      AvailabilityZone: us-west-2b
      VpcId: '{{ createVpc.vpcId }}'
    outputs:
      - Name: secondSubnetId
        Selector: $.Subnet.SubnetId
        Type: String
    nextStep: createDCSecGroup
  - name: createDCSecGroup
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
      Service: ec2
      Api: CreateSecurityGroup
      GroupName: SampleDCSecGroup
      Description: Security Group for Sample Domain Controllers
      VpcId: '{{ createVpc.vpcId }}'
    outputs:
      - Name: dcSecGroupId
        Selector: $.GroupId
        Type: String
```

```
      nextStep: authIngressDCTraffic
    - name: authIngressDCTraffic
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: AuthorizeSecurityGroupIngress
        GroupId: '{{ createDCSecGroup.dcSecGroupId }}'
        IpPermissions:
          - FromPort: -1
            IpProtocol: '-1'
            IpRanges:
              - CidrIp: 0.0.0.0/0
                Description: Allow all traffic between Domain Controllers
      nextStep: verifyInstanceProfile
    - name: verifyInstanceProfile
      action: aws:waitForAwsResourceProperty
      maxAttempts: 5
      onFailure: Abort
      inputs:
        Service: iam
        Api: ListInstanceProfilesForRole
        RoleName: sampleSSMInstanceRole
        PropertySelector: '$.InstanceProfiles[0].Arn'
        DesiredValues:
          - '{{ createSSMInstanceProfile.instanceProfileArn }}'
      nextStep: iamEventualConsistency
    - name: iamEventualConsistency
      action: aws:sleep
      inputs:
        Duration: PT2M
      nextStep: launchDC1
    - name: launchDC1
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: RunInstances
        BlockDeviceMappings:
          - DeviceName: /dev/sda1
            Ebs:
              DeleteOnTermination: true
              VolumeSize: 50
              VolumeType: gp2
```

```
            - DeviceName: xvdf
              Ebs:
                DeleteOnTermination: true
                VolumeSize: 100
                VolumeType: gp2
          IamInstanceProfile:
            Arn: '{{ createSSMInstanceProfile.instanceProfileArn }}'
          ImageId: '{{ getLatestWindowsAmi.amiId }}'
          InstanceType: t2.micro
          MaxCount: 1
          MinCount: 1
          PrivateIpAddress: 10.0.100.50
          SecurityGroupIds:
            - '{{ createDCSecGroup.dcSecGroupId }}'
          SubnetId: '{{ createDCSubnet1.firstSubnetId }}'
          TagSpecifications:
            - ResourceType: instance
              Tags:
                - Key: Name
                  Value: SampleDC1
        outputs:
          - Name: pdcInstanceId
            Selector: '$.Instances[0].InstanceId'
            Type: String
        nextStep: launchDC2
      - name: launchDC2
        action: aws:executeAwsApi
        onFailure: Abort
        inputs:
          Service: ec2
          Api: RunInstances
          BlockDeviceMappings:
            - DeviceName: /dev/sda1
              Ebs:
                DeleteOnTermination: true
                VolumeSize: 50
                VolumeType: gp2
            - DeviceName: xvdf
              Ebs:
                DeleteOnTermination: true
                VolumeSize: 100
                VolumeType: gp2
          IamInstanceProfile:
            Arn: '{{ createSSMInstanceProfile.instanceProfileArn }}'
```

```
        ImageId: '{{ getLatestWindowsAmi.amiId }}'
        InstanceType: t2.micro
        MaxCount: 1
        MinCount: 1
        PrivateIpAddress: 10.0.101.50
        SecurityGroupIds:
          - '{{ createDCSecGroup.dcSecGroupId }}'
        SubnetId: '{{ createDCSubnet2.secondSubnetId }}'
        TagSpecifications:
          - ResourceType: instance
            Tags:
              - Key: Name
                Value: SampleDC2
      outputs:
        - Name: adcInstanceId
          Selector: '$.Instances[0].InstanceId'
          Type: String
      nextStep: verifyDCInstanceState
    - name: verifyDCInstanceState
      action: aws:waitForAwsResourceProperty
      inputs:
        Service: ec2
        Api: DescribeInstanceStatus
        IncludeAllInstances: true
        InstanceIds:
          - '{{ launchDC1.pdcInstanceId }}'
          - '{{ launchDC2.adcInstanceId }}'
        PropertySelector: '$.InstanceStatuses..InstanceState.Name'
        DesiredValues:
          - running
      nextStep: verifyInstancesOnlineSSM
    - name: verifyInstancesOnlineSSM
      action: aws:waitForAwsResourceProperty
      timeoutSeconds: 600
      inputs:
        Service: ssm
        Api: DescribeInstanceInformation
        InstanceInformationFilterList:
          - key: InstanceIds
            valueSet:
              - '{{ launchDC1.pdcInstanceId }}'
              - '{{ launchDC2.adcInstanceId }}'
        PropertySelector: '$.InstanceInformationList..PingStatus'
        DesiredValues:
```

```
            - Online
        nextStep: installADRoles
      - name: installADRoles
        action: aws:runCommand
        inputs:
          DocumentName: AWS-RunPowerShellScript
          InstanceIds:
            - '{{ launchDC1.pdcInstanceId }}'
            - '{{ launchDC2.adcInstanceId }}'
          Parameters:
            commands: |-
              try {
                  Install-WindowsFeature -Name AD-Domain-Services -
IncludeManagementTools
              }
              catch {
                  Write-Error "Failed to install ADDS Role."
              }
        nextStep: setAdminPassword
      - name: setAdminPassword
        action: aws:runCommand
        inputs:
          DocumentName: AWS-RunPowerShellScript
          InstanceIds:
            - '{{ launchDC1.pdcInstanceId }}'
          Parameters:
            commands:
              - net user Administrator "sampleAdminPass123!"
        nextStep: createForest
      - name: createForest
        action: aws:runCommand
        inputs:
          DocumentName: AWS-RunPowerShellScript
          InstanceIds:
            - '{{ launchDC1.pdcInstanceId }}'
          Parameters:
            commands: |-
              $dsrmPass = 'sample123!' | ConvertTo-SecureString -asPlainText -Force
              try {
                  Install-ADDSForest -DomainName "sample.com" -DomainMode 6
 -ForestMode 6 -InstallDNS -DatabasePath "D:\NTDS" -SysvolPath "D:\SYSVOL" -
SafeModeAdministratorPassword $dsrmPass -Force
              }
              catch {
```

```
                Write-Error $_
            }
            try {
                Add-DnsServerForwarder -IPAddress "10.0.100.2"
            }
            catch {
                Write-Error $_
            }
        nextStep: associateDhcpOptions
      - name: associateDhcpOptions
        action: aws:executeAwsApi
        onFailure: Abort
        inputs:
          Service: ec2
          Api: AssociateDhcpOptions
          DhcpOptionsId: '{{ createDhcpOptions.dhcpOptionsId }}'
          VpcId: '{{ createVpc.vpcId }}'
        nextStep: waitForADServices
      - name: waitForADServices
        action: aws:sleep
        inputs:
          Duration: PT1M
        nextStep: promoteADC
      - name: promoteADC
        action: aws:runCommand
        inputs:
          DocumentName: AWS-RunPowerShellScript
          InstanceIds:
            - '{{ launchDC2.adcInstanceId }}'
          Parameters:
            commands: |-
              ipconfig /renew
              $dsrmPass = 'sample123!' | ConvertTo-SecureString -asPlainText -Force
              $domAdminUser = "sample\Administrator"
              $domAdminPass = "sampleAdminPass123!" | ConvertTo-SecureString -
asPlainText -Force
              $domAdminCred = New-Object
 System.Management.Automation.PSCredential($domAdminUser,$domAdminPass)

              try {
                  Install-ADDSDomainController -DomainName "sample.com" -InstallDNS
 -DatabasePath "D:\NTDS" -SysvolPath "D:\SYSVOL" -SafeModeAdministratorPassword
 $dsrmPass -Credential $domAdminCred -Force
              }
```

```
            catch {
                Write-Error $_
            }
```

JSON

JSON

```
   {
     "description": "Custom Automation Deployment Example",
     "schemaVersion": "0.3",
     "assumeRole": "{{ AutomationAssumeRole }}",
     "parameters": {
       "AutomationAssumeRole": {
         "type": "String",
         "description": "(Optional) The ARN of the role that allows
 Automation to perform the actions on your behalf. If no role is specified,
 Systems Manager Automation uses your IAM permissions to run this runbook.",
         "default": ""
       }
     },
     "mainSteps": [
       {
         "name": "getLatestWindowsAmi",
         "action": "aws:executeAwsApi",
         "onFailure": "Abort",
         "inputs": {
           "Service": "ssm",
           "Api": "GetParameter",
           "Name": "/aws/service/ami-windows-latest/Windows_Server-2016-
English-Full-Base"
         },
         "outputs": [
           {
             "Name": "amiId",
             "Selector": "$.Parameter.Value",
             "Type": "String"
           }
         ],
         "nextStep": "createSSMInstanceRole"
       },
       {
```

```
          "name": "createSSMInstanceRole",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "iam",
            "Api": "CreateRole",
            "AssumeRolePolicyDocument": "{\"Version\":\"2012-10-17\",
\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":{\"Service\":
[\"ec2.amazonaws.com\"]},\"Action\":[\"sts:AssumeRole\"]}]}",
            "RoleName": "sampleSSMInstanceRole"
          },
          "nextStep": "attachManagedSSMPolicy"
        },
        {
          "name": "attachManagedSSMPolicy",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "iam",
            "Api": "AttachRolePolicy",
            "PolicyArn": "arn:aws:iam::aws:policy/service-role/
AmazonSSMManagedInstanceCore",
            "RoleName": "sampleSSMInstanceRole"
          },
          "nextStep": "createSSMInstanceProfile"
        },
        {
          "name": "createSSMInstanceProfile",
          "action":"aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "iam",
            "Api": "CreateInstanceProfile",
            "InstanceProfileName": "sampleSSMInstanceRole"
          },
          "outputs": [
            {
              "Name": "instanceProfileArn",
              "Selector": "$.InstanceProfile.Arn",
              "Type": "String"
            }
          ],
          "nextStep": "addSSMInstanceRoleToProfile"
        },
```

```
        {
          "name": "addSSMInstanceRoleToProfile",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "iam",
            "Api": "AddRoleToInstanceProfile",
            "InstanceProfileName": "sampleSSMInstanceRole",
            "RoleName": "sampleSSMInstanceRole"
          },
          "nextStep": "createVpc"
        },
        {
          "name": "createVpc",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "ec2",
            "Api": "CreateVpc",
            "CidrBlock": "10.0.100.0/22"
          },
          "outputs": [
            {
              "Name": "vpcId",
              "Selector": "$.Vpc.VpcId",
              "Type": "String"
            }
          ],
          "nextStep": "getMainRtb"
        },
        {
          "name": "getMainRtb",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "ec2",
            "Api": "DescribeRouteTables",
            "Filters": [
              {
                "Name": "vpc-id",
                "Values": ["{{ createVpc.vpcId }}"]
              }
            ]
          },
```

```
      "outputs": [
        {
          "Name": "mainRtbId",
          "Selector": "$.RouteTables[0].RouteTableId",
          "Type": "String"
        }
      ],
      "nextStep": "verifyMainRtb"
    },
    {
      "name": "verifyMainRtb",
      "action": "aws:assertAwsResourceProperty",
      "onFailure": "Abort",
      "inputs": {
        "Service": "ec2",
        "Api": "DescribeRouteTables",
        "RouteTableIds": ["{{ getMainRtb.mainRtbId }}"],
        "PropertySelector": "$.RouteTables[0].Associations[0].Main",
        "DesiredValues": ["True"]
      },
      "nextStep": "createPubSubnet"
    },
    {
      "name": "createPubSubnet",
      "action": "aws:executeAwsApi",
      "onFailure": "Abort",
      "inputs": {
        "Service": "ec2",
        "Api": "CreateSubnet",
        "CidrBlock": "10.0.103.0/24",
        "AvailabilityZone": "us-west-2c",
        "VpcId": "{{ createVpc.vpcId }}"
      },
      "outputs":[
        {
          "Name": "pubSubnetId",
          "Selector": "$.Subnet.SubnetId",
          "Type": "String"
        }
      ],
      "nextStep": "createPubRtb"
    },
    {
      "name": "createPubRtb",
```

```
        "action": "aws:executeAwsApi",
        "onFailure": "Abort",
        "inputs": {
          "Service": "ec2",
          "Api": "CreateRouteTable",
          "VpcId": "{{ createVpc.vpcId }}"
        },
        "outputs": [
          {
            "Name": "pubRtbId",
            "Selector": "$.RouteTable.RouteTableId",
            "Type": "String"
          }
        ],
        "nextStep": "createIgw"
      },
      {
        "name": "createIgw",
        "action": "aws:executeAwsApi",
        "onFailure": "Abort",
        "inputs": {
          "Service": "ec2",
          "Api": "CreateInternetGateway"
        },
        "outputs": [
          {
            "Name": "igwId",
            "Selector": "$.InternetGateway.InternetGatewayId",
            "Type": "String"
          }
        ],
        "nextStep": "attachIgw"
      },
      {
        "name": "attachIgw",
        "action": "aws:executeAwsApi",
        "onFailure": "Abort",
        "inputs": {
          "Service": "ec2",
          "Api": "AttachInternetGateway",
          "InternetGatewayId": "{{ createIgw.igwId }}",
          "VpcId": "{{ createVpc.vpcId }}"
        },
        "nextStep": "allocateEip"
```

```
        },
        {
          "name": "allocateEip",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "ec2",
            "Api": "AllocateAddress",
            "Domain": "vpc"
          },
          "outputs": [
            {
              "Name": "eipAllocationId",
              "Selector": "$.AllocationId",
              "Type": "String"
            }
          ],
          "nextStep": "createNatGw"
        },
        {
          "name": "createNatGw",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "ec2",
            "Api": "CreateNatGateway",
            "AllocationId": "{{ allocateEip.eipAllocationId }}",
            "SubnetId": "{{ createPubSubnet.pubSubnetId }}"
          },
          "outputs":[
            {
              "Name": "natGwId",
              "Selector": "$.NatGateway.NatGatewayId",
              "Type": "String"
            }
          ],
          "nextStep": "verifyNatGwAvailable"
        },
        {
          "name": "verifyNatGwAvailable",
          "action": "aws:waitForAwsResourceProperty",
          "timeoutSeconds": 150,
          "inputs": {
            "Service": "ec2",
```

```
            "Api": "DescribeNatGateways",
            "NatGatewayIds": [
              "{{ createNatGw.natGwId }}"
            ],
            "PropertySelector": "$.NatGateways[0].State",
            "DesiredValues": [
              "available"
            ]
          },
          "nextStep": "createNatRoute"
        },
        {
          "name": "createNatRoute",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "ec2",
            "Api": "CreateRoute",
            "DestinationCidrBlock": "0.0.0.0/0",
            "NatGatewayId": "{{ createNatGw.natGwId }}",
            "RouteTableId": "{{ getMainRtb.mainRtbId }}"
          },
          "nextStep": "createPubRoute"
        },
        {
          "name": "createPubRoute",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "ec2",
            "Api": "CreateRoute",
            "DestinationCidrBlock": "0.0.0.0/0",
            "GatewayId": "{{ createIgw.igwId }}",
            "RouteTableId": "{{ createPubRtb.pubRtbId }}"
          },
          "nextStep": "setPubSubAssoc"
        },
        {
          "name": "setPubSubAssoc",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "ec2",
            "Api": "AssociateRouteTable",
```

```
            "RouteTableId": "{{ createPubRtb.pubRtbId }}",
            "SubnetId": "{{ createPubSubnet.pubSubnetId }}"
          }
        },
        {
          "name": "createDhcpOptions",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "ec2",
            "Api": "CreateDhcpOptions",
            "DhcpConfigurations": [
              {
                "Key": "domain-name-servers",
                "Values": ["10.0.100.50,10.0.101.50"]
              },
              {
                "Key": "domain-name",
                "Values": ["sample.com"]
              }
            ]
          },
          "outputs": [
            {
              "Name": "dhcpOptionsId",
              "Selector": "$.DhcpOptions.DhcpOptionsId",
              "Type": "String"
            }
          ],
          "nextStep": "createDCSubnet1"
        },
        {
          "name": "createDCSubnet1",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "ec2",
            "Api": "CreateSubnet",
            "CidrBlock": "10.0.100.0/24",
            "AvailabilityZone": "us-west-2a",
            "VpcId": "{{ createVpc.vpcId }}"
          },
          "outputs": [
            {
```

```
            "Name": "firstSubnetId",
            "Selector": "$.Subnet.SubnetId",
            "Type": "String"
          }
        ],
        "nextStep": "createDCSubnet2"
      },
      {
        "name": "createDCSubnet2",
        "action": "aws:executeAwsApi",
        "onFailure": "Abort",
        "inputs": {
          "Service": "ec2",
          "Api": "CreateSubnet",
          "CidrBlock": "10.0.101.0/24",
          "AvailabilityZone": "us-west-2b",
          "VpcId": "{{ createVpc.vpcId }}"
        },
        "outputs": [
          {
            "Name": "secondSubnetId",
            "Selector": "$.Subnet.SubnetId",
            "Type": "String"
          }
        ],
        "nextStep": "createDCSecGroup"
      },
      {
        "name": "createDCSecGroup",
        "action": "aws:executeAwsApi",
        "onFailure": "Abort",
        "inputs": {
          "Service": "ec2",
          "Api": "CreateSecurityGroup",
          "GroupName": "SampleDCSecGroup",
          "Description": "Security Group for Example Domain Controllers",
          "VpcId": "{{ createVpc.vpcId }}"
        },
        "outputs": [
          {
            "Name": "dcSecGroupId",
            "Selector": "$.GroupId",
            "Type": "String"
          }
```

```
          ],
          "nextStep": "authIngressDCTraffic"
        },
        {
          "name": "authIngressDCTraffic",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "ec2",
            "Api": "AuthorizeSecurityGroupIngress",
            "GroupId": "{{ createDCSecGroup.dcSecGroupId }}",
            "IpPermissions": [
              {
                "FromPort": -1,
                "IpProtocol": "-1",
                "IpRanges": [
                  {
                    "CidrIp": "0.0.0.0/0",
                    "Description": "Allow all traffic between Domain
Controllers"
                  }
                ]
              }
            ]
          },
          "nextStep": "verifyInstanceProfile"
        },
        {
          "name": "verifyInstanceProfile",
          "action": "aws:waitForAwsResourceProperty",
          "maxAttempts": 5,
          "onFailure": "Abort",
          "inputs": {
            "Service": "iam",
            "Api": "ListInstanceProfilesForRole",
            "RoleName": "sampleSSMInstanceRole",
            "PropertySelector": "$.InstanceProfiles[0].Arn",
            "DesiredValues": [
              "{{ createSSMInstanceProfile.instanceProfileArn }}"
            ]
          },
          "nextStep": "iamEventualConsistency"
        },
        {
```

```
      "name": "iamEventualConsistency",
      "action": "aws:sleep",
      "inputs": {
        "Duration": "PT2M"
      },
      "nextStep": "launchDC1"
    },
    {
      "name": "launchDC1",
      "action": "aws:executeAwsApi",
      "onFailure": "Abort",
      "inputs": {
        "Service": "ec2",
        "Api": "RunInstances",
        "BlockDeviceMappings": [
          {
            "DeviceName": "/dev/sda1",
            "Ebs": {
              "DeleteOnTermination": true,
              "VolumeSize": 50,
              "VolumeType": "gp2"
            }
          },
          {
            "DeviceName": "xvdf",
            "Ebs": {
              "DeleteOnTermination": true,
              "VolumeSize": 100,
              "VolumeType": "gp2"
            }
          }
        ],
        "IamInstanceProfile": {
          "Arn": "{{ createSSMInstanceProfile.instanceProfileArn }}"
        },
        "ImageId": "{{ getLatestWindowsAmi.amiId }}",
        "InstanceType": "t2.micro",
        "MaxCount": 1,
        "MinCount": 1,
        "PrivateIpAddress": "10.0.100.50",
        "SecurityGroupIds": [
          "{{ createDCSecGroup.dcSecGroupId }}"
        ],
        "SubnetId": "{{ createDCSubnet1.firstSubnetId }}",
```

```
            "TagSpecifications": [
              {
                "ResourceType": "instance",
                "Tags": [
                  {
                    "Key": "Name",
                    "Value": "SampleDC1"
                  }
                ]
              }
            ]
          },
          "outputs": [
            {
              "Name": "pdcInstanceId",
              "Selector": "$.Instances[0].InstanceId",
              "Type": "String"
            }
          ],
          "nextStep": "launchDC2"
        },
        {
          "name": "launchDC2",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "ec2",
            "Api": "RunInstances",
            "BlockDeviceMappings": [
              {
                "DeviceName": "/dev/sda1",
                "Ebs": {
                  "DeleteOnTermination": true,
                  "VolumeSize": 50,
                  "VolumeType": "gp2"
                }
              },
              {
                "DeviceName": "xvdf",
                "Ebs": {
                  "DeleteOnTermination": true,
                  "VolumeSize": 100,
                  "VolumeType": "gp2"
                }
```

```
            }
          ],
          "IamInstanceProfile": {
            "Arn": "{{ createSSMInstanceProfile.instanceProfileArn }}"
          },
          "ImageId": "{{ getLatestWindowsAmi.amiId }}",
          "InstanceType": "t2.micro",
          "MaxCount": 1,
          "MinCount": 1,
          "PrivateIpAddress": "10.0.101.50",
          "SecurityGroupIds": [
            "{{ createDCSecGroup.dcSecGroupId }}"
          ],
          "SubnetId": "{{ createDCSubnet2.secondSubnetId }}",
          "TagSpecifications": [
            {
              "ResourceType": "instance",
              "Tags": [
                {
                  "Key": "Name",
                  "Value": "SampleDC2"
                }
              ]
            }
          ]
        },
        "outputs": [
          {
            "Name": "adcInstanceId",
            "Selector": "$.Instances[0].InstanceId",
            "Type": "String"
          }
        ],
        "nextStep": "verifyDCInstanceState"
      },
      {
        "name": "verifyDCInstanceState",
        "action": "aws:waitForAwsResourceProperty",
        "inputs": {
          "Service": "ec2",
          "Api": "DescribeInstanceStatus",
          "IncludeAllInstances": true,
          "InstanceIds": [
            "{{ launchDC1.pdcInstanceId }}",
```

```
            "{{ launchDC2.adcInstanceId }}"
          ],
          "PropertySelector": "$.InstanceStatuses[0].InstanceState.Name",
          "DesiredValues": [
            "running"
          ]
        },
        "nextStep": "verifyInstancesOnlineSSM"
      },
      {
        "name": "verifyInstancesOnlineSSM",
        "action": "aws:waitForAwsResourceProperty",
        "timeoutSeconds": 600,
        "inputs": {
          "Service": "ssm",
          "Api": "DescribeInstanceInformation",
          "InstanceInformationFilterList": [
            {
              "key": "InstanceIds",
              "valueSet": [
                "{{ launchDC1.pdcInstanceId }}",
                "{{ launchDC2.adcInstanceId }}"
              ]
            }
          ],
          "PropertySelector": "$.InstanceInformationList[0].PingStatus",
          "DesiredValues": [
            "Online"
          ]
        },
        "nextStep": "installADRoles"
      },
      {
        "name": "installADRoles",
        "action": "aws:runCommand",
        "inputs": {
          "DocumentName": "AWS-RunPowerShellScript",
          "InstanceIds": [
            "{{ launchDC1.pdcInstanceId }}",
            "{{ launchDC2.adcInstanceId }}"
          ],
          "Parameters": {
            "commands": [
              "try {",
```

```
                "  Install-WindowsFeature -Name AD-Domain-Services -
IncludeManagementTools",
                "}",
                "catch {",
                "  Write-Error \"Failed to install ADDS Role.\"",
                "}"
              ]
            }
          },
          "nextStep": "setAdminPassword"
        },
        {
          "name": "setAdminPassword",
          "action": "aws:runCommand",
          "inputs": {
            "DocumentName": "AWS-RunPowerShellScript",
            "InstanceIds": [
              "{{ launchDC1.pdcInstanceId }}"
            ],
            "Parameters": {
              "commands": [
                "net user Administrator \"sampleAdminPass123!\""
              ]
            }
          },
          "nextStep": "createForest"
        },
        {
          "name": "createForest",
          "action": "aws:runCommand",
          "inputs": {
            "DocumentName": "AWS-RunPowerShellScript",
            "InstanceIds": [
              "{{ launchDC1.pdcInstanceId }}"
            ],
            "Parameters": {
              "commands": [
                "$dsrmPass = 'sample123!' | ConvertTo-SecureString -
asPlainText -Force",
                "try {",
                "   Install-ADDSForest -DomainName \"sample.com\" -DomainMode
 6 -ForestMode 6 -InstallDNS -DatabasePath \"D:\\NTDS\" -SysvolPath \"D:\
\SYSVOL\" -SafeModeAdministratorPassword $dsrmPass -Force",
                "}",
```

```
            "catch {",
            "   Write-Error $_",
            "}",
            "try {",
            "   Add-DnsServerForwarder -IPAddress \"10.0.100.2\"",
            "}",
            "catch {",
            "   Write-Error $_",
            "}"
          ]
        }
      },
      "nextStep": "associateDhcpOptions"
    },
    {
      "name": "associateDhcpOptions",
      "action": "aws:executeAwsApi",
      "onFailure": "Abort",
      "inputs": {
        "Service": "ec2",
        "Api": "AssociateDhcpOptions",
        "DhcpOptionsId": "{{ createDhcpOptions.dhcpOptionsId }}",
        "VpcId": "{{ createVpc.vpcId }}"
      },
      "nextStep": "waitForADServices"
    },
    {
      "name": "waitForADServices",
      "action": "aws:sleep",
      "inputs": {
        "Duration": "PT1M"
      },
      "nextStep": "promoteADC"
    },
    {
      "name": "promoteADC",
      "action": "aws:runCommand",
      "inputs": {
        "DocumentName": "AWS-RunPowerShellScript",
        "InstanceIds": [
          "{{ launchDC2.adcInstanceId }}"
        ],
        "Parameters": {
          "commands": [
```

```
                "ipconfig /renew",
                "$dsrmPass = 'sample123!' | ConvertTo-SecureString -
asPlainText -Force",
                "$domAdminUser = \"sample\\Administrator\"",
                "$domAdminPass = \"sampleAdminPass123!\" | ConvertTo-
SecureString -asPlainText -Force",
                "$domAdminCred = New-Object
 System.Management.Automation.PSCredential($domAdminUser,$domAdminPass)",
                "try {",
                "   Install-ADDSDomainController -DomainName \"sample.com
\" -InstallDNS -DatabasePath \"D:\\NTDS\" -SysvolPath \"D:\\SYSVOL\" -
SafeModeAdministratorPassword $dsrmPass -Credential $domAdminCred -Force",
                "}",
                "catch {",
                "   Write-Error $_",
                "}"
              ]
            }
          }
        }
      ]
    }
```

Restaurar um volume raiz do snapshot mais recente

O sistema operacional em um volume raiz pode ser corrompido por vários motivos. Por exemplo, após uma operação de patch, as instâncias podem não inicializar com êxito, devido a um kernel ou registro corrompido. A automatização de tarefas comuns de solução de problemas, como restaurar um volume raiz do snapshot mais recente feito antes da operação de patch, pode reduzir o tempo de inatividade e agilizar seus esforços de solução de problemas. AWS Systems Manager As ações do Automation podem ajudar você a fazer isso. O Automation é uma ferramenta do AWS Systems Manager.

O runbook AWS Systems Manager do exemplo a seguir realiza estas ações:

- Usa a ação de automação `aws:executeAwsApi` para recuperar detalhes do volume raiz da instância.
- Usa a ação de automação `aws:executeScript` para recuperar o snapshot mais recente do volume raiz.

- Usa a ação de automação `aws:branch` para continuar a automação, se um snapshot for encontrado para o volume raiz.

YAML

```
   ---
   description: Custom Automation Troubleshooting Example
   schemaVersion: '0.3'
   assumeRole: "{{ AutomationAssumeRole }}"
   parameters:
     AutomationAssumeRole:
       type: String
       description: "(Required) The ARN of the role that allows Automation to
perform
         the actions on your behalf. If no role is specified, Systems Manager
Automation
         uses your IAM permissions to use this runbook."
       default: ''
     InstanceId:
         type: String
         description: "(Required) The Instance Id whose root EBS volume you want to
restore the latest Snapshot."
         default: ''
   mainSteps:
   - name: getInstanceDetails
     action: aws:executeAwsApi
     onFailure: Abort
     inputs:
       Service: ec2
       Api: DescribeInstances
       InstanceIds:
       - "{{ InstanceId }}"
     outputs:
       - Name: availabilityZone
         Selector: "$.Reservations[0].Instances[0].Placement.AvailabilityZone"
         Type: String
       - Name: rootDeviceName
         Selector: "$.Reservations[0].Instances[0].RootDeviceName"
         Type: String
     nextStep: getRootVolumeId
   - name: getRootVolumeId
     action: aws:executeAwsApi
```

```
    onFailure: Abort
    inputs:
      Service: ec2
      Api: DescribeVolumes
      Filters:
      -  Name: attachment.device
         Values: ["{{ getInstanceDetails.rootDeviceName }}"]
      -  Name: attachment.instance-id
         Values: ["{{ InstanceId }}"]
    outputs:
      - Name: rootVolumeId
        Selector: "$.Volumes[0].VolumeId"
        Type: String
    nextStep: getSnapshotsByStartTime
  - name: getSnapshotsByStartTime
    action: aws:executeScript
    timeoutSeconds: 45
    onFailure: Abort
    inputs:
      Runtime: python3.8
      Handler: getSnapshotsByStartTime
      InputPayload:
        rootVolumeId : "{{ getRootVolumeId.rootVolumeId }}"
      Script: |-
        def getSnapshotsByStartTime(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          rootVolumeId = events['rootVolumeId']
          snapshotsQuery = ec2.describe_snapshots(
            Filters=[
              {
                "Name": "volume-id",
                "Values": [rootVolumeId]
              }
            ]
          )
          if not snapshotsQuery['Snapshots']:
            noSnapshotFoundString = "NoSnapshotFound"
            return { 'noSnapshotFound' : noSnapshotFoundString }
          else:
            jsonSnapshots = snapshotsQuery['Snapshots']
```

```
            sortedSnapshots = sorted(jsonSnapshots, key=lambda k: k['StartTime'],
reverse=True)
            latestSortedSnapshotId = sortedSnapshots[0]['SnapshotId']
            return { 'latestSnapshotId' : latestSortedSnapshotId }
    outputs:
    - Name: Payload
      Selector: $.Payload
      Type: StringMap
    - Name: latestSnapshotId
      Selector: $.Payload.latestSnapshotId
      Type: String
    - Name: noSnapshotFound
      Selector: $.Payload.noSnapshotFound
      Type: String
    nextStep: branchFromResults
  - name: branchFromResults
    action: aws:branch
    onFailure: Abort
    inputs:
      Choices:
      - NextStep: createNewRootVolumeFromSnapshot
        Not:
          Variable: "{{ getSnapshotsByStartTime.noSnapshotFound }}"
          StringEquals: "NoSnapshotFound"
    isEnd: true
  - name: createNewRootVolumeFromSnapshot
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
      Service: ec2
      Api: CreateVolume
      AvailabilityZone: "{{ getInstanceDetails.availabilityZone }}"
      SnapshotId: "{{ getSnapshotsByStartTime.latestSnapshotId }}"
    outputs:
      - Name: newRootVolumeId
        Selector: "$.VolumeId"
        Type: String
    nextStep: stopInstance
  - name: stopInstance
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
      Service: ec2
      Api: StopInstances
```

```
        InstanceIds:
        - "{{ InstanceId }}"
      nextStep: verifyVolumeAvailability
    - name: verifyVolumeAvailability
      action: aws:waitForAwsResourceProperty
      timeoutSeconds: 120
      inputs:
        Service: ec2
        Api: DescribeVolumes
        VolumeIds:
        - "{{ createNewRootVolumeFromSnapshot.newRootVolumeId }}"
        PropertySelector: "$.Volumes[0].State"
        DesiredValues:
        - "available"
      nextStep: verifyInstanceStopped
    - name: verifyInstanceStopped
      action: aws:waitForAwsResourceProperty
      timeoutSeconds: 120
      inputs:
        Service: ec2
        Api: DescribeInstances
        InstanceIds:
        - "{{ InstanceId }}"
        PropertySelector: "$.Reservations[0].Instances[0].State.Name"
        DesiredValues:
        - "stopped"
      nextStep: detachRootVolume
    - name: detachRootVolume
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: DetachVolume
        VolumeId: "{{ getRootVolumeId.rootVolumeId }}"
      nextStep: verifyRootVolumeDetached
    - name: verifyRootVolumeDetached
      action: aws:waitForAwsResourceProperty
      timeoutSeconds: 30
      inputs:
        Service: ec2
        Api: DescribeVolumes
        VolumeIds:
        - "{{ getRootVolumeId.rootVolumeId }}"
        PropertySelector: "$.Volumes[0].State"
```

```
      DesiredValues:
      - "available"
    nextStep: attachNewRootVolume
  - name: attachNewRootVolume
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
      Service: ec2
      Api: AttachVolume
      Device: "{{ getInstanceDetails.rootDeviceName }}"
      InstanceId: "{{ InstanceId }}"
      VolumeId: "{{ createNewRootVolumeFromSnapshot.newRootVolumeId }}"
    nextStep: verifyNewRootVolumeAttached
  - name: verifyNewRootVolumeAttached
    action: aws:waitForAwsResourceProperty
    timeoutSeconds: 30
    inputs:
      Service: ec2
      Api: DescribeVolumes
      VolumeIds:
      - "{{ createNewRootVolumeFromSnapshot.newRootVolumeId }}"
      PropertySelector: "$.Volumes[0].Attachments[0].State"
      DesiredValues:
      - "attached"
    nextStep: startInstance
  - name: startInstance
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
      Service: ec2
      Api: StartInstances
      InstanceIds:
      - "{{ InstanceId }}"
```

JSON

```
{
   "description": "Custom Automation Troubleshooting Example",
   "schemaVersion": "0.3",
   "assumeRole": "{{ AutomationAssumeRole }}",
   "parameters": {
      "AutomationAssumeRole": {
```

```
            "type": "String",
            "description": "(Required) The ARN of the role that allows Automation
to perform the actions on your behalf. If no role is specified, Systems Manager
Automation uses your IAM permissions to run this runbook.",
            "default": ""
         },
         "InstanceId": {
            "type": "String",
            "description": "(Required) The Instance Id whose root EBS volume you
want to restore the latest Snapshot.",
            "default": ""
         }
      },
      "mainSteps": [
         {
            "name": "getInstanceDetails",
            "action": "aws:executeAwsApi",
            "onFailure": "Abort",
            "inputs": {
               "Service": "ec2",
               "Api": "DescribeInstances",
               "InstanceIds": [
                  "{{ InstanceId }}"
               ]
            },
            "outputs": [
               {
                  "Name": "availabilityZone",
                  "Selector":
"$.Reservations[0].Instances[0].Placement.AvailabilityZone",
                  "Type": "String"
               },
               {
                  "Name": "rootDeviceName",
                  "Selector": "$.Reservations[0].Instances[0].RootDeviceName",
                  "Type": "String"
               }
            ],
            "nextStep": "getRootVolumeId"
         },
         {
            "name": "getRootVolumeId",
            "action": "aws:executeAwsApi",
            "onFailure": "Abort",
```

```
            "inputs": {
               "Service": "ec2",
               "Api": "DescribeVolumes",
               "Filters": [
                  {
                     "Name": "attachment.device",
                     "Values": [
                        "{{ getInstanceDetails.rootDeviceName }}"
                     ]
                  },
                  {
                     "Name": "attachment.instance-id",
                     "Values": [
                        "{{ InstanceId }}"
                     ]
                  }
               ]
            },
            "outputs": [
               {
                  "Name": "rootVolumeId",
                  "Selector": "$.Volumes[0].VolumeId",
                  "Type": "String"
               }
            ],
            "nextStep": "getSnapshotsByStartTime"
         },
         {
            "name": "getSnapshotsByStartTime",
            "action": "aws:executeScript",
            "timeoutSeconds": 45,
            "onFailure": "Continue",
            "inputs": {
               "Runtime": "python3.8",
               "Handler": "getSnapshotsByStartTime",
               "InputPayload": {
                  "rootVolumeId": "{{ getRootVolumeId.rootVolumeId }}"
               },
               "Attachment": "getSnapshotsByStartTime.py"
            },
            "outputs": [
               {
                  "Name": "Payload",
                  "Selector": "$.Payload",
```

```
                  "Type": "StringMap"
               },
               {
                  "Name": "latestSnapshotId",
                  "Selector": "$.Payload.latestSnapshotId",
                  "Type": "String"
               },
               {
                  "Name": "noSnapshotFound",
                  "Selector": "$.Payload.noSnapshotFound",
                  "Type": "String"
               }
            ],
            "nextStep": "branchFromResults"
         },
         {
            "name": "branchFromResults",
            "action": "aws:branch",
            "onFailure": "Abort",
            "inputs": {
               "Choices": [
                  {
                     "NextStep": "createNewRootVolumeFromSnapshot",
                     "Not": {
                        "Variable":
"{{ getSnapshotsByStartTime.noSnapshotFound }}",
                        "StringEquals": "NoSnapshotFound"
                     }
                  }
               ]
            },
            "isEnd": true
         },
         {
            "name": "createNewRootVolumeFromSnapshot",
            "action": "aws:executeAwsApi",
            "onFailure": "Abort",
            "inputs": {
               "Service": "ec2",
               "Api": "CreateVolume",
               "AvailabilityZone": "{{ getInstanceDetails.availabilityZone }}",
               "SnapshotId": "{{ getSnapshotsByStartTime.latestSnapshotId }}"
            },
            "outputs": [
```

```
            {
               "Name": "newRootVolumeId",
               "Selector": "$.VolumeId",
               "Type": "String"
            }
         ],
         "nextStep": "stopInstance"
      },
      {
         "name": "stopInstance",
         "action": "aws:executeAwsApi",
         "onFailure": "Abort",
         "inputs": {
            "Service": "ec2",
            "Api": "StopInstances",
            "InstanceIds": [
               "{{ InstanceId }}"
            ]
         },
         "nextStep": "verifyVolumeAvailability"
      },
      {
         "name": "verifyVolumeAvailability",
         "action": "aws:waitForAwsResourceProperty",
         "timeoutSeconds": 120,
         "inputs": {
            "Service": "ec2",
            "Api": "DescribeVolumes",
            "VolumeIds": [
               "{{ createNewRootVolumeFromSnapshot.newRootVolumeId }}"
            ],
            "PropertySelector": "$.Volumes[0].State",
            "DesiredValues": [
               "available"
            ]
         },
         "nextStep": "verifyInstanceStopped"
      },
      {
         "name": "verifyInstanceStopped",
         "action": "aws:waitForAwsResourceProperty",
         "timeoutSeconds": 120,
         "inputs": {
            "Service": "ec2",
```

```
            "Api": "DescribeInstances",
            "InstanceIds": [
               "{{ InstanceId }}"
            ],
            "PropertySelector": "$.Reservations[0].Instances[0].State.Name",
            "DesiredValues": [
               "stopped"
            ]
         },
         "nextStep": "detachRootVolume"
      },
      {
         "name": "detachRootVolume",
         "action": "aws:executeAwsApi",
         "onFailure": "Abort",
         "inputs": {
            "Service": "ec2",
            "Api": "DetachVolume",
            "VolumeId": "{{ getRootVolumeId.rootVolumeId }}"
         },
         "nextStep": "verifyRootVolumeDetached"
      },
      {
         "name": "verifyRootVolumeDetached",
         "action": "aws:waitForAwsResourceProperty",
         "timeoutSeconds": 30,
         "inputs": {
            "Service": "ec2",
            "Api": "DescribeVolumes",
            "VolumeIds": [
               "{{ getRootVolumeId.rootVolumeId }}"
            ],
            "PropertySelector": "$.Volumes[0].State",
            "DesiredValues": [
               "available"
            ]
         },
         "nextStep": "attachNewRootVolume"
      },
      {
         "name": "attachNewRootVolume",
         "action": "aws:executeAwsApi",
         "onFailure": "Abort",
         "inputs": {
```

```
                "Service": "ec2",
                "Api": "AttachVolume",
                "Device": "{{ getInstanceDetails.rootDeviceName }}",
                "InstanceId": "{{ InstanceId }}",
                "VolumeId": "{{ createNewRootVolumeFromSnapshot.newRootVolumeId }}"
             },
             "nextStep": "verifyNewRootVolumeAttached"
          },
          {
             "name": "verifyNewRootVolumeAttached",
             "action": "aws:waitForAwsResourceProperty",
             "timeoutSeconds": 30,
             "inputs": {
                "Service": "ec2",
                "Api": "DescribeVolumes",
                "VolumeIds": [
                   "{{ createNewRootVolumeFromSnapshot.newRootVolumeId }}"
                ],
                "PropertySelector": "$.Volumes[0].Attachments[0].State",
                "DesiredValues": [
                   "attached"
                ]
             },
             "nextStep": "startInstance"
          },
          {
             "name": "startInstance",
             "action": "aws:executeAwsApi",
             "onFailure": "Abort",
             "inputs": {
                "Service": "ec2",
                "Api": "StartInstances",
                "InstanceIds": [
                   "{{ InstanceId }}"
                ]
             }
          }
       ],
       "files": {
            "getSnapshotsByStartTime.py": {
                "checksums": {
                    "sha256": "sampleETagValue"
                }
            }
```

```
        }
    }
```

Crie uma AMI e uma cópia entre regiões

Criar uma Amazon Machine Image (AMI) de uma instância é um processo comum usado em backup e recuperação. Também é possível optar por copiar uma AMI em outra Região da AWS como parte de uma arquitetura de recuperação de desastres. A automação de tarefas de manutenção comuns poderá reduzir o tempo de inatividade se um problema exigir failover. AWS Systems Manager As ações do Automation podem ajudar você a fazer isso. O Automation é uma ferramenta do AWS Systems Manager.

O runbook AWS Systems Manager de exemplo a seguir realiza estas ações:

- Usa a ação de automação `aws:executeAwsApi` para criar uma AMI.
- Usa a ação de automação `aws:waitForAwsResourceProperty` para confirmar a disponibilidade da AMI.
- Usa a ação de automação `aws:executeScript` para copiar a AMI na região de destino.

YAML

```
    ---
    description: Custom Automation Backup and Recovery Example
    schemaVersion: '0.3'
    assumeRole: "{{ AutomationAssumeRole }}"
    parameters:
      AutomationAssumeRole:
        type: String
        description: "(Required) The ARN of the role that allows Automation to
perform
          the actions on your behalf. If no role is specified, Systems Manager
Automation
          uses your IAM permissions to use this runbook."
        default: ''
      InstanceId:
        type: String
        description: "(Required) The ID of the EC2 instance."
        default: ''
    mainSteps:
```

```
    - name: createImage
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: CreateImage
        InstanceId: "{{ InstanceId }}"
        Name: "Automation Image for {{ InstanceId }}"
        NoReboot: false
      outputs:
        - Name: newImageId
          Selector: "$.ImageId"
          Type: String
      nextStep: verifyImageAvailability
    - name: verifyImageAvailability
      action: aws:waitForAwsResourceProperty
      timeoutSeconds: 600
      inputs:
        Service: ec2
        Api: DescribeImages
        ImageIds:
        - "{{ createImage.newImageId }}"
        PropertySelector: "$.Images[0].State"
        DesiredValues:
        - available
      nextStep: copyImage
    - name: copyImage
      action: aws:executeScript
      timeoutSeconds: 45
      onFailure: Abort
      inputs:
        Runtime: python3.8
        Handler: crossRegionImageCopy
        InputPayload:
          newImageId : "{{ createImage.newImageId }}"
        Script: |-
          def crossRegionImageCopy(events,context):
            import boto3

            #Initialize client
            ec2 = boto3.client('ec2', region_name='us-east-1')
            newImageId = events['newImageId']

            ec2.copy_image(
```

```
            Name='DR Copy for ' + newImageId,
            SourceImageId=newImageId,
            SourceRegion='us-west-2'
          )
```

JSON

```
   {
      "description": "Custom Automation Backup and Recovery Example",
      "schemaVersion": "0.3",
      "assumeRole": "{{ AutomationAssumeRole }}",
      "parameters": {
         "AutomationAssumeRole": {
            "type": "String",
            "description": "(Required) The ARN of the role that allows Automation
to perform\nthe actions on your behalf. If no role is specified, Systems Manager
Automation\nuses your IAM permissions to run this runbook.",
            "default": ""
         },
         "InstanceId": {
            "type": "String",
            "description": "(Required) The ID of the EC2 instance.",
            "default": ""
         }
      },
      "mainSteps": [
         {
            "name": "createImage",
            "action": "aws:executeAwsApi",
            "onFailure": "Abort",
            "inputs": {
               "Service": "ec2",
               "Api": "CreateImage",
               "InstanceId": "{{ InstanceId }}",
               "Name": "Automation Image for {{ InstanceId }}",
               "NoReboot": false
            },
            "outputs": [
               {
                  "Name": "newImageId",
                  "Selector": "$.ImageId",
                  "Type": "String"
```

```
            }
         ],
         "nextStep": "verifyImageAvailability"
      },
      {
         "name": "verifyImageAvailability",
         "action": "aws:waitForAwsResourceProperty",
         "timeoutSeconds": 600,
         "inputs": {
            "Service": "ec2",
            "Api": "DescribeImages",
            "ImageIds": [
               "{{ createImage.newImageId }}"
            ],
            "PropertySelector": "$.Images[0].State",
            "DesiredValues": [
               "available"
            ]
         },
         "nextStep": "copyImage"
      },
      {
         "name": "copyImage",
         "action": "aws:executeScript",
         "timeoutSeconds": 45,
         "onFailure": "Abort",
         "inputs": {
            "Runtime": "python3.8",
            "Handler": "crossRegionImageCopy",
            "InputPayload": {
               "newImageId": "{{ createImage.newImageId }}"
            },
            "Attachment": "crossRegionImageCopy.py"
         }
      }
   ],
   "files": {
      "crossRegionImageCopy.py": {
         "checksums": {
            "sha256": "sampleETagValue"
         }
      }
   }
```

```
    }
```

Criar parâmetros de entrada que preenchem os recursos da AWS

O Automation, uma ferramenta do Systems Manager, preenche recursos da AWS no AWS Management Console que correspondem ao tipo de recurso definido para um parâmetro de entrada. Recursos na sua Conta da AWS, que correspondem ao tipo de recurso, são exibidos em uma lista suspensa para você escolher. Você pode definir tipos de parâmetros de entrada para instâncias do Amazon Elastic Compute Cloud (Amazon EC2), buckets do Amazon Simple Storage Service (Amazon S3) e funções do AWS Identity and Access Management (IAM). As definições de tipo suportadas e as expressões regulares usadas para localizar recursos correspondentes são as seguintes:

- `AWS::EC2::Instance::Id` - `^m?i-([a-z0-9]{8}|[a-z0-9]{17})$`
- `List<AWS::EC2::Instance::Id>` - `^m?i-([a-z0-9]{8}|[a-z0-9]{17})$`
- `AWS::S3::Bucket::Name` - `^[0-9a-z][a-z0-9\\-\\.]{3,63}$`
- `List<AWS::S3::Bucket::Name>` - `^[0-9a-z][a-z0-9\\-\\.]{3,63}$`
- `AWS::IAM::Role::Arn` - `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):iam::[0-9]{12}:role/.*$`
- `List<AWS::IAM::Role::Arn>` - `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):iam::[0-9]{12}:role/.*$`

O exemplo a seguir são tipos de parâmetros de entrada definidos no conteúdo do runbook.

YAML

```
description: Enables encryption on an Amazon S3 bucket
schemaVersion: '0.3'
assumeRole: '{{ AutomationAssumeRole }}'
parameters:
  BucketName:
    type: 'AWS::S3::Bucket::Name'
    description: (Required) The name of the Amazon S3 bucket you want to encrypt.
  SSEAlgorithm:
    type: String
    description: (Optional) The server-side encryption algorithm to use for the
 default encryption.
    default: AES256
```

```
  AutomationAssumeRole:
    type: 'AWS::IAM::Role::Arn'
    description: (Optional) The Amazon Resource Name (ARN) of the role that allows
 Automation to perform the actions on your behalf.
    default: ''
mainSteps:
  - name: enableBucketEncryption
    action: 'aws:executeAwsApi'
    inputs:
      Service: s3
      Api: PutBucketEncryption
      Bucket: '{{BucketName}}'
      ServerSideEncryptionConfiguration:
        Rules:
          - ApplyServerSideEncryptionByDefault:
              SSEAlgorithm: '{{SSEAlgorithm}}'
    isEnd: true
```

JSON

```
{
   "description": "Enables encryption on an Amazon S3 bucket",
   "schemaVersion": "0.3",
   "assumeRole": "{{ AutomationAssumeRole }}",
   "parameters": {
      "BucketName": {
         "type": "AWS::S3::Bucket::Name",
         "description": "(Required) The name of the Amazon S3 bucket you want to
 encrypt."
      },
      "SSEAlgorithm": {
         "type": "String",
         "description": "(Optional) The server-side encryption algorithm to use for
 the default encryption.",
         "default": "AES256"
      },
      "AutomationAssumeRole": {
         "type": "AWS::IAM::Role::Arn",
         "description": "(Optional) The Amazon Resource Name (ARN) of the role that
 allows Automation to perform the actions on your behalf.",
         "default": ""
      }
   },
```

```
   "mainSteps": [
      {
         "name": "enableBucketEncryption",
         "action": "aws:executeAwsApi",
         "inputs": {
            "Service": "s3",
            "Api": "PutBucketEncryption",
            "Bucket": "{{BucketName}}",
            "ServerSideEncryptionConfiguration": {
               "Rules": [
                  {
                     "ApplyServerSideEncryptionByDefault": {
                        "SSEAlgorithm": "{{SSEAlgorithm}}"
                     }
                  }
               ]
            }
         },
         "isEnd": true
      }
   ]
}
```

Uso do Document Builder para criar runbooks

Se os runbooks públicos do AWS Systems Manager, não forem compatíveis com todas as ações que você quiser executar em seus recursos da AWS, você poderá criar seus próprios runbooks. Para criar um runbook personalizado, é possível criar manualmente um arquivo local em formato YAML ou JSON com as ações de automação apropriadas. Como alternativa, é possível usar o Document Builder no console do Systems Manager Automation para criar um runbook personalizado.

Usando o Document Builder, é possível adicionar etapas de ação de automação ao runbook personalizado e fornecer os parâmetros necessários sem precisar usar sintaxes JSON ou YAML. Depois de adicionar etapas e criar o runbook, o sistema converterá as ações adicionadas para o formato YAML que o Systems Manager poderá usar para executar a automação.

Runbooks são compatíveis com o uso de Markdown, uma linguagem de marcação que permite adicionar descrições do estilo wiki a runbooks e etapas individuais dentro do runbook. Para obter mais informações sobre como usar Markdown, consulte Usar Markdown na AWS.

Criação de um runbook usando o Document Builder

Antes de começar

Recomendamos que você leia sobre as diferentes ações que podem ser usadas em um runbook. Para ter mais informações, consulte Referência de ações do Systems Manager Automation.

Para criar um runbook usando o Document Builder

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Documents.
3. Escolha Criar automação.
4. Em Name (Nome), insira um nome descritivo para o runbook.
5. Em Document description (Descrição do documento), forneça a descrição de estilo markdown para o runbook. Você pode fornecer instruções para usar o runbook, as etapas numeradas ou qualquer outro tipo de informação para descrever o runbook. Consulte o texto padrão para obter informações sobre como formatar o conteúdo.

   > **Tip**
   >
   > Alterne entre Hide preview (Ocultar visualização) e Show preview (Exibir visualização) para ver a aparência do conteúdo de descrição à medida que redige.

6. (Opcional) Em Assume role (Função assumida), insira o nome ou ARN de uma função de serviço para executar ações em seu nome. Se você não especificar uma função, o Automation usará as permissões de acesso do usuário que executar a automação.

   > **Important**
   >
   > Para runbooks não pertencentes à Amazon que usam a ação `aws:executeScript`, uma função deverá ser especificada. Para ter mais informações, consulte Permissões para usar runbooks.

7. (Opcional) Em Outputs (Saídas), insira saídas para a automação desse runbook a serem disponibilizadas para outros processos.

   Por exemplo, se o runbook criar uma nova AMI, você poderá especificar ["CreateImage.ImageId"] e usar essa saída para criar novas instâncias em uma automação subsequente.

8. (Opcional) Expanda a seção Input parameters (Parâmetros de entrada) e faça o seguinte.

   1. Em Parameter name (Nome do parâmetro), insira um nome descritivo para o parâmetro do runbook que estiver criando.
   2. Em Type (Tipo), escolha um tipo para o parâmetro, como `String` ou `MapList`.
   3. Em Required (Obrigatório), siga um destes procedimentos:
      - Escolha Yes (Sim) se um valor para esse parâmetro de runbook precisar ser fornecido no runtime.
      - Selecione No (Não) se o parâmetro não for obrigatório e insira um valor de parâmetro padrão (opcional) em Default value (Valor padrão).
   4. Em Description (Descrição), insira uma descrição para o parâmetro do runbook.

   > **Note**
   >
   > Para adicionar mais parâmetros do runbook, escolha Add a parameter (Adicionar um parâmetro). Para remover um parâmetro do runbook, escolha o botão X (Remover).

9. (Opcional) Expanda a seção Target type (Tipo de destino) e escolha um tipo de destino para definir os tipos de recursos nos quais a automação pode ser executada. Por exemplo, para executar um runbook em instâncias do EC2, escolha `/AWS::EC2::Instance`.

   > **Note**
   >
   > Se você especificar um valor de “`/`”, o runbook poderá ser executado em todos os tipos de recursos. Para obter uma lista dos tipos de recursos válidos, consulte Referência de tipos de recursos da AWS no Manual do usuário do AWS CloudFormation.

10. (Opcional) Expanda a seção Document tags (Tags de documento) e insira um ou mais pares de chave-valor para aplicar ao runbook. As tags facilitam a identificação, organização e pesquisa de recursos.
11. Na seção Step 1 (Etapa 1), forneça as seguintes informações.
    - Em Step name (Nome da etapa), insira um nome descritivo para a primeira etapa da automação.
    - Para Action type (Tipo de ação), selecione o tipo de ação a ser usado para esta etapa.

Para obter uma lista e informações sobre os tipos de ação disponíveis, consulte Referência de ações do Systems Manager Automation.

- Em Description (Descrição), insira uma descrição para a etapa de automação. Você pode usar Markdown para formatar o texto.
- Dependendo do Action type (Tipo de ação) selecionado, insira as entradas necessárias para o tipo de ação na seção Step inputs (Entradas da etapa). Por exemplo, se você selecionou a ação `aws:approve`, deverá especificar um valor para a propriedade `Approvers`.

  Para obter informações sobre os campos de entrada de etapa, consulte a entrada em Referência de ações do Systems Manager Automation para o tipo de ação selecionado. Por exemplo: `aws:executeStateMachine` – Executa uma máquina de estado do AWS Step Functions.
- (Opcional) Em Additional inputs (Entradas adicionais), forneça os valores de entrada adicionais necessários para o runbook. Os tipos de entrada disponíveis dependem do tipo de ação selecionado para a etapa. (Observe que alguns tipos de ação exigem valores de entrada.)

  > **Note**
  >
  > Para adicionar mais entradas, escolha Add optional input (Adicionar entrada opcional). Para remover uma entrada, escolha o botão X (Remover).
- (Opcional) Em Outputs (Saídas), insira saídas para essa etapa a serem disponibilizadas para outros processos.

  > **Note**
  >
  > Outputs (Saídas) não está disponível para todos os tipos de ação.
- (Opcional) Expanda a seção Common properties (Propriedades comuns) e especifique as propriedades para as ações comuns a todas as ações de automação. Por exemplo, em Timeout seconds (Segundos de tempo limite), você pode fornecer um valor em segundos para especificar quanto tempo a etapa pode ser executada antes de ser interrompida.

  Para ter mais informações, consulte Propriedades compartilhadas por todas as ações.

> **Note**
>
> Para adicionar mais etapas, selecione Add step (Adicionar etapa) e repita o procedimento de criação de uma etapa. Para remover uma etapa, escolha Remove step (Remover etapa).

12. Escolha Create automation (Criar automação) para salvar o runbook.

Crie um runbook que execute scripts

O procedimento a seguir mostra como usar o Document Builder no console do AWS Systems Manager Automation para criar um runbook personalizado que execute um script.

A primeira etapa do runbook criado executa um script para executar uma instância do Amazon Elastic Compute Cloud (Amazon EC2). A segunda etapa executa outro script para monitorar para que a verificação de status da instância seja alterada `ok`. Depois, um status geral de `Success` é relatado para a automação.

Antes de começar

Você deve ter concluído as etapas a seguir:

- Verifique se você tem privilégios de administrador ou se recebeu as permissões adequadas para acessar o Systems Manager no AWS Identity and Access Management (IAM).

  Para ter mais informações, consulte Verificar o acesso do usuário aos runbooks.
- Verifique se você tem uma função de serviço do IAM para o Automation (também conhecida como uma função assumida) em sua Conta da AWS. A função é necessária pois essa demonstração usa a ação `aws:executeScript`.

  Para obter informações sobre como criar essa função, consulte Configurar o acesso a uma função de serviço (função assumida) para automações.

  Para obter informações sobre o requisito de função de serviço do IAM para a execução do `aws:executeScript`, consulte Permissões para usar runbooks.
- Verifique se você tem permissão para executar instâncias do EC2.

  Para obter informações, consulte IAM e Amazon EC2 no Guia do usuário do Amazon EC2.

Para criar um runbook personalizado usando o Document Builder

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Documents.
3. Escolha Criar automação.
4. Em Name (Nome), digite este nome descritivo para o runbook: **`LaunchInstanceAndCheckStatus`**.
5. (Opcional) Em Document description (Descrição do documento), substitua o texto padrão por uma descrição desse runbook, usando Markdown (Marcação). Veja um exemplo a seguir.

```
##Title: LaunchInstanceAndCheckState
    -----
    **Purpose**: This runbook first launches an EC2 instance using the AMI
 ID provided in the parameter ```imageId```. The second step of this runbook
 continuously checks the instance status check value for the launched instance
 until the status ```ok``` is returned.

    ##Parameters:
    -----
    Name | Type | Description | Default Value
    ------------- | ------------- | ------------- | -------------
    assumeRole | String | (Optional) The ARN of the role that allows Automation to
 perform the actions on your behalf. | -
    imageId  | String | (Optional) The AMI ID to use for launching the instance.
 The default value uses the latest Amazon Linux AMI ID available. | {{ ssm:/aws/
service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-gp2 }}
```

6. Em Assume role (Função assumida) insira o ARN da função de serviço do IAM para o Automation (função assumida) para a execução de automação, no formato **`arn:aws:iam::111122223333:role/AutomationServiceRole`**. Substitua o ID da Conta da AWS por 111122223333.

   A função especificada é usada para fornecer as permissões necessárias para iniciar a automação.

> **Important**
>
> Para runbooks não pertencentes à Amazon que usam a ação `aws:executeScript`, uma função deverá ser especificada. Para ter mais informações, consulte Permissões para usar runbooks.

7. Expanda Input parameters (Parâmetros de entrada) e faça o seguinte.

   1. Em Parameter name (Nome do parâmetro), insira **`imageId`**.
   2. Em Type (Tipo), escolha **`String`**.
   3. Em Required (Obrigatório), escolha No.
   4. Em Default value (Valor padrão), insira o seguinte.

      ```
      {{ ssm:/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-gp2 }}
      ```

      > **Note**
      >
      > Esse valor executa uma instância do Amazon EC2 usando o ID da Amazon Machine Image (AMI) do Amazon Linux 1 mais recente. Se você quiser usar uma AMI diferente, substitua o valor pelo ID da sua AMI.

   5. Em Description (Descrição), insira o seguinte.

      ```
      (Optional) The AMI ID to use for launching the instance. The default value uses
       the latest released Amazon Linux AMI ID.
      ```

8. Escolha Add a parameter (Adicionar um parâmetro) para criar o segundo parâmetro, **`tagValue`**, e insira o seguinte.

   1. Em Parameter name (Nome do parâmetro), insira **`tagValue`**.
   2. Em Type (Tipo), escolha **`String`**.
   3. Em Required (Obrigatório), escolha No.
   4. Em Default value (Valor padrão), insira **`LaunchedBySsmAutomation`**. Isso adiciona o valor `Name:LaunchedBySsmAutomation` do par de chaves da tag à instância.
   5. Em Description (Descrição), insira o seguinte.

```
(Optional) The tag value to add to the instance. The default value is
 LaunchedBySsmAutomation.
```

9. Escolha Add a parameter (Adicionar um parâmetro) para criar o terceiro parâmetro, **`instanceType`**, e insira as informações a seguir.

    1. Em Parameter name (Nome do parâmetro), insira **`instanceType`**.
    2. Em Type (Tipo), escolha **`String`**.
    3. Em Required (Obrigatório), escolha No.
    4. Em Default value (Valor padrão), insira **`t2.micro`**.
    5. Em Parameter description (Descrição do parâmetro), insira o seguinte.

    ```
    (Optional) The instance type to use for the instance. The default value is
     t2.micro.
    ```

10. Expanda Target type (Tipo de destino) e escolha **`"/"`**.
11. (Opcional) Expanda Document tags (Tags do documento) para aplicar tags de recurso ao runbook. Em Tag key (Chave de tag), insira **`Purpose`** e, em Tag value (Valor da tag), insira **`LaunchInstanceAndCheckState`**.
12. Na seção Step 1 (Etapa 1), conclua as etapas a seguir.

    1. Em Step name (Nome da etapa), insira este nome descritivo para a primeira etapa da automação: **`LaunchEc2Instance`**.
    2. Em Action type (Tipo de ação), escolha Run a script (Executar um script) (**`aws:executeScript`**).
    3. Em Description (Descrição), insira uma descrição para a etapa de automação, como a seguinte.

    ```
    **About This Step**

       This step first launches an EC2 instance using the ```aws:executeScript```
     action and the provided script.
    ```

    4. Expanda Inputs (Entradas).
    5. Em Runtime, escolha a linguagem do runtime a ser usada para executar o script fornecido.

6. Em Handler (Manipulador), insira **`launch_instance`**. Esse é o nome da função declarado no script a seguir.

   > **Note**
   >
   > Isso não é necessário para o PowerShell.

7. Em Script, substitua o conteúdo padrão pelo seguinte. Certifique-se de corresponder o script com o valor de runtime correspondente.

   Python

   ```
   def launch_instance(events, context):
         import boto3
         ec2 = boto3.client('ec2')

         image_id = events['image_id']
         tag_value = events['tag_value']
         instance_type = events['instance_type']

         tag_config = {'ResourceType': 'instance', 'Tags': [{'Key':'Name',
    'Value':tag_value}]}

         res = ec2.run_instances(ImageId=image_id, InstanceType=instance_type,
    MaxCount=1, MinCount=1, TagSpecifications=[tag_config])

         instance_id = res['Instances'][0]['InstanceId']

         print('[INFO] 1 EC2 instance is successfully launched', instance_id)

         return { 'InstanceId' : instance_id }
   ```

   PowerShell

   ```
   Install-Module AWS.Tools.EC2 -Force
       Import-Module AWS.Tools.EC2

       $payload = $env:InputPayload | ConvertFrom-Json

       $imageid = $payload.image_id

       $tagvalue = $payload.tag_value
   ```

```
    $instanceType = $payload.instance_type

    $type = New-Object Amazon.EC2.InstanceType -ArgumentList $instanceType

    $resource = New-Object Amazon.EC2.ResourceType -ArgumentList 'instance'

    $tag = @{Key='Name';Value=$tagValue}

    $tagSpecs = New-Object Amazon.EC2.Model.TagSpecification

    $tagSpecs.ResourceType = $resource

    $tagSpecs.Tags.Add($tag)

    $res = New-EC2Instance -ImageId $imageId -MinCount 1 -MaxCount 1 -
InstanceType $type -TagSpecification $tagSpecs

    return @{'InstanceId'=$res.Instances.InstanceId}
```

8. Expanda Additional inputs (Entradas adicionais).
9. Em Input name (Nome da entrada), escolha InputPayload. Em Input value (Valor de entrada), insira os dados YAML a seguir.

```
image_id: "{{ imageId }}"
    tag_value: "{{ tagValue }}"
    instance_type: "{{ instanceType }}"
```

13. Expanda Outputs (Saídas) e faça o seguinte:
    - Em Nome, digite **`payload`**.
    - Em Selector (Seletor), insira **`$.Payload`**.
    - Em Type (Tipo), escolha `StringMap`.
14. Selecione Add step (Adicionar etapa) para adicionar uma segunda etapa ao runbook. A segunda etapa consulta o status da instância executada na Etapa 1 e aguarda até que o status retornado seja `ok`.
15. Na seção Step 2 (Etapa 2), faça o seguinte.
    1. Em Step name (Nome da etapa), insira este nome descritivo para a segunda etapa da automação: **`WaitForInstanceStatusOk`**.

2. Em Action type (Tipo de ação), escolha Run a script (Executar um script) (**`aws:executeScript`**).
3. Em Description (Descrição), insira uma descrição para a etapa de automação, como a seguinte.

```
**About This Step**

   The script continuously polls the instance status check value for the
 instance launched in Step 1 until the ```ok``` status is returned.
```

4. Em Runtime, escolha a linguagem do runtime a ser usada para executar o script fornecido.
5. Em Handler (Manipulador), insira **`poll_instance`**. Esse é o nome da função declarado no script a seguir.

> **Note**
>
> Isso não é necessário para o PowerShell.

6. Em Script, substitua o conteúdo padrão pelo seguinte. Certifique-se de corresponder o script com o valor de runtime correspondente.

   Python

```
def poll_instance(events, context):
      import boto3
      import time

      ec2 = boto3.client('ec2')

      instance_id = events['InstanceId']

      print('[INFO] Waiting for instance status check to report ok',
 instance_id)

      instance_status = "null"

      while True:
        res = ec2.describe_instance_status(InstanceIds=[instance_id])

        if len(res['InstanceStatuses']) == 0:
          print("Instance status information is not available yet")
```

```
          time.sleep(5)
          continue

        instance_status = res['InstanceStatuses'][0]['InstanceStatus']
['Status']

        print('[INFO] Polling to get status of the instance', instance_status)

        if instance_status == 'ok':
          break

        time.sleep(10)

      return {'Status': instance_status, 'InstanceId': instance_id}
```

PowerShell

```
    Install-Module AWS.Tools.EC2 -Force

    $inputPayload = $env:InputPayload | ConvertFrom-Json

    $instanceId = $inputPayload.payload.InstanceId

    $status = Get-EC2InstanceStatus -InstanceId $instanceId

    while ($status.Status.Status -ne 'ok'){
       Write-Host 'Polling get status of the instance', $instanceId

       Start-Sleep -Seconds 5

       $status = Get-EC2InstanceStatus -InstanceId $instanceId
    }

    return @{Status = $status.Status.Status; InstanceId = $instanceId}
```

7. Expanda Additional inputs (Entradas adicionais).
8. Em Input name (Nome da entrada), escolha InputPayload. Em Input value (Valor de entrada), insira o seguinte:

```
{{ LaunchEc2Instance.payload }}
```

16. Escolha Create automation (Criar automação) para salvar o runbook.

Uso de scripts em runbooks

Runbooks do Automation oferecem suporte à execução de scripts como parte da automação. O Automation é uma ferramenta do AWS Systems Manager. Usando runbooks, você pode executar scripts diretamente na AWS sem precisar criar um ambiente de computação separado para executar os scripts. Como os documentos de automação podem executar etapas de script junto com outros tipos de etapas de automação, como aprovações, você pode intervir manualmente em situações críticas ou ambíguas. Você pode enviar o resultados das ações do `aws:executeScript` nos runbooks para o Amazon CloudWatch Logs. Para ter mais informações, consulte Registro de saída de ações do Automation em log com o CloudWatch Logs.

Permissões para usar runbooks

Para usar um runbook, o Systems Manager deve usar as permissões de uma função do AWS Identity and Access Management (IAM). O método usado pelo Automation para determinar quais permissões da função usar depende de alguns fatores e se uma etapa usa a ação `aws:executeScript`.

Para runbooks que não usam o `aws:executeScript`, o Automation usa uma das duas fontes de permissões:

- As permissões de uma função de serviço do IAM, ou função assumida, que é especificada no runbook ou transmitida como um parâmetro.
- Se nenhum perfil de serviço do IAM estiver especificado, as permissões do usuário que iniciou a automação.

Quando uma etapa em um documento do runbook incluir a ação `aws:executeScript`, uma função de serviço do IAM (função assumida) sempre será necessária se o script do Python ou PowerShell especificado para a ação estiver chamando quaisquer ações da API da AWS. A automação verifica a existência dessa função na seguinte ordem:

- As permissões de uma função de serviço do IAM, ou função assumida, que é especificada no runbook ou transmitida como um parâmetro.
- Se nenhuma função for encontrada, o Automation tentará executar o script do Python ou PowerShell especificado para `aws:executeScript` sem permissões. Se o script estiver chamando uma operação de API da AWS (por exemplo, a operação Amazon EC2 `CreateImage`),

ou tentando agir em um recurso da AWS (como uma instância do EC2), a etapa que contém o script falhará e o Systems Manager retornará uma mensagem de erro relatando a falha.

Adicionar scripts a runbooks

Você pode adicionar scripts aos runbooks, incluindo o script em linha como parte de uma etapa no runbook. Também é possível anexar scripts ao runbook fazendo upload dos scripts de sua máquina local ou especificando um bucket do Amazon Simple Storage Service (Amazon S3) onde os scripts estão localizados. Após a conclusão de uma etapa que executa um script, o resultado do script estará disponível como um objeto JSON, que poderá ser usado como entrada para etapas subsequentes no runbook. Para obter mais informações sobre a ação `aws:executeScript` e como usar anexos para scripts, consulte `aws:executeScript` – Executa um script.

Restrições de script para runbooks

Runbooks impõem um limite de cinco anexos do arquivo. Os scripts podem ser na forma de um script do Python (.py), um script do PowerShell Core (.ps1) ou anexados como conteúdo em um arquivo .zip.

Uso de instruções condicionais em runbooks

Por padrão, as etapas que você define na seção `mainSteps` de um runbook são executadas em ordem sequencial. Depois que uma ação é concluída, a próxima ação especificada na seção `mainSteps` começa. Além disso, se uma ação falhar, todo a automação falhará (por padrão). Você pode usar a ação de automação `aws:branch` e as opções do runbook descritas nesta seção para criar automações que executam ramificações condicionais. Isso significa que você pode criar fluxos automações que vão para outra etapa após avaliar diferentes escolhas ou que dinamicamente respondem a alterações quando uma etapa é concluída. Veja a seguir uma lista de opções que você pode usar para criar fluxos de trabalho de automação dinâmicos:

- **`aws:branch`**: esta ação de automação permite que você crie a automação dinâmica que avalia várias opções em uma única etapa e, em seguida, salta para outra etapa no runbook com base nos resultados da avaliação.
- **`nextStep`**: esta opção especifica qual etapa de uma automação deve ser processada imediatamente após a conclusão bem-sucedida de uma etapa.
- **`isEnd`**: esta opção interrompe a execução de uma automação no final de determinada etapa. O valor padrão desta opção é falso.

- **`isCritical`**: esta opção designa uma etapa como essencial para a conclusão bem-sucedida da automação. Se uma etapa com essa designação falhar, o Automation relatará o status final da automação como `Failed`. O valor padrão desta opção é `true`.
- **`onFailure`**: esta opção indica se a automação deve ser anulada, deve continuar ou seguir para outra etapa, em caso de falha. O valor padrão desta opção é anular.

A seção a seguir descreve a ação de automação `aws:branch`. Para obter mais informações sobre as opções do `nextStep`, `isEnd`, `isCritical` e `onFailure`, consulte Runbooks `aws:branch` de exemplo.

Trabalhar com a ação **`aws:branch`**

A ação `aws:branch` oferece as opções de ramificação condicional mais dinâmicas para automações. Como mencionado anteriormente, essa ação permite que a automação avalie várias condições em uma única etapa e depois vá para uma nova etapa com base nos resultados da avaliação. A ação `aws:branch` funciona como uma instrução `IF-ELIF-ELSE` na programação.

Veja a seguir um exemplo de YAML de uma etapa `aws:branch`:

```
- name: ChooseOSforCommands
  action: aws:branch
  inputs:
    Choices:
    - NextStep: runPowerShellCommand
      Variable: "{{GetInstance.platform}}"
      StringEquals: Windows
    - NextStep: runShellCommand
      Variable: "{{GetInstance.platform}}"
      StringEquals: Linux
    Default:
      PostProcessing
```

Quando você especifica a ação `aws:branch` para uma etapa, você especifica as `Choices` que a automação deve avaliar. A automação pode avaliar as `Choices` com base no valor de um parâmetro especificado na seção `Parameters` do runbook. A automação também pode avaliar as `Choices` com base no resultado de uma etapa anterior.

A automação avalia cada opção usando uma expressão booleana. Se a avaliação determinar que a primeira opção é `true`, a automação pulará para a etapa designada para essa opção. Se a avaliação determinar que a primeira opção é `false`, a automação avaliará a próxima opção. Se a

sua etapa inclui três `Choices` ou mais, a automação avaliará cada opção em ordem sequencial até avaliar uma opção que seja `true`. A automação pula para a etapa designada para a opção `true`.

Se nenhuma das `Choices` for `true`, a automação verificará se a etapa contém um valor `Default`. Um valor `Default` define uma etapa para a qual a automação deve saltar, se nenhuma das opções for `true`. Se nenhum valor `Default` for especificado para a etapa, a automação processará a próxima etapa no runbook.

Aqui está uma etapa `aws:branch` em YAML chamada chooseOSfromParameter. A etapa inclui duas `Choices`: (`NextStep: runWindowsCommand`) e (`NextStep: runLinuxCommand`). A automação avalia essas `Choices` para determinar qual comando deve ser executado para o sistema operacional apropriado. A `Variable` para cada opção usa `{{OSName}}`, que é um parâmetro que o autor do runbook definiu na seção `Parameters` do runbook.

```
mainSteps:
- name: chooseOSfromParameter
  action: aws:branch
  inputs:
    Choices:
    - NextStep: runWindowsCommand
      Variable: "{{OSName}}"
      StringEquals: Windows
    - NextStep: runLinuxCommand
      Variable: "{{OSName}}"
      StringEquals: Linux
```

Aqui está uma etapa `aws:branch` em YAML chamada chooseOSfromOutput. A etapa inclui duas `Choices`: (`NextStep: runPowerShellCommand`) e (`NextStep: runShellCommand`). A automação avalia essas `Choices` para determinar qual comando deve ser executado para o sistema operacional apropriado. A `Variable` para cada opção usa `{{GetInstance.platform}}`, que é o resultado de uma etapa anterior no runbook. Este exemplo também inclui uma opção chamada `Default`. Se a automação avaliar as duas `Choices`, e nenhuma opção for `true`, a automação pulará para uma etapa chamada `PostProcessing`.

```
mainSteps:
- name: chooseOSfromOutput
  action: aws:branch
  inputs:
    Choices:
```

```
    - NextStep: runPowerShellCommand
      Variable: "{{GetInstance.platform}}"
      StringEquals: Windows
    - NextStep: runShellCommand
      Variable: "{{GetInstance.platform}}"
      StringEquals: Linux
    Default:
      PostProcessing
```

Criar uma etapa **aws:branch** em um runbook

Quando você cria uma etapa `aws:branch` em um runbook, você define as `Choices` que a automação deve avaliar para determinar a etapa que a automação deve passar em seguida. Como mencionado anteriormente, `Choices` são avaliadas usando uma expressão booliana. Cada opção deve definir as seguintes opções:

- NextStep: a próxima etapa no runbook para processar se a opção designada é `true`.
- Variável: especifique o nome de um parâmetro definido na seção `Parameters` do runbook, uma variável definida na seção `Variables` ou especifique um objeto de saída de uma etapa anterior.

  Especifique valores de variáveis usando o formato a seguir.

  `Variable: "{{variable name}}"`

  Especifique valores de parâmetros usando o formato a seguir.

  `Variable: "{{parameter name}}"`

  Especifique variáveis de objetos de saída usando o seguinte formato:

  `Variable: "{{previousStepName.outputName}}"`

  > **Note**
  >
  > A criação da variável de saída é descrita em mais detalhes na próxima seção, Sobre a criação de variáveis de saída.

- Operação: os critérios usados para avaliar a escolha, como `StringEquals: Linux`. A ação `aws:branch` oferece suporte às seguintes operações:

Operações de string

- StringEquals
- EqualsIgnoreCase
- StartsWith
- EndsWith
- Contém

Operações numéricas

- NumericEquals
- NumericGreater
- NumericLesser
- NumericGreaterOrEquals
- NumericLesser
- NumericLesserOrEquals

Operação booleana

- BooleanEquals

> **Important**
>
> Quando você cria um runbook, o sistema valida cada operação dele. Se uma operação não for suportada, o sistema retornará um erro ao tentar criar o runbook.

- Padrão: especifique uma etapa de contingência para a qual a automação deve ir se nenhuma das `Choices` for `true`.

> **Note**
>
> Se você não quiser especificar um valor `Default`, especifique a opção `isEnd`. Se nenhuma das `Choices` for `true` e nenhum valor `Default` for especificado, a automação será interrompida no final da etapa.

Use os modelos a seguir para ajudar a construir a etapa `aws:branch` no runbook. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

YAML

```
mainSteps:
- name: step name
  action: aws:branch
  inputs:
    Choices:
    - NextStep: step to jump to if evaluation for this choice is true
      Variable: "{{parameter name or output from previous step}}"
      Operation type: Operation value
    - NextStep: step to jump to if evaluation for this choice is true
      Variable: "{{parameter name or output from previous step}}"
      Operation type: Operation value
    Default:
      step to jump to if all choices are false
```

JSON

```
{
   "mainSteps":[
      {
         "name":"a name for the step",
         "action":"aws:branch",
         "inputs":{
            "Choices":[
               {
                  "NextStep":"step to jump to if evaluation for this choice is
 true",
                  "Variable":"{{parameter name or output from previous step}}",
                  "Operation type":"Operation value"
               },
               {
                  "NextStep":"step to jump to if evaluation for this choice is
 true",
                  "Variable":"{{parameter name or output from previous step}}",
                  "Operation type":"Operation value"
               }
            ],
            "Default":"step to jump to if all choices are false"
         }
```

```
      }
    ]
}
```

Sobre a criação de variáveis de saída

Para criar uma opção `aws:branch` que faz referência à saída de uma etapa anterior, você precisa identificar o nome da etapa anterior e o nome do campo de saída. Você então combina os nomes da etapa e o campo usando o seguinte formato:

`Variable: "{{previousStepName.outputName}}"`

Por exemplo, a primeira etapa do exemplo a seguir é chamada de `GetInstance`. Depois, em `outputs`, há um campo chamado `platform`. Na segunda etapa (`ChooseOSforCommands`), o autor deseja fazer referência à saída do campo de plataforma como uma variável. Para criar a variável, basta combinar o nome da etapa (GetInstance) e o nome do campo de saída (plataforma) para criar `Variable: "{{GetInstance.platform}}"`.

```
mainSteps:
- Name: GetInstance
  action: aws:executeAwsApi
  inputs:
    Service: ssm
    Api: DescribeInstanceInformation
    Filters:
    - Key: InstanceIds
      Values: ["{{ InstanceId }}"]
  outputs:
  - Name: myInstance
    Selector: "$.InstanceInformationList[0].InstanceId"
    Type: String
  - Name: platform
    Selector: "$.InstanceInformationList[0].PlatformType"
    Type: String
- name: ChooseOSforCommands
  action: aws:branch
  inputs:
    Choices:
    - NextStep: runPowerShellCommand
      Variable: "{{GetInstance.platform}}"
      StringEquals: Windows
```

```
    - NextStep: runShellCommand
      Variable: "{{GetInstance.platform}}"
      StringEquals: Linux
    Default:
      Sleep
```

Aqui está um exemplo que mostra como *"Variable": "{{describeInstance.Platform}}"* é criado na etapa anterior e na saída.

```
- name: describeInstance
  action: aws:executeAwsApi
  onFailure: Abort
  inputs:
    Service: ec2
    Api: DescribeInstances
    InstanceIds:
    - "{{ InstanceId }}"
  outputs:
  - Name: Platform
    Selector: "$.Reservations[0].Instances[0].Platform"
    Type: String
  nextStep: branchOnInstancePlatform
- name: branchOnInstancePlatform
  action: aws:branch
  inputs:
    Choices:
    - NextStep: runEC2RescueForWindows
      Variable: "{{ describeInstance.Platform }}"
      StringEquals: windows
    Default: runEC2RescueForLinux
```

Runbooks **`aws:branch`** de exemplo

Aqui estão alguns exemplos de runbooks que usam o `aws:branch`.

Exemplo 1: usar **`aws:branch`** com uma variável de saída para executar comandos com base no tipo de sistema operacional

Na primeira etapa deste exemplo (`GetInstance`), o autor do runbook usa a ação `aws:executeAwsApi` para chamar a operação de API `ssm DescribeInstanceInformation`. O autor usa essa ação para determinar o tipo de sistema operacional que está sendo usado por uma instância. A ação `aws:executeAwsApi` emite o ID da instância e o tipo de plataforma.

Na segunda etapa (`ChooseOSforCommands`), o autor usa a ação `aws:branch` com duas `Choices` (`NextStep: runPowerShellCommand`) e (`NextStep: runShellCommand`). A automação avalia o sistema operacional da instância usando o resultado da etapa anterior (`Variable: "{{GetInstance.platform}}"`). A automação pula para uma etapa do sistema operacional designado.

```
---
schemaVersion: '0.3'
assumeRole: "{{AutomationAssumeRole}}"
parameters:
  AutomationAssumeRole:
    default: ""
    type: String
mainSteps:
- name: GetInstance
  action: aws:executeAwsApi
  inputs:
    Service: ssm
    Api: DescribeInstanceInformation
  outputs:
  - Name: myInstance
    Selector: "$.InstanceInformationList[0].InstanceId"
    Type: String
  - Name: platform
    Selector: "$.InstanceInformationList[0].PlatformType"
    Type: String
- name: ChooseOSforCommands
  action: aws:branch
  inputs:
    Choices:
    - NextStep: runPowerShellCommand
      Variable: "{{GetInstance.platform}}"
      StringEquals: Windows
    - NextStep: runShellCommand
      Variable: "{{GetInstance.platform}}"
      StringEquals: Linux
    Default:
      Sleep
- name: runShellCommand
  action: aws:runCommand
  inputs:
    DocumentName: AWS-RunShellScript
    InstanceIds:
```

```
    - "{{GetInstance.myInstance}}"
    Parameters:
      commands:
      - ls
  isEnd: true
- name: runPowerShellCommand
  action: aws:runCommand
  inputs:
    DocumentName: AWS-RunPowerShellScript
    InstanceIds:
    - "{{GetInstance.myInstance}}"
    Parameters:
      commands:
      - ls
  isEnd: true
- name: Sleep
  action: aws:sleep
  inputs:
    Duration: PT3S
```

Exemplo 2: usar **`aws:branch`** com uma variável de parâmetro para executar comandos com base no tipo de sistema operacional

O autor do runbook define várias opções de parâmetro no início do runbook, na seção `parameters`. Um parâmetro é chamado de `OperatingSystemName`. Na primeira etapa (`ChooseOS`), o autor usa a ação `aws:branch` com duas `Choices` (`NextStep: runWindowsCommand`) e (`NextStep: runLinuxCommand`). A variável para essas `Choices` faz referência à opção de parâmetro especificada na seção de parâmetros (`Variable: "{{OperatingSystemName}}"`). Quando o usuário executa esse runbook, ele especifica um valor em runtime para `OperatingSystemName`. A automação usa o parâmetro de runtime durante a avaliação das `Choices`. A automação pula para uma etapa do sistema operacional designado com base no parâmetro do runtime para `OperatingSystemName`.

```
---
schemaVersion: '0.3'
assumeRole: "{{AutomationAssumeRole}}"
parameters:
  AutomationAssumeRole:
    default: ""
    type: String
  OperatingSystemName:
    type: String
```

```
  LinuxInstanceId:
    type: String
  WindowsInstanceId:
    type: String
mainSteps:
- name: ChooseOS
  action: aws:branch
  inputs:
    Choices:
    - NextStep: runWindowsCommand
      Variable: "{{OperatingSystemName}}"
      StringEquals: windows
    - NextStep: runLinuxCommand
      Variable: "{{OperatingSystemName}}"
      StringEquals: linux
    Default:
      Sleep
- name: runLinuxCommand
  action: aws:runCommand
  inputs:
    DocumentName: "AWS-RunShellScript"
    InstanceIds:
    - "{{LinuxInstanceId}}"
    Parameters:
      commands:
      - ls
  isEnd: true
- name: runWindowsCommand
  action: aws:runCommand
  inputs:
    DocumentName: "AWS-RunPowerShellScript"
    InstanceIds:
    - "{{WindowsInstanceId}}"
    Parameters:
      commands:
      - date
  isEnd: true
- name: Sleep
  action: aws:sleep
  inputs:
    Duration: PT3S
```

Criar automações de ramificação complexas com operadores

Você pode criar automações de ramificação complexas usando os operadores `And`, `Or`, e `Not` em suas etapas `aws:branch`.

O operador 'And' (‘E’)

Use o operador `And` quando você quiser que muitas variáveis sejam `true` para uma opção. No exemplo a seguir, a primeira opção avalia se uma instância está `running` e usa o sistema operacional `Windows`. Se a avaliação de ambas as variáveis for verdadeira, a automação pulará para a etapa `runPowerShellCommand`. Se uma ou mais das variáveis for `false`, a automação avaliará as variáveis para a segunda opção.

```
mainSteps:
- name: switch2
  action: aws:branch
  inputs:
    Choices:
    - And:
      - Variable: "{{GetInstance.pingStatus}}"
        StringEquals: running
      - Variable: "{{GetInstance.platform}}"
        StringEquals: Windows
      NextStep: runPowerShellCommand

    - And:
      - Variable: "{{GetInstance.pingStatus}}"
        StringEquals: running
      - Variable: "{{GetInstance.platform}}"
        StringEquals: Linux
      NextStep: runShellCommand
    Default:
      sleep3
```

O operador 'Or' (‘Ou’)

Use o operador `Or` quando você quiser que qualquer de diversas variáveis sejam verdadeiras para uma opção. No exemplo a seguir, a primeira opção avalia se uma string de parâmetro é `Windows` e se a saída de uma etapa AWS Lambda é verdadeira. Se a avaliação determinar que qualquer uma das variáveis é verdadeira, a automação pulará para a etapa `RunPowerShellCommand`. Se as duas variáveis forem falsas, a automação avaliará as variáveis para a segunda opção.

```
- Or:
  - Variable: "{{parameter1}}"
    StringEquals: Windows
  - Variable: "{{BooleanParam1}}"
    BooleanEquals: true
  NextStep: RunPowershellCommand
- Or:
  - Variable: "{{parameter2}}"
    StringEquals: Linux
  - Variable: "{{BooleanParam2}}"
    BooleanEquals: true
  NextStep: RunShellScript
```

O operador 'Not' (‘Não’)

Use o operador `Not` quando você desejar pular para uma etapa definida quando uma variável não for verdadeira. No exemplo a seguir, a primeira opção avalia se uma string de parâmetro é `Not Linux`. Se a avaliação determinar que a variável não é Linux, a automação pulará para a etapa `sleep2`. Se a avaliação da primeira opção determinar que ela é Linux, a automação avaliará a próxima opção.

```
mainSteps:
- name: switch
  action: aws:branch
  inputs:
    Choices:
    - NextStep: sleep2
      Not:
        Variable: "{{testParam}}"
        StringEquals: Linux
    - NextStep: sleep1
      Variable: "{{testParam}}"
      StringEquals: Windows
    Default:
      sleep3
```

Exemplos de como usar opções condicionais

Esta seção inclui vários exemplos de como usar as opções dinâmicas em um runbook. Cada exemplo nesta seção estende o seguinte runbook: Este runbook tem duas ações. A primeira ação é chamada de `InstallMsiPackage`. Ela usa a ação `aws:runCommand` para instalar um

aplicativo em uma instância do Windows Server. A segunda ação é chamada de `TestInstall`. Ela usa a ação `aws:invokeLambdaFunction` para executar um teste do aplicativo instalado, se o aplicativo foi instalado com êxito. A etapa um especifica `onFailure: Abort`. Isso significa que, se a instalação da aplicação não foi bem-sucedida, a automação será interrompida antes da segunda etapa.

Exemplo 1: runbook com duas ações lineares

```
---
schemaVersion: '0.3'
description: Install MSI package and run validation.
assumeRole: "{{automationAssumeRole}}"
parameters:
  automationAssumeRole:
    type: String
    description: "(Required) Assume role."
  packageName:
    type: String
    description: "(Required) MSI package to be installed."
  instanceIds:
    type: String
    description: "(Required) Comma separated list of instances."
mainSteps:
- name: InstallMsiPackage
  action: aws:runCommand
  maxAttempts: 2
  onFailure: Abort
  inputs:
    InstanceIds:
    - "{{instanceIds}}"
    DocumentName: AWS-RunPowerShellScript
    Parameters:
      commands:
      - msiexec /i {{packageName}}
- name: TestInstall
  action: aws:invokeLambdaFunction
  maxAttempts: 1
  timeoutSeconds: 500
  inputs:
    FunctionName: TestLambdaFunction
...
```

Criar uma automação dinâmica que pula para etapas diferentes usando a opção **`onFailure`**

O exemplo a seguir usa as opções `onFailure: step:`*`step name`*, `nextStep` e `isEnd` para criar uma automação dinâmica. Nesse exemplo, se a ação `InstallMsiPackage` falhar, a automação saltará para uma ação chamada PostFailure (`onFailure: step:PostFailure`) para executar uma função do AWS Lambda, a fim de realizar alguma ação no evento em que a instalação falhou. Se a instalação for bem-sucedida, o processo de automação pulará para a ação TestInstall (`nextStep: TestInstall`). Ambas as etapas `TestInstall` e `PostFailure` usam a opção `isEnd` (`isEnd: true`) para que a automação termine quando uma dessas etapas for concluída.

> **Note**
>
> O uso da opção `isEnd` na última etapa da seção `mainSteps` é opcional. Se a última etapa não pular para outras etapas, a automação será interrompida depois de executar a ação na última etapa.

Exemplo 2: automação dinâmica que salta para diferentes etapas

```
mainSteps
- name: InstallMsiPackage
  action: aws:runCommand
  onFailure: step:PostFailure
  maxAttempts: 2
  inputs:
    InstanceIds:
    - "{{instanceIds}}"
    DocumentName: AWS-RunPowerShellScript
    Parameters:
      commands:
      - msiexec /i {{packageName}}
  nextStep: TestInstall
- name: TestInstall
  action: aws:invokeLambdaFunction
  maxAttempts: 1
  timeoutSeconds: 500
  inputs:
    FunctionName: TestLambdaFunction
  isEnd: true
- name: PostFailure
  action: aws:invokeLambdaFunction
  maxAttempts: 1
  timeoutSeconds: 500
```

```
  inputs:
    FunctionName: PostFailureRecoveryLambdaFunction
  isEnd: true
...
```

> **Note**
>
> Antes de processar um runbook, o sistema verifica se o runbook não cria um loop infinito. Se um loop infinito for detectado, a automação retornará um erro e um círculo de rastreamento mostrando as etapas que criam o loop.

Criar uma automação dinâmica que define etapas essenciais

Você pode especificar que uma etapa é essencial para o sucesso geral da automação. Se uma etapa essencial falhar, o Automation informará o status da automação como `Failed`, mesmo se uma ou mais etapas forem executadas com êxito. No exemplo a seguir, o usuário identifica a etapa VerifyDependencies se a etapa InstallMsiPackage falhar (`onFailure: step:VerifyDependencies`). O usuário especifica que a etapa `InstallMsiPackage` não é essencial (`isCritical: false`). Neste exemplo, se o aplicativo falhar ao instalar, a automação processará a etapa `VerifyDependencies` para determinar se uma ou mais dependências estão ausentes, o que deve ter causado a falha na instalação do aplicativo.

Exemplo 3: definir as etapas essenciais para a automação

```
---
name: InstallMsiPackage
action: aws:runCommand
onFailure: step:VerifyDependencies
isCritical: false
maxAttempts: 2
inputs:
  InstanceIds:
  - "{{instanceIds}}"
  DocumentName: AWS-RunPowerShellScript
  Parameters:
    commands:
    - msiexec /i {{packageName}}
nextStep: TestPackage
...
```

Uso de saídas de ações como entradas

Várias ações de automação retornam saídas predefinidas. Você pode passar essas saídas como entradas para etapas posteriores em seu runbook usando o formato `{{stepName.outputName}}`. Também é possível definir saídas personalizadas para várias ações de automação em seus runbooks. Isso permite que você execute scripts ou invoque operações de API para outros Serviços da AWS uma vez, para que você possa reutilizar os valores como entradas em ações posteriores. Os tipos de parâmetro em runbooks são estáticos. Isso significa que o tipo de parâmetro não pode ser alterado depois de definido. Para definir uma saída de etapa, forneça os seguintes campos:

- Nome (obrigatório): o nome da saída usado para referenciar o valor de saída em etapas posteriores.
- Seletor (obrigatório): a expressão JSONPath usada para determinar o valor de saída.
- Tipo (opcional): o tipo de dados do valor retornado pelo campo seletor. Os valores de tipo válidos são `String`, `Integer`, `Boolean`, `StringList` ,`StringMap`, `MapList`. O valor padrão é `String`.

Se o valor de uma saída não corresponder ao tipo de dados que você especificou, o Automation tentará converter o tipo de dados. Por exemplo, se o valor retornado for um `Integer`, mas o `Type` especificado for `String`, o valor final da saída será um valor `String`. As seguintes conversões de tipos são permitidas:

- Valores `String` podem ser convertidos em `StringList`, `Integer` e `Boolean`.
- Valores `Integer` podem ser convertidos em `String` e `StringList`.
- Valores `Boolean` podem ser convertidos em `String` e `StringList`.
- Valores `StringList`, `IntegerList` ou `BooleanList` contendo um elemento podem ser convertidos em `String`, `Integer` ou `Boolean`.

Ao usar parâmetros ou saídas com ações de automação, o tipo de dados não pode ser alterado dinamicamente na entrada de uma ação.

Aqui está um exemplo de runbook que demonstra como definir saídas de ações e como referenciar o valor como entrada para uma ação posterior. Os runbooks fazem o seguinte:

- Usa a ação `aws:executeAwsApi` para chamar a operação de API do Amazon EC2 DescribeImages para obter o nome de uma determinada AMI do Windows Server 2016. Ela gera o ID de imagem como `ImageId`.

- Usa a ação `aws:executeAwsApi` para chamar a operação de API do Amazon EC2 RunInstances para iniciar uma instância que usa o `ImageId` da etapa anterior. Ela gera o ID de instância como `InstanceId`.
- Usa a ação `aws:waitForAwsResourceProperty` para sondar a operação de API do Amazon EC2 DescribeInstanceStatus para aguardar que a instância alcance o estado `running`. A ação atinge o tempo limite em 60 segundos. A etapa atinge o limite de tempo se o estado da instância não chegar a `running` após 60 segundos de sondagem.
- Usa a ação `aws:assertAwsResourceProperty` para chamar a operação da API `DescribeInstanceStatus` do Amazon EC2 para afirmar que a instância está no estado `running`. A etapa falhará se o estado da instância não for `running`.

```
---
description: Sample runbook using AWS API operations
schemaVersion: '0.3'
assumeRole: "{{ AutomationAssumeRole }}"
parameters:
  AutomationAssumeRole:
    type: String
    description: "(Optional) The ARN of the role that allows Automation to perform the
 actions on your behalf."
    default: ''
  ImageName:
    type: String
    description: "(Optional) Image Name to launch EC2 instance with."
    default: "Windows_Server-2022-English-Full-Base*"
mainSteps:
- name: getImageId
  action: aws:executeAwsApi
  inputs:
    Service: ec2
    Api: DescribeImages
    Filters:
    - Name: "name"
      Values:
      - "{{ ImageName }}"
  outputs:
  - Name: ImageId
    Selector: "$.Images[0].ImageId"
    Type: "String"
- name: launchOneInstance
```

```
  action: aws:executeAwsApi
  inputs:
    Service: ec2
    Api: RunInstances
    ImageId: "{{ getImageId.ImageId }}"
    MaxCount: 1
    MinCount: 1
  outputs:
  - Name: InstanceId
    Selector: "$.Instances[0].InstanceId"
    Type: "String"
- name: waitUntilInstanceStateRunning
  action: aws:waitForAwsResourceProperty
  timeoutSeconds: 60
  inputs:
    Service: ec2
    Api: DescribeInstanceStatus
    InstanceIds:
    - "{{ launchOneInstance.InstanceId }}"
    PropertySelector: "$.InstanceStatuses[0].InstanceState.Name"
    DesiredValues:
    - running
- name: assertInstanceStateRunning
  action: aws:assertAwsResourceProperty
  inputs:
    Service: ec2
    Api: DescribeInstanceStatus
    InstanceIds:
    - "{{ launchOneInstance.InstanceId }}"
    PropertySelector: "$.InstanceStatuses[0].InstanceState.Name"
    DesiredValues:
    - running
outputs:
- "launchOneInstance.InstanceId"
...
```

Cada uma das ações de automação descritas anteriormente permite que você chame uma operação de API específica, determinando o namespace do serviço, o nome da operação de API, os parâmetros de entrada e os parâmetros de saída. As entradas são definidas pela operação de API que você escolher. Você pode visualizar as operações de API (também chamadas de métodos), escolhendo um serviço na navegação à esquerda na seguinte página de Referência de serviços: Escolha um método na seção Client (Cliente) para o serviço que você deseja invocar. Por exemplo,

todas as operações de API (métodos) do Amazon Relational Database Service (Amazon RDS) estão listadas na seguinte página: Amazon RDS methods (Métodos do Amazon RDS).

Você pode visualizar o esquema para cada ação de automação nos seguintes locais:

- `aws:assertAwsResourceProperty`: define um estado do recurso da AWS ou o estado do evento
- `aws:executeAwsApi`: chama e executa as operações de API do AWS
- `aws:waitForAwsResourceProperty`: aguarde uma propriedade de recurso da AWS

Os esquemas incluem as descrições dos campos obrigatórios para o uso de cada ação.

Usar os campos Selector/PropertySelector (Seletor/Seletor de propriedades)

Cada ação do Automation exige que você especifique uma saída `Selector` (para`aws:executeAwsApi`) ou um `PropertySelector` (para `aws:assertAwsResourceProperty` e `aws:waitForAwsResourceProperty`). Esses campos são usados para processar a resposta do JSON de uma operação de API da AWS. Esses campos usam a sintaxe JSONPath.

Este é um exemplo para ajudar a ilustrar esse conceito para a ação `aws:executeAwsAPi`:

```
---
mainSteps:
- name: getImageId
  action: aws:executeAwsApi
  inputs:
    Service: ec2
    Api: DescribeImages
    Filters:
      - Name: "name"
        Values:
          - "{{ ImageName }}"
  outputs:
    - Name: ImageId
      Selector: "$.Images[0].ImageId"
      Type: "String"
...
```

Na etapa `aws:executeAwsApi` do `getImageId`, a automação invoca a operação da API `DescribeImages` e recebe uma resposta do `ec2`. A automação então aplica `Selector -`

"$.Images[0].ImageId" à resposta da API e atribui o valor selecionado à variável ImageId do resultado. Outras etapas na mesma automação podem usar o valor de ImageId especificando "{{ getImageId.ImageId }}".

Este é um exemplo para ajudar a ilustrar esse conceito para a ação aws:waitForAwsResourceProperty:

```
---
- name: waitUntilInstanceStateRunning
  action: aws:waitForAwsResourceProperty
  # timeout is strongly encouraged for action - aws:waitForAwsResourceProperty
  timeoutSeconds: 60
  inputs:
    Service: ec2
    Api: DescribeInstanceStatus
    InstanceIds:
    - "{{ launchOneInstance.InstanceId }}"
    PropertySelector: "$.InstanceStatuses[0].InstanceState.Name"
    DesiredValues:
    - running
...
```

Na etapa aws:waitForAwsResourceProperty do waitUntilInstanceStateRunning, a automação invoca a operação da API DescribeInstanceStatus e recebe uma resposta do ec2. Depois, a automação aplica PropertySelector - "$.InstanceStatuses[0].InstanceState.Name" à resposta e verifica se o valor retornado corresponde a um valor na lista DesiredValues (neste caso, running). A etapa repetirá o processo até que a resposta retorne um estado de instância de running.

Uso de JSONPath em runbooks

Uma expressão JSONPath é uma string que começa com “$”. que é usado para selecionar um ou mais componentes em um elemento JSON. A lista a seguir inclui informações sobre operadores JSONPath que são compatíveis com o Systems Manager Automation:

- Filho com notação de pontos (.): use com um objeto JSON. Esse operador seleciona o valor de uma chave específica.
- Varredura profunda (..): use com um elemento JSON. Esse operador examina o elemento JSON nível por nível e seleciona uma lista de valores com a chave específica. O tipo de retorno desse

operador é sempre uma matriz JSON. No contexto de um tipo de saída da etapa de automação, o operador pode ser StringList ou MapList.

- Índice de matriz ([ ]): use com uma matriz JSON. Esse operador obtém o valor de um índice específico.
- Filtro ([?(***expression***)]): use com uma matriz JSON. Esse operador filtra os valores da matriz JSON que correspondem aos critérios definidos na expressão do filtro. As expressões de filtro só podem usar os seguintes operadores: ==, !=, >, <, >= ou <=. A combinação de várias expressões de filtro com AND (&&) ou OR (||) não é possível. O tipo de retorno desse operador é sempre uma matriz JSON.

Para compreender melhor os operadores JSONPath, revise a seguinte resposta do JSON da operação de API `DescribeInstances` do ec2. Abaixo dessa resposta estão vários exemplos que mostram resultados diferentes aplicando diferentes expressões JSONPath à com base na operação da API `DescribeInstances`.

```
{
    "NextToken": "abcdefg",
    "Reservations": [
        {
            "OwnerId": "123456789012",
            "ReservationId": "r-abcd12345678910",
            "Instances": [
                {
                    "ImageId": "ami-12345678",
                    "BlockDeviceMappings": [
                        {
                            "Ebs": {
                                "DeleteOnTermination": true,
                                "Status": "attached",
                                "VolumeId": "vol-000000000000"
                            },
                            "DeviceName": "/dev/xvda"
                        }
                    ],
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    }
                }
            ],
```

```
            "Groups": []
        },
        {
            "OwnerId": "123456789012",
            "ReservationId": "r-12345678910abcd",
            "Instances": [
                {
                    "ImageId": "ami-12345678",
                    "BlockDeviceMappings": [
                        {
                            "Ebs": {
                                "DeleteOnTermination": true,
                                "Status": "attached",
                                "VolumeId": "vol-111111111111"
                            },
                            "DeviceName": "/dev/xvda"
                        }
                    ],
                    "State": {
                        "Code": 80,
                        "Name": "stopped"
                    }
                }
            ],
            "Groups": []
        }
    ]
}
```

Exemplo 1 do JSONPath: obtenha uma string específica de uma resposta JSON

```
JSONPath:
$.Reservations[0].Instances[0].ImageId

Returns:
"ami-12345678"

Type: String
```

Exemplo 2 do JSONPath: obtenha um booliano específico de uma resposta JSON

```
JSONPath:
$.Reservations[0].Instances[0].BlockDeviceMappings[0].Ebs.DeleteOnTermination
```

```
Returns:
true

Type: Boolean
```

Exemplo 3 do JSONPath: obtenha um número inteiro específico de uma resposta JSON

```
JSONPath:
$.Reservations[0].Instances[0].State.Code

Returns:
16

Type: Integer
```

Exemplo 4 do JSONPath: faça uma verificação detalhada de uma resposta JSON e, em seguida, obtenha todos os valores para VolumeId como uma StringList

```
JSONPath:
$.Reservations..BlockDeviceMappings..VolumeId

Returns:
[
   "vol-000000000000",
   "vol-111111111111"
]

Type: StringList
```

Exemplo 5 do JSONPath: obter um objeto específico BlockDeviceMappings como um StringMap

```
JSONPath:
$.Reservations[0].Instances[0].BlockDeviceMappings[0]

Returns:
{
   "Ebs" : {
      "DeleteOnTermination" : true,
      "Status" : "attached",
      "VolumeId" : "vol-000000000000"
   },
```

```
   "DeviceName" : "/dev/xvda"
}

Type: StringMap
```

Exemplo 6 do JSONPath: faça uma verificação detalhada de uma resposta JSON e, em seguida, obtenha todos os objetos de estado como uma MapList

```
JSONPath:
$.Reservations..Instances..State

Returns:
[
   {
      "Code" : 16,
      "Name" : "running"
   },
   {
      "Code" : 80,
      "Name" : "stopped"
   }
]

Type: MapList
```

Exemplo 7 do JSONPath: filtro para instâncias no estado **`running`**

```
JSONPath:
$.Reservations..Instances[?(@.State.Name == 'running')]

Returns:
[
  {
    "ImageId": "ami-12345678",
    "BlockDeviceMappings": [
      {
        "Ebs": {
          "DeleteOnTermination": true,
          "Status": "attached",
          "VolumeId": "vol-000000000000"
        },
        "DeviceName": "/dev/xvda"
      }
```

```
    ],
    "State": {
      "Code": 16,
      "Name": "running"
    }
  }
]

Type: MapList
```

Exemplo 8 do JSONPath: retornar o **`ImageId`** de instâncias que não estão no estado **`running`**

```
JSONPath:
$.Reservations..Instances[?(@.State.Name != 'running')].ImageId

Returns:
[
  "ami-12345678"
]

Type: StringList | String
```

Criação de integrações de webhooks para o Automation

Para enviar mensagens usando webhooks durante uma automação, crie uma integração. As integrações podem ser invocadas durante uma automação usando a ação `aws:invokeWebhook` em seu runbook. Se você ainda não criou um webhook, consulte Criação de webhooks para integrações. Para saber mais a respeito da ação `aws:invokeWebhook`, consulte `aws:invokeWebhook` — Invoque uma integração de webhook do Automation.

Conforme mostrado nos procedimentos a seguir, você pode criar uma integração usando o console do Automation do Systems Manager ou sua ferramenta da linha de comando preferida.

Criação de integrações (console)

Para criar uma integração para o Automation (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação à esquerda, escolha Automation (Automação).
3. Escolha a guia Integrations (Integrações).

4. Selecione Add integration (Adicionar integração), e escolha Webhook.
5. Insira os valores necessários e todos os valores opcionais que você deseja incluir para a integração.
6. Selecione Add (Adicionar) para criar a integração.

Criação de integrações (linha de comando)

Para criar uma integração usando ferramentas da linha de comando, você deve criar o parâmetro `SecureString` necessário para uma integração. O Automation usa um namespace reservado no Parameter Store, uma ferramenta do Systems Manager, para armazenar informações sobre sua integração. Se você criar uma integração usando o AWS Management Console, o Automation lida com esse processo para você. Seguindo o namespace, você deve especificar o tipo de integração que deseja criar e, em seguida, o nome da sua integração. Atualmente, o Automation oferece suporte a Integrações do tipo `webhook`.

Os campos com suporte a integrações do tipo `webhook` são os seguintes:

- Descrição
- headers
- payload
- URL

Antes de começar

Caso ainda não tenha feito isso, instale e configure a AWS Command Line Interface (AWS CLI) ou o Ferramentas da AWS para PowerShell. Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI e Instalar o Ferramentas da AWS para PowerShell.

Para criar uma integração para o Automation (linha de comando)

- Execute os seguintes comandos para criar o parâmetro `SecureString` necessário para uma integração. Substitua cada *`espaço reservado para recurso de exemplo`* por suas próprias informações. O namespace `/d9d01087-4a3f-49e0-b0b4-d568d7826553/ssm/integrations/webhook/` é reservado no Parameter Store para integrações. O nome do parâmetro deve usar esse namespace seguido do nome da sua integração. Por exemplo, `/d9d01087-4a3f-49e0-b0b4-d568d7826553/ssm/integrations/webhook/`*`myWebhookIntegration`*.

Linux & macOS

```
aws ssm put-parameter \
    --name "/d9d01087-4a3f-49e0-b0b4-d568d7826553/ssm/integrations/
webhook/myWebhookIntegration" \
    --type "SecureString" \
    --data-type "aws:ssm:integration" \
    --value '{"description": "My first webhook integration for Automation.",
 "url": "myWebHookURL"}'
```

Windows

```
aws ssm put-parameter ^
    --name "/d9d01087-4a3f-49e0-b0b4-d568d7826553/ssm/integrations/
webhook/myWebhookIntegration" ^
    --type "SecureString" ^
    --data-type "aws:ssm:integration" ^
    --value  "{\"description\":\"My first webhook integration for Automation.\",
\"url\":\"myWebHookURL\"}"
```

PowerShell

```
Write-SSMParameter `
    -Name "/d9d01087-4a3f-49e0-b0b4-d568d7826553/ssm/integrations/
webhook/myWebhookIntegration" `
    -Type "SecureString"
    -DataType "aws:ssm:integration"
    -Value '{"description": "My first webhook integration for Automation.",
 "url": "myWebHookURL"}'
```

Criação de webhooks para integrações

Ao criar webhooks com seu provedor, observe o seguinte:

- O protocolo deve ser o HTTPS.
- Há suporte para os cabeçalhos de solicitação personalizados.
- Um corpo da solicitação padrão pode ser especificado.
- O corpo da solicitação padrão pode ser substituído quando uma integração for invocada usando a ação `aws:invokeWebhook`.

Gerenciar tempos limite em runbooks

A propriedade `timeoutSeconds` é compartilhada por todas as ações de automação. Você pode usar essa propriedade para especificar o valor do tempo limite de execução de uma ação. Além disso, você pode alterar como um tempo limite de ação afeta a automação e o status geral da execução. Você pode fazer isso definindo também as propriedades compartilhadas `onFailure` e `isCritical` para uma ação.

Por exemplo, dependendo do caso de uso, quando uma ação atinge o tempo limite, você poderá preferir que a automação continue com uma ação diferente e não afete o status geral da automação. Neste exemplo, você especifica o tempo de espera antes que o tempo limite da ação seja esgotado usando a propriedade `timeoutSeconds`. Especifique então a ação ou etapa que a automação deve executar no caso do tempo limite expirar. Especifique um valor usando o formato `step:`*`step name`* da propriedade `onFailure`, em vez do valor padrão de `Abort`. Por padrão, se o tempo limite de uma ação expirar, o status de execução da automação será `Timed Out`. Para evitar que um tempo limite afete o status da execução da automação, especifique `false` para a propriedade `isCritical`.

O exemplo a seguir mostra como definir as propriedades compartilhadas para uma ação descrita nesse cenário.

YAML

```
- name: verifyImageAvailability
  action: 'aws:waitForAwsResourceProperty'
  timeoutSeconds: 600
  isCritical: false
  onFailure: 'step:getCurrentImageState'
  inputs:
    Service: ec2
    Api: DescribeImages
    ImageIds:
      - '{{ createImage.newImageId }}'
    PropertySelector: '$.Images[0].State'
    DesiredValues:
      - available
  nextStep: copyImage
```

JSON

```
{
```

```
    "name": "verifyImageAvailability",
    "action": "aws:waitForAwsResourceProperty",
    "timeoutSeconds": 600,
    "isCritical": false,
    "onFailure": "step:getCurrentImageState",
    "inputs": {
        "Service": "ec2",
        "Api": "DescribeImages",
        "ImageIds": [
            "{{ createImage.newImageId }}"
        ],
        "PropertySelector": "$.Images[0].State",
        "DesiredValues": [
            "available"
        ]
    },
    "nextStep": "copyImage"
}
```

Para obter mais informações sobre propriedades compartilhadas por todas as ações de automação, consulte Propriedades compartilhadas por todas as ações.

Referência de runbooks do Systems Manager Automation

Para ajudar você a começar rapidamente, o AWS Systems Manager fornece runbooks predefinidos. Esses runbooks são mantidos pela Amazon Web Services, AWS Support e AWS Config. A referência de runbooks descreve cada um dos runbooks predefinidos fornecidos pelo Systems Manager, Suporte e AWS Config. Para obter mais informações, consulte Referência de runbooks do Systems Manager Automation.

Tutoriais

Os tutoriais a seguir ajudam você a usar o AWS Systems Manager Automation para lidar com casos de uso comuns. Esses tutoriais demonstram como usar seus próprios runbooks, runbooks predefinidos fornecidos pelo Automation e outras ferramentas do Systems Manager com outros Serviços da AWS.

Sumário

- Atualizar o AMIs
  - Atualizar uma AMI Linux

- Atualizar uma AMI (AWS CLI) Linux
- Atualizar uma AMIWindows Server
- Atualize uma AMI dourada usando Automation, AWS Lambda e Parameter Store
  - Tarefa 1: Criar um parâmetro no Systems Manager (Parameter Store)
  - Tarefa 2: Criar uma função do IAM para o AWS Lambda
  - Tarefa 3: Criar uma função do AWS Lambda
  - Tarefa 4: Criar um runbook e aplicar patches à AMI
- Atualizar AMIs usando o Automation e Jenkins
- Atualização de AMIs para grupos do Auto Scaling
  - Criar o runbook PatchAMIAndUpdateASG
- Uso de runbooks de autoatendimento do AWS Support
  - Executar a ferramenta EC2Rescue em instâncias inacessíveis
    - Como funciona
    - Antes de começar
      - Conceder permissões ao AWSSupport-EC2Rescue para realizar ações nas instâncias
        - Conceder permissões usando políticas do IAM
        - Conceder permissões usando um modelo do AWS CloudFormation
    - Executar a Automação
  - Redefinir senhas e chaves SSH em instâncias do EC2
    - Como funciona
    - Antes de começar
      - Conceder a AWSSupport-EC2Rescue permissões para realizar ações em suas instâncias
        - Conceder permissões usando políticas do IAM
        - Conceder permissões usando um modelo do AWS CloudFormation
    - Executar a Automação
- Transferência de dados para o Automation usando transformadores de entrada

Atualizar o AMIs

Os tutoriais a seguir explicam como atualizar as Amazon Machine Image (AMIs) para incluir os patches mais recentes.

Tópicos

- Atualizar uma AMI Linux
- Atualizar uma AMI (AWS CLI) Linux
- Atualizar uma AMIWindows Server
- Atualize uma AMI dourada usando Automation, AWS Lambda e Parameter Store
- Atualizar AMIs usando o Automation e Jenkins
- Atualização de AMIs para grupos do Auto Scaling

Atualizar uma AMI Linux

Esta demonstração do Systems Manager Automation mostra como usar o console ou a AWS CLI e o runbook `AWS-UpdateLinuxAmi` para atualizar uma AMI Linux com as versões mais atualizadas dos pacotes especificados. O Automation é uma ferramenta do AWS Systems Manager. O runbook `AWS-UpdateLinuxAmi` também automatiza a instalação de pacotes e configurações adicionais específicos de sites. Você pode atualizar várias distribuições do Linux usando esta demonstração, incluindo Ubuntu Server, CentOS, RHEL, SLES ou AMIs do Amazon Linux. Para obter uma lista completa das versões compatíveis do Linux, consulte Pré-requisitos da Patch Manager.

O runbook `AWS-UpdateLinuxAmi` permite automatizar tarefas de manutenção de imagens sem precisar criar o runbook em JSON ou YAML. Você pode usar o runbook `AWS-UpdateLinuxAmi` para realizar os seguintes tipos de tarefas:

- Atualize todos os pacotes de distribuição e software da Amazon em um Amazon Linux, Red Hat Enterprise Linux, Ubuntu Server, SUSE Linux Enterprise Server ou CentOS Amazon Machine Image (AMI). Esse é o comportamento padrão do runbook.
- Instale o AWS Systems Manager SSM Agent em uma imagem existente para habilitar ferramentas do Systems Manager, como a execução remota de comandos usando o AWS Systems Manager Run Command ou a coleta de inventário de software usando o Inventory.
- Instalar pacotes de software adicionais.

Antes de começar

Antes de começar a trabalhar com runbooks, configure funções e, opcionalmente, o EventBridge para o Automation. Para ter mais informações, consulte Configurar a automação. Esta demonstração também requer que você especifique o nome de um perfil de instância do AWS Identity and Access Management (IAM). Para obter mais informações sobre como criar um perfil de instância do IAM, consulte Configurar permissões de instância obrigatórias para o Systems Manager.

O runbook `AWS-UpdateLinuxAmi` aceita os seguintes parâmetros de entrada:

Parameter	Tipo	Descrição
SourceAmiId	String	(Obrigatório) O ID da AMI de origem.
IamInstanceProfileName	String	(Obrigatório) O nome do perfil para o perfil de instância do IAM criado em Configurar permissões de instância obrigatórias para o Systems Manager. A função de perfil de instância concede ao Automation permissão para realizar ações em suas instâncias, como executar comandos ou iniciar e interromper serviços. O runbook usa apenas o nome da função de perfil da instância. Se você especificar o nome do recurso da Amazon (ARN), a automação falhará.
AutomationAssumeRole	String	(Obrigatório) O nome da função de serviço do IAM que você criou em Configurar a automação. A função de serviço (também chamada de função assumida) concede ao Automation permissão

Parameter	Tipo	Descrição
		para assumir sua função do IAM e realizar ações em seu nome. Por exemplo, a função de serviço permite que o Automation crie uma nova AMI ao executar a ação `aws:createImage` em um runbook. Para esse parâmetro, o ARN completo deve ser especificado.
TargetAmiName	String	(Opcional) O nome da nova AMI após a sua criação. O nome padrão é uma string gerada pelo sistema que inclui o ID da AMI de origem, bem como a data e a hora de criação.
InstanceType	String	(Opcional) O tipo de instância a ser executada como o host do espaço de trabalho. Os tipos de instância variam de acordo com a região. O tipo padrão é t2.micro.
PreUpdateScript	String	(Opcional) URL de um script a ser executado antes de as atualizações serem aplicadas. O padrão (\"none\") é não executar um script.

Parameter	Tipo	Descrição
PostUpdateScript	String	(Opcional) URL de um script a ser executado depois de as atualizações de pacote serem aplicadas. O padrão (\"none\") é não executar um script.
IncludePackages	String	(Opcional) Somente atualiza esses pacotes nomeados. Por padrão (\"all\"), todas as atualizações disponíveis são aplicadas.
ExcludePackages	String	(Opcional) Nomes de pacotes para evitar atualizações, em todas as condições. Por padrão (\"none\"), nenhum pacote é excluído.

Etapas da Automação

O runbook `AWS-UpdateLinuxAmi` inclui as seguintes ações de automação, por padrão.

Etapa 1: launchInstance (ação **`aws:runInstances`**)

Esta etapa executa uma instância usando dados de usuário do Amazon Elastic Compute Cloud (Amazon EC2) e uma função de perfil de instância do IAM. Os dados de usuário instalam o SSM Agent apropriado, com base no sistema operacional. Instalar o SSM Agent permite que você utilize ferramentas do Systems Manager como Run Command, State Manager e Inventory.

Etapa 2: updateOSSoftware (ação **`aws:runCommand`**)

Essa etapa executa os seguintes comandos na instância executada:

- Baixa um script de atualização do Amazon S3.
- Executa um script de pré-atualização opcional.
- Atualiza pacotes de distribuição e softwares da Amazon.
- Executa um script de pós-atualização opcional.

O log de execução é armazenado na pasta /tmp para que o usuário possa visualizá-lo mais tarde.

Se quiser atualizar um conjunto específico de pacotes, forneça a lista usando o parâmetro `IncludePackages`. Quando essa lista é fornecida, o sistema tenta atualizar somente esses pacotes e suas dependências. Nenhuma outra atualização é realizada. Por padrão, quando nenhum pacote de inclusão é especificado, o programa atualiza todos os pacotes disponíveis.

Se quiser excluir a atualização de um conjunto específico de pacotes, forneça a lista ao parâmetro `ExcludePackages`. Se essa lista for fornecida, os pacotes permanecerão na sua versão atual, independentemente de qualquer outra opção especificada. Por padrão, quando nenhum pacote de exclusão é especificado, nenhum pacote é excluído.

Etapa 3: stopInstance (ação **`aws:changeInstanceState`**)

Essa etapa interrompe a instância atualizada.

Etapa 4: createImage (ação **`aws:createImage`**)

Essa etapa cria uma nova AMI com um nome descritivo que a vincula ao ID de origem e ao horário de criação. Por exemplo: “AMI Gerado pelo EC2 Automation em {{global:DATE_TIME}} do {{SourceAmiId}}” DATE_TIME e SourceID representam variáveis do Automation.

Etapa 5: terminateInstance (ação **`aws:changeInstanceState`**)

Essa etapa limpa a automação, encerrando a instância em execução.

Saída

A automação retorna o novo ID da AMI como resultado.

> **Note**
>
> Por padrão, quando o Automation executa o runbook `AWS-UpdateLinuxAmi`, o sistema cria uma instância temporária na VPC padrão (172.30.0.0/16). Se tiver excluído a VPC padrão, você receberá o seguinte erro:
> `VPC not defined 400`
> Para resolver esse problema, você deve fazer uma cópia do runbook `AWS-UpdateLinuxAmi` e especificar um ID de sub-rede. Para ter mais informações, consulte VPC não definida 400.

Para criar uma AMI com patch aplicado usando o Automation (AWS Systems Manager)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação à esquerda, escolha Automation (Automação).
3. Escolha Execute automation.
4. Na lista Automation document (Documento de automação), escolha **`AWS-UpdateLinuxAmi`**.
5. Na seção Detalhes do documento, verifique se a Versão do documento está definida como Versão padrão no runtime.
6. Escolha Próximo.
7. Na seção Execution mode (Modo de execução), escolha Simple Execution (Execução simples).
8. Na seção Input parameters (Parâmetros de entrada), insira as informações coletadas na seção Before you begin (Antes de começar).
9. Clique em Executar. O console exibe o status de execução da Automação.

Após a conclusão da automação, execute uma instância de teste da AMI atualizada para verificar as alterações.

> **Note**
>
> Se alguma etapa da automação falhar, as informações sobre a falha serão listadas na página Automation Executions (Execuções do Automation). A automação foi concebida para terminar a instância temporária após a conclusão bem-sucedida de todas as tarefas. Se uma etapa falhar, o sistema talvez não encerre a instância. Então, se uma etapa falhar, encerre manualmente a instância temporária.

Atualizar uma AMI (AWS CLI) Linux

Esta demonstração do AWS Systems Manager Automation mostra como usar a AWS Command Line Interface (AWS CLI) e o runbook `AWS-UpdateLinuxAmi` do Systems Manager para aplicar patches automaticamente a uma Amazon Machine Image (AMI) do Linux com as versões mais atualizadas dos pacotes especificados. O Automation é uma ferramenta do AWS Systems Manager. O runbook `AWS-UpdateLinuxAmi` também automatiza a instalação de pacotes e configurações adicionais específicos de sites. Você pode atualizar várias distribuições do Linux usando esta demonstração, incluindo Ubuntu Server, CentOS, RHEL, SLES ou AMIs do Amazon Linux. Para obter uma lista completa das versões compatíveis do Linux, consulte Pré-requisitos da Patch Manager.

O runbook `AWS-UpdateLinuxAmi` permite automatizar tarefas de manutenção de imagens sem precisar criar o runbook em JSON ou YAML. Você pode usar o runbook `AWS-UpdateLinuxAmi` para realizar os seguintes tipos de tarefas:

- Atualize todos os pacotes de distribuição e o software da Amazon em um Amazon Linux, Red Hat Enterprise Linux, Ubuntu Server, SLES ou CentOS Amazon Machine Image (AMI). Esse é o comportamento padrão do runbook.
- Instale o AWS Systems Manager SSM Agent em uma imagem existente para habilitar os recursos do Systems Manager, como a execução remota de comandos usando o AWS Systems Manager Run Command ou a coleta de inventário de software usando o Inventory.
- Instalar pacotes de software adicionais.

Antes de começar

Antes de começar a trabalhar com runbooks, configure funções e, opcionalmente, o EventBridge para o Automation. Para obter mais informações, consulte Configurar a automação. Esta demonstração também requer que você especifique o nome de um perfil de instância do AWS Identity and Access Management (IAM). Para obter mais informações sobre como criar um perfil de instância do IAM, consulte Configurar permissões de instância obrigatórias para o Systems Manager.

O runbook `AWS-UpdateLinuxAmi` aceita os seguintes parâmetros de entrada:

Parameter	Tipo	Descrição
SourceAmiId	String	(Obrigatório) O ID da AMI de origem. Você pode fazer referência automaticamente ao ID mais recente de uma AMI do Amazon EC2 para Linux usando um parâmetro público do AWS Systems Manager Parameter Store . Para obter mais informaçõ es, consulte Consultar os IDs de AMI do Amazon Linux mais recentes usando o AWS

Parameter	Tipo	Descrição
		Systems Manager Parameter Store.
IamInstanceProfileName	String	(Obrigatório) O nome do perfil para o perfil de instância do IAM criado em Configurar permissões de instância obrigatórias para o Systems Manager. A função de perfil de instância concede ao Automation permissão para realizar ações em suas instâncias, como executar comandos ou iniciar e interromper serviços. O runbook usa apenas o nome da função de perfil da instância.

Parameter	Tipo	Descrição
AutomationAssumeRole	String	(Obrigatório) O nome da função de serviço do IAM que você criou em Configurar a automação. A função de serviço (também chamada de função assumida) concede ao Automation permissão para assumir sua função do IAM e realizar ações em seu nome. Por exemplo, a função de serviço permite que o Automation crie uma nova AMI ao executar a ação `aws:createImage` em um runbook. Para esse parâmetro, o ARN completo deve ser especificado.
TargetAmiName	String	(Opcional) O nome da nova AMI após a sua criação. O nome padrão é uma string gerada pelo sistema que inclui o ID da AMI de origem, bem como a data e a hora de criação.
InstanceType	String	(Opcional) O tipo de instância a ser executada como o host do espaço de trabalho. Os tipos de instância variam de acordo com a região. O tipo padrão é t2.micro.

Parameter	Tipo	Descrição
PreUpdateScript	String	(Opcional) URL de um script a ser executado antes de as atualizações serem aplicadas . O padrão (\"none\") é não executar um script.
PostUpdateScript	String	(Opcional) URL de um script a ser executado depois de as atualizações de pacote serem aplicadas. O padrão (\"none\") é não executar um script.
IncludePackages	String	(Opcional) Somente atualiza esses pacotes nomeados. Por padrão (\"all\"), todas as atualizações disponíveis são aplicadas.
ExcludePackages	String	(Opcional) Nomes de pacotes para evitar atualizações, em todas as condições. Por padrão (\"none\"), nenhum pacote é excluído.

Etapas da Automação

O runbook `AWS-UpdateLinuxAmi` inclui as seguintes etapas, por padrão.

Etapa 1: launchInstance (ação **`aws:runInstances`**)

Esta etapa executa uma instância usando dados de usuário do Amazon Elastic Compute Cloud (Amazon EC2) e uma função de perfil de instância do IAM. Os dados de usuário instalam o SSM Agent apropriado, com base no sistema operacional. Instalar o SSM Agent permite que você utilize ferramentas do Systems Manager como Run Command, State Manager e Inventory.

Etapa 2: updateOSSoftware (ação **`aws:runCommand`**)

Essa etapa executa os seguintes comandos na instância executada:

- Baixe um script de atualização do Amazon Simple Storage Service (Amazon S3).
- Executa um script de pré-atualização opcional.
- Atualiza pacotes de distribuição e softwares da Amazon.
- Executa um script de pós-atualização opcional.

O log de execução é armazenado na pasta /tmp para que o usuário possa visualizá-lo mais tarde.

Se quiser atualizar um conjunto específico de pacotes, forneça a lista usando o parâmetro `IncludePackages`. Quando essa lista é fornecida, o sistema tenta atualizar somente esses pacotes e suas dependências. Nenhuma outra atualização é realizada. Por padrão, quando nenhum pacote de inclusão é especificado, o programa atualiza todos os pacotes disponíveis.

Se quiser excluir a atualização de um conjunto específico de pacotes, forneça a lista ao parâmetro `ExcludePackages`. Se essa lista for fornecida, os pacotes permanecerão na sua versão atual, independentemente de qualquer outra opção especificada. Por padrão, quando nenhum pacote de exclusão é especificado, nenhum pacote é excluído.

Etapa 3: stopInstance (ação **`aws:changeInstanceState`**)

Essa etapa interrompe a instância atualizada.

Etapa 4: createImage (ação **`aws:createImage`**)

Essa etapa cria uma nova AMI com um nome descritivo que a vincula ao ID de origem e ao horário de criação. Por exemplo: “AMI Generated by EC2 Automation on {{global:DATE_TIME}} from {{SourceAmiId}}” em que DATE_TIME e SourceID representam variáveis de Automação.

Etapa 5: terminateInstance (ação **`aws:changeInstanceState`**)

Essa etapa limpa a automação, encerrando a instância em execução.

Saída

A automação retorna o novo ID da AMI como resultado.

> **Note**
>
> Por padrão, quando o Automation executa o runbook `AWS-UpdateLinuxAmi`, o sistema cria uma instância temporária na VPC padrão (172.30.0.0/16). Se tiver excluído a VPC padrão, você receberá o seguinte erro:
> `VPC not defined 400`
> Para resolver esse problema, você deve fazer uma cópia do runbook `AWS-UpdateLinuxAmi` e especificar um ID de sub-rede. Para obter mais informações, consulte VPC não definida 400.

Para criar uma AMI com patch aplicado usando o Automation

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Execute o comando a seguir para executar o runbook `AWS-UpdateLinuxAmi`. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   ```
   aws ssm start-automation-execution \
       --document-name "AWS-UpdateLinuxAmi" \
       --parameters \
       SourceAmiId=AMI ID, \
       IamInstanceProfileName=IAM instance profile, \
       AutomationAssumeRole='arn:aws:iam::
   {{global:ACCOUNT_ID}}:role/AutomationServiceRole'
   ```

   O comando retorna um ID de execução. Copie esse ID para a área de transferência. Você usará esse ID para visualizar o status da automação.

   ```
   {
       "AutomationExecutionId": "automation execution ID"
   }
   ```

3. Para visualizar a automação usando a AWS CLI, execute o seguinte comando:

   ```
   aws ssm describe-automation-executions
   ```

4. Para visualizar detalhes sobre o andamento da automação, execute o comando a seguir. Substitua *automation execution ID* (ID de execução da automação) por suas próprias informações.

```
aws ssm get-automation-execution --automation-execution-id automation execution ID
```

O processo de atualização pode demorar 30 minutos ou mais para ser concluído.

> **Note**
>
> Você pode também monitorar o status da automação no console. Na lista, escolha a automação que você acabou de processar e depois escolha a guia Steps (Etapas). Esta guia mostra o status das ações de automação.

Após a conclusão da automação, execute uma instância de teste da AMI atualizada para verificar as alterações.

> **Note**
>
> Se alguma etapa da automação falhar, as informações sobre a falha serão listadas na página Automation Executions (Execuções do Automation). A automação foi concebida para terminar a instância temporária após a conclusão bem-sucedida de todas as tarefas. Se uma etapa falhar, o sistema talvez não encerre a instância. Então, se uma etapa falhar, encerre manualmente a instância temporária.

Atualizar uma AMIWindows Server

O runbook `AWS-UpdateWindowsAmi` permite automatizar tarefas de manutenção de imagens em Amazon Machine Image (AMI) do Amazon Windows, sem precisar criar o runbook em JSON ou YAML. Este runbook tem suporte para o Windows Server 2008 R2 ou posterior. Você pode usar o runbook `AWS-UpdateWindowsAmi` para realizar os seguintes tipos de tarefas:

- Instalar todas as atualizações do Windows e atualizar softwares da Amazon (comportamento padrão).
- Instalar atualizações específicas do Windows e atualizar softwares da Amazon.
- Personalize uma AMI usando seus scripts.

Antes de começar

Antes de começar a trabalhar com runbooks, configure funções para o Automation a fim de adicionar uma política `iam:PassRole` que referencia o ARN do perfil da instância ao qual você deseja conceder acesso. Opcionalmente, configure o Amazon EventBridge para o Automation, uma ferramenta do AWS Systems Manager. Para obter mais informações, consulte Configurar a automação. Esta demonstração também requer que você especifique o nome de um perfil de instância do AWS Identity and Access Management (IAM). Para obter mais informações sobre como criar um perfil de instância do IAM, consulte Configurar permissões de instância obrigatórias para o Systems Manager.

> **Note**
>
> As atualizações do AWS Systems Manager SSM Agent são normalmente implementadas em regiões diferentes e em momentos distintos. Quando você personalizar ou atualizar uma AMI, use apenas AMIs de origem publicadas para a região na qual você está trabalhando. Isso garante que você trabalhe com o SSM Agent mais recente lançado nessa região e evite problemas de compatibilidade.

O runbook `AWS-UpdateWindowsAmi` aceita os seguintes parâmetros de entrada:

Parameter	Tipo	Descrição
SourceAmiId	String	(Obrigatório) O ID da AMI de origem. Você pode fazer referência automatic amente ao ID de uma AMI do Windows Server mais recente, ID usando um parâmetro público Parameter Store do Systems Manager. Para obter mais informações, consulte Consultar os mais recentes IDs da AMI do Windows usando o Parameter Store do AWS Systems Manager.

Parameter	Tipo	Descrição
SubnetId	String	(Opcional) A sub-rede na qual você deseja iniciar a instância temporária. Você deve especificar um valor para esse parâmetro se tiver excluído a VPC padrão.
IamInstanceProfileName	String	(Obrigatório) O nome do perfil para o perfil de instância do IAM criado em Configurar permissões de instância obrigatórias para o Systems Manager. A função de perfil de instância concede ao Automation permissão para realizar ações em suas instâncias, como executar comandos ou iniciar e interromper serviços. O runbook usa apenas o nome da função de perfil da instância.

Parameter	Tipo	Descrição
AutomationAssumeRole	String	(Obrigatório) O nome da função de serviço do IAM que você criou em Configurar a automação. A função de serviço (também chamada de função assumida) concede ao Automation permissão para assumir sua função do IAM e realizar ações em seu nome. Por exemplo, a função de serviço permite que o Automation crie uma nova AMI ao executar a ação `aws:createImage` em um runbook. Para esse parâmetro, o ARN completo deve ser especificado.
TargetAmiName	String	(Opcional) O nome da nova AMI após a sua criação. O nome padrão é uma string gerada pelo sistema que inclui o ID da AMI de origem, bem como a data e a hora de criação.
InstanceType	String	(Opcional) O tipo de instância a ser executada como o host do espaço de trabalho. Os tipos de instância variam de acordo com a região. O tipo padrão é t2.medium.

Parameter	Tipo	Descrição
PreUpdateScript	String	(Opcional) Um script a ser executado antes de atualizar a AMI. Insira um script no runbook ou no runtime como um parâmetro.
PostUpdateScript	String	(Opcional) Um script a ser executado depois de atualizar a AMI. Insira um script no runbook ou no runtime como um parâmetro.
IncludeKbs	String	(Opcional) Especifique um ou mais IDs de artigo da Base de Dados de Conhecimento Microsoft (KB) para incluir. Você pode instalar vários IDs usando valores separados por vírgulas. Formatos válidos: KB9876543 ou 9876543.
ExcludeKbs	String	(Opcional) Especifique um ou mais IDs de artigo da Base de Dados de Conhecimento Microsoft (KB) para excluir. Você pode excluir vários IDs usando valores separados por vírgulas. Formatos válidos: KB9876543 ou 9876543.

Parameter	Tipo	Descrição
Categorias	String	(Opcional) Especifique uma ou mais categorias de atualização. Você pode filtrar categorias usando valores separados por vírgulas. Opções: Atualizaç ão crítica, Atualização de segurança, Atualização de definição, Pacote de atualizaç ões, Pacote de serviços, Ferramenta, Atualização ou Driver. Os formatos válidos incluem uma única entrada, por exemplo: Atualização crítica. Como alternativa, você pode especificar uma lista separada por vírgulas: Atualização crítica, Atualizaç ão de segurança, Atualização de definição.

Parameter	Tipo	Descrição
SeverityLevels	String	(Opcional) Especifique um ou mais níveis de gravidade MSRC associados a uma atualização. Você pode filtrar os níveis de gravidade usando valores separados por vírgulas. Opções: Crítica, Importante, Baixa, Moderada ou Não especificada. Os formatos válidos incluem uma única entrada, por exemplo: Crítica. Como alternativa, você pode especificar uma lista separada por vírgulas: Crítica, Importante, Baixa.

Etapas da Automação

O runbook `AWS-UpdateWindowsAmi` inclui as seguintes etapas, por padrão.

Etapa 1: launchInstance (ação **`aws:runInstances`**)

Esta etapa executa uma instância com uma função de perfil da instância do IAM no `SourceAmiID` especificado.

Etapa 2: runPreUpdateScript (ação **`aws:runCommand`**)

Essa etapa permite especificar um script como uma string que é executada antes que as atualizações sejam instaladas.

Etapa 3: updateEC2Config (ação **`aws:runCommand`**)

Esta etapa usa o runbook `AWS-InstallPowerShellModule` para baixar um módulo PowerShell público da AWS. O Systems Manager verifica a integridade do módulo usando um hash SHA-256. Em seguida, o Systems Manager verifica o sistema operacional para determinar se deve atualizar EC2Config ou EC2Launch. EC2Config é executado no Windows Server 2008 R2 até o Windows Server 2012 R2. EC2Launch é executado no Windows Server 2016.

Etapa 4: updateSSMAgent (ação **aws:runCommand**)

Esta etapa atualiza o SSM Agent usando o runbook `AWS-UpdateSSMAgent`.

Etapa 5: updateAWSPVDriver (ação **aws:runCommand**)

Esta etapa atualiza os drivers de rede do AWS PV usando o runbook `AWS-ConfigureAWSPackage`.

Etapa 6: updateAwsEnaNetworkDriver (ação **aws:runCommand**)

Esta etapa atualiza os drivers de rede do AWS ENA usando o runbook `AWS-ConfigureAWSPackage`.

Etapa 7: installWindowsUpdates (ação **aws:runCommand**)

Essa etapa instala atualizações do Windows usando o runbook `AWS-InstallWindowsUpdates`. Por padrão, o Systems Manager procura e instala todas as atualizações ausentes. Você pode alterar o comportamento padrão especificando um dos seguintes parâmetros: `IncludeKbs`, `ExcludeKbs`, `Categories` ou `SeverityLevels`.

Etapa 8: runPostUpdateScript (ação **aws:runCommand**)

Essa etapa permite especificar um script como uma string que é executada após a instalação das atualizações.

Etapa 9: runSysprepGeneralize (ação **aws:runCommand**)

Esta etapa usa o runbook `AWS-InstallPowerShellModule` para baixar um módulo PowerShell público da AWS. O Systems Manager verifica a integridade do módulo usando um hash SHA-256. Em seguida, o Systems Manager executa o sysprep usando métodos com suporte da AWS para EC2Launch (Windows Server 2016) ou EC2Config (Windows Server 2008 R2 a 2012 R2).

Etapa 10: stopInstance (ação **aws:changeInstanceState**)

Essa etapa interrompe a instância atualizada.

Etapa 11: createImage (ação **aws:createImage**)

Essa etapa cria uma nova AMI com um nome descritivo que a vincula ao ID de origem e ao horário de criação. Por exemplo: “AMI Generated by EC2 Automation on {{global:DATE_TIME}} from {{SourceAmiId}}” em que DATE_TIME e SourceID representam variáveis de Automação.

Etapa 12: TerminateInstance (ação **aws:changeInstanceState**)

Essa etapa limpa a automação, encerrando a instância em execução.

Saída

Essa seção permite designar as saídas de várias etapas ou valores de qualquer parâmetro como saída da Automação. Por padrão, o resultado é o ID da AMI do Windows atualizada, criada pela automação.

> **Note**
>
> Por padrão, quando o Automation executa o runbook `AWS-UpdateWindowsAmi` e cria uma instância temporária, o sistema usa a VPC padrão (172.30.0.0/16). Se tiver excluído a VPC padrão, você receberá o seguinte erro:
> VPC não definida 400
> Para resolver esse problema, você deve fazer uma cópia do runbook `AWS-UpdateWindowsAmi` e especificar um ID de sub-rede. Para obter mais informações, consulte VPC não definida 400.

Para criar uma AMI Windows com patch aplicado usando o Automation

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Execute o comando a seguir para executar o runbook `AWS-UpdateWindowsAmi`. Substitua cada *`espaço reservado para recurso de exemplo`* por suas próprias informações. O exemplo de comando abaixo usa uma AMI recente do Amazon EC2 para minimizar o número de patches que precisam ser aplicados. Se você executar esse comando mais de uma vez, deverá especificar um valor exclusivo para `targetAMIname`. Os nomes da AMI deverão ser exclusivos.

   ```
   aws ssm start-automation-execution \
       --document-name="AWS-UpdateWindowsAmi" \
       --parameters SourceAmiId='AMI ID',IamInstanceProfileName='IAM
    instance profile',AutomationAssumeRole='arn:aws:iam::
   {{global:ACCOUNT_ID}}:role/AutomationServiceRole'
   ```

   O comando retorna um ID de execução. Copie esse ID para a área de transferência. Você usará esse ID para visualizar o status da automação.

   ```
   {
   ```

```
    "AutomationExecutionId": "automation execution ID"
}
```

3. Para visualizar a automação usando a AWS CLI, execute o seguinte comando:

```
aws ssm describe-automation-executions
```

4. Para visualizar detalhes sobre o andamento da automação, execute o comando a seguir.

```
aws ssm get-automation-execution
    --automation-execution-id automation execution ID
```

> **Note**
>
> Dependendo do número de patches aplicados, o processo de aplicação de patches do Windows, executado nessa automação de amostra, poderá demorar 30 minutos ou mais para ser concluído.

Atualize uma AMI dourada usando Automation, AWS Lambda e Parameter Store

O exemplo a seguir usa o modelo onde uma organização mantém suas próprias AMIs e aplica patches a elas periodicamente, em vez de se basear em AMIs do Amazon Elastic Compute Cloud (Amazon EC2).

O procedimento a seguir mostra como aplicar patches de sistema operacional (SO) automaticamente a uma AMI que já é considerada a AMI mais atualizada ou mais recente. No exemplo, o valor padrão do parâmetro `SourceAmiId` é definido por um parâmetro do AWS Systems Manager Parameter Store chamado `latestAmi`. O valor de `latestAmi` é atualizado por uma função AWS Lambda invocada no final da automação. Como resultado desse processo de automação, o tempo e o esforço gastos na aplicação de patches das AMIs são minimizados, pois o patch é sempre aplicado à AMI mais atualizada. O Parameter Store e o Automation são ferramentas do AWS Systems Manager.

Antes de começar

Configure as funções do Automation e, opcionalmente, o Amazon EventBridge para Automation. Para obter mais informações, consulte Configurar a automação.

Conteúdo

- Tarefa 1: Criar um parâmetro no Systems Manager (Parameter Store)
- Tarefa 2: Criar uma função do IAM para o AWS Lambda
- Tarefa 3: Criar uma função do AWS Lambda
- Tarefa 4: Criar um runbook e aplicar patches à AMI

Tarefa 1: Criar um parâmetro no Systems Manager (Parameter Store)

Crie um parâmetro de string no Parameter Store que use as seguintes informações:

- Name (Nome): `latestAmi`.
- Valor: um ID de AMI. Por exemplo: `ami-188d6e0e`.

Para obter mais informações sobre como criar um parâmetro String usando o Parameter Store, consulte Criar parâmetros do Parameter Store no Systems Manager.

Tarefa 2: Criar uma função do IAM para o AWS Lambda

Use o procedimento a seguir para criar uma função de serviço do IAM para o AWS Lambda. Essas políticas dão ao Lambda a permissão necessária para atualizar o valor do parâmetro `latestAmi` usando uma função do Lambda e do Systems Manager.

Para criar uma função de serviço do IAM para o Lambda

1. Faça login no AWS Management Console e abra o console do IAM, em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Políticas e, em seguida, Criar política.
3. Selecione a guia JSON.
4. Substitua os conteúdos padrão pela política a seguir. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   JSON

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

```
            "Effect": "Allow",
            "Action": "logs:CreateLogGroup",
            "Resource": "arn:aws:logs:us-east-1:111122223333:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:111122223333:log-group:/aws/
lambda/function name:*"
            ]
        }
    ]
}
```

5. Escolha Próximo: tags.
6. (Opcional) Adicione um ou mais pares de chave-valor de etiqueta para organizar, monitorar ou controlar acesso para essa política.
7. Escolha Próximo: revisar.
8. Na página Revisar política, em Nome, digite um nome para a política em linha, como **amiLambda**.
9. Escolha Criar política.
10. Repita as etapas 2 e 3.
11. Cole a política a seguir. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

    JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ssm:PutParameter",
            "Resource": "arn:aws:ssm:us-
east-1:111122223333:parameter/latestAmi"
```

```
        },
        {
            "Effect": "Allow",
            "Action": "ssm:DescribeParameters",
            "Resource": "*"
        }
    ]
}
```

12. Escolha Próximo: tags.
13. (Opcional) Adicione um ou mais pares de chave-valor de etiqueta para organizar, monitorar ou controlar acesso para essa política.
14. Escolha Próximo: revisar.
15. Na página Revisar política, em Nome, digite um nome para a política em linha, como **`amiParameter`**.
16. Escolha Criar política.
17. No painel de navegação, escolha Perfis e Criar perfil.
18. Em seguida, em Caso de uso, escolha Lambda e escolha Próximo.
19. Na página Adicionar permissões, use o campo Pesquisar para localizar as duas políticas criadas anteriormente.
20. Marque a caixa de seleção ao lado das políticas e, em seguida, escolha Próximo.
21. Em Role name (Nome da função), insira um nome para a nova função, como **`lambda-ssm-role`** ou outro nome que você preferir.

    > **Note**
    >
    > Como várias entidades podem fazer referência à função, não é possível alterar o nome da função depois que ela é criada.

22. (Opcional) Adicione um ou mais pares chave-valor de etiquetas para organizar, acompanhar ou controlar o acesso a esse perfil e, em seguida, escolha Criar perfil.

Tarefa 3: Criar uma função do AWS Lambda

Use o seguinte procedimento para criar uma função do Lambda que atualize automaticamente o valor do parâmetro `latestAmi`.

Criar uma função do Lambda

1. Faça login no AWS Management Console e abra o console do AWS Lambda em https://console.aws.amazon.com/lambda/.
2. Escolha a opção Criar função.
3. Na página Create function, selecione Author from scratch.
4. Em Function name (Nome da função), insira **`Automation-UpdateSsmParam`**.
5. Em Runtime, escolha Python 3.8.
6. Em Arquitetura, selecione o tipo de processador de computador que o Lambda usará para executar a função, x86_64 ou arm64.
7. Na seção Permissões, expanda Alterar perfil de execução padrão.
8. Selecione Use an existing role (Usar uma função existente) e escolha a função de serviço do Lambda criada na Tarefa 2.
9. Escolha a opção Criar função.
10. Na área Origem do código, na guia lambda_function, exclua o código pré-preenchido no campo e cole a amostra de código a seguir.

```
from __future__ import print_function

import json
import boto3

print('Loading function')


#Updates an SSM parameter
#Expects parameterName, parameterValue
def lambda_handler(event, context):
    print("Received event: " + json.dumps(event, indent=2))

    # get SSM client
    client = boto3.client('ssm')

    #confirm  parameter exists before updating it
    response = client.describe_parameters(
       Filters=[
          {
           'Key': 'Name',
           'Values': [ event['parameterName'] ]
```

```
          },
        ]
    )

    if not response['Parameters']:
        print('No such parameter')
        return 'SSM parameter not found.'

    #if parameter has a Description field, update it PLUS the Value
    if 'Description' in response['Parameters'][0]:
        description = response['Parameters'][0]['Description']

        response = client.put_parameter(
          Name=event['parameterName'],
          Value=event['parameterValue'],
          Description=description,
          Type='String',
          Overwrite=True
        )

    #otherwise just update Value
    else:
        response = client.put_parameter(
          Name=event['parameterName'],
          Value=event['parameterValue'],
          Type='String',
          Overwrite=True
        )

    responseString = 'Updated parameter %s with value %s.' %
(event['parameterName'], event['parameterValue'])

    return responseString
```

11. Escolha Arquivo, Salvar.
12. Para testar a função do Lambda, no menu Teste, escolha Configurar evento de teste.
13. Em Event name (Nome do evento), insira um nome para o evento de teste, como **`MyTestEvent`**.
14. Substitua o texto existente pelo seguinte JSON. Substitua *`AMI ID`* (ID da AMI) por suas próprias informações para definir o valor do parâmetro `latestAmi`.

```
{
```

```
   "parameterName":"latestAmi",
   "parameterValue":"AMI ID"
}
```

15. Escolha Salvar.
16. Selecione Test (Testar) para testar a função. Na guia Resultado da execução, o status deve ser informado como Com êxito, junto com outros detalhes sobre a atualização.

Tarefa 4: Criar um runbook e aplicar patches à AMI

Use o procedimento a seguir para criar e executar um runbook que aplica patches à AMI especificada para o parâmetro latestAmi. Depois que a automação for concluída, o valor de latestAmi será atualizado com o ID da AMI que acabou de receber patch. As automações subsequentes usarão a AMI criada pela execução anterior.

Para criar e executar o runbook

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Documents.
3. Em Criar documento, escolha Automation.
4. Em Nome, digite **`UpdateMyLatestWindowsAmi`**.
5. Escolha a guia Editor e depois escolha Edit (Editar).
6. Escolha OK quando solicitado.
7. No campo Editor de documentos, substitua o conteúdo padrão pelo conteúdo do runbook de amostra YAML apresentado a seguir.

```
---
description: Systems Manager Automation Demo - Patch AMI and Update ASG
schemaVersion: '0.3'
assumeRole: '{{ AutomationAssumeRole }}'
parameters:
  AutomationAssumeRole:
    type: String
    description: '(Required) The ARN of the role that allows Automation to perform
 the actions on your behalf. If no role is specified, Systems Manager Automation
 uses your IAM permissions to execute this document.'
    default: ''
  SourceAMI:
    type: String
```

```
    description: The ID of the AMI you want to patch.
    default: '{{ ssm:latestAmi }}'
  SubnetId:
    type: String
    description: The ID of the subnet where the instance from the SourceAMI
 parameter is launched.
  SecurityGroupIds:
    type: StringList
    description: The IDs of the security groups to associate with the instance
 that's launched from the SourceAMI parameter.
  NewAMI:
    type: String
    description: The name of of newly patched AMI.
    default: 'patchedAMI-{{global:DATE_TIME}}'
  InstanceProfile:
    type: String
    description: The name of the IAM instance profile you want the source instance
 to use.
  SnapshotId:
    type: String
    description: (Optional) The snapshot ID to use to retrieve a patch baseline
 snapshot.
    default: ''
  RebootOption:
    type: String
    description: '(Optional) Reboot behavior after a patch Install operation. If
 you choose NoReboot and patches are installed, the instance is marked as non-
compliant until a subsequent reboot and scan.'
    allowedValues:
      - NoReboot
      - RebootIfNeeded
    default: RebootIfNeeded
  Operation:
    type: String
    description: (Optional) The update or configuration to perform on the instance.
 The system checks if patches specified in the patch baseline are installed on the
 instance. The install operation installs patches missing from the baseline.
    allowedValues:
      - Install
      - Scan
    default: Install
mainSteps:
  - name: startInstances
    action: 'aws:runInstances'
```

```
    timeoutSeconds: 1200
    maxAttempts: 1
    onFailure: Abort
    inputs:
      ImageId: '{{ SourceAMI }}'
      InstanceType: m5.large
      MinInstanceCount: 1
      MaxInstanceCount: 1
      IamInstanceProfileName: '{{ InstanceProfile }}'
      SubnetId: '{{ SubnetId }}'
      SecurityGroupIds: '{{ SecurityGroupIds }}'
  - name: verifyInstanceManaged
    action: 'aws:waitForAwsResourceProperty'
    timeoutSeconds: 600
    inputs:
      Service: ssm
      Api: DescribeInstanceInformation
      InstanceInformationFilterList:
        - key: InstanceIds
          valueSet:
            - '{{ startInstances.InstanceIds }}'
      PropertySelector: '$.InstanceInformationList[0].PingStatus'
      DesiredValues:
        - Online
    onFailure: 'step:terminateInstance'
  - name: installPatches
    action: 'aws:runCommand'
    timeoutSeconds: 7200
    onFailure: Abort
    inputs:
      DocumentName: AWS-RunPatchBaseline
      Parameters:
        SnapshotId: '{{SnapshotId}}'
        RebootOption: '{{RebootOption}}'
        Operation: '{{Operation}}'
      InstanceIds:
        - '{{ startInstances.InstanceIds }}'
  - name: stopInstance
    action: 'aws:changeInstanceState'
    maxAttempts: 1
    onFailure: Continue
    inputs:
      InstanceIds:
        - '{{ startInstances.InstanceIds }}'
```

```
      DesiredState: stopped
  - name: createImage
    action: 'aws:createImage'
    maxAttempts: 1
    onFailure: Continue
    inputs:
      InstanceId: '{{ startInstances.InstanceIds }}'
      ImageName: '{{ NewAMI }}'
      NoReboot: false
      ImageDescription: Patched AMI created by Automation
  - name: terminateInstance
    action: 'aws:changeInstanceState'
    maxAttempts: 1
    onFailure: Continue
    inputs:
      InstanceIds:
        - '{{ startInstances.InstanceIds }}'
      DesiredState: terminated
  - name: updateSsmParam
    action: aws:invokeLambdaFunction
    timeoutSeconds: 1200
    maxAttempts: 1
    onFailure: Abort
    inputs:
        FunctionName: Automation-UpdateSsmParam
        Payload: '{"parameterName":"latestAmi",
 "parameterValue":"{{createImage.ImageId}}"}'
outputs:
- createImage.ImageId
```

8. Escolha Criar automação.
9. No painel de navegação, selecione Automation e Execute automation (Executar automação).
10. Na página Choose document (Escolher documento), escolha a guia Owned by me (Pertencem a mim).
11. Procure o runbook UpdateMyLatestWindowsAmi e selecione o botão no cartão UpdateMyLatestWindowsAmi.
12. Escolha Próximo.
13. Escolha Simple execution (Execução simples).
14. Especificar valores para os parâmetro de entrada.
15. Clique em Executar.

16. Após a conclusão da automação, escolha Parameter Store no painel de navegação e confirme se o novo valor para `latestAmi` corresponde ao valor retornado pela automação. Você também pode verificar se o novo ID da AMI corresponde à saída do Automation na seção AMIs do console do Amazon EC2.

Atualizar AMIs usando o Automation e Jenkins

Se a sua organização usar o software Jenkins em um pipeline de CI/CD, você poderá adicionar o Automation como uma etapa de pós-compilação para pré-instalar as versões da aplicação na Amazon Machine Images (AMIs). O Automation é uma ferramenta do AWS Systems Manager. Você também pode usar o recurso de agendamento do Jenkins para chamar o Automation e criar sua própria cadência de aplicação de patches de sistema operacional (SO).

O exemplo abaixo mostra como invocar o Automation em um servidor Jenkins em execução on-premises ou no Amazon Elastic Compute Cloud (Amazon EC2). Para realizar a autenticação, o servidor Jenkins usa credenciais da AWS com base em uma política do IAM criada no exemplo e anexada ao seu perfil de instância.

> **Note**
>
> Certifique-se de seguir as práticas recomendadas de segurança do Jenkins ao configurar sua instância.

Antes de começar

Conclua as seguintes tarefas antes de configurar o Automation com o Jenkins:

- Conclua o exemplo Atualize uma AMI dourada usando Automation, AWS Lambda e Parameter Store. O exemplo a seguir usa o runbook UpdateMyLatestWindowsAmi criado nesse exemplo.
- Configure as funções do IAM para o Automation. O Systems Manager requer uma função de perfil da instância e um ARN da função de serviço para processar automações. Para obter mais informações, consulte Configurar a automação.

Como criar uma política do IAM para o servidor Jenkins

1. Faça login no AWS Management Console e abra o console do IAM, em https://console.aws.amazon.com/iam/.

2. No painel de navegação, selecione Políticas e, em seguida, Criar política.
3. Selecione a guia JSON.
4. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ssm:StartAutomationExecution",
            "Resource": [
                "arn:aws:ssm:us-
east-1:111122223333:document/UpdateMyLatestWindowsAmi",
                "arn:aws:ssm:us-east-1:111122223333:automation-
definition/UpdateMyLatestWindowsAmi:$DEFAULT"
            ]
        }
    ]
}
```

5. Selecione Revisar política.
6. Na página Revisar política, em Nome, digite um nome para a política em linha, como **`JenkinsPolicy`**.
7. Escolha Criar política.
8. No painel de navegação, selecione Perfis.
9. Escolha o perfil de instância que está anexado ao seu servidor Jenkins.
10. Na guia Permissões, selecione Adicionar permissões e escolha Anexar políticas.
11. Na seção Outras políticas de permissões, insira o nome da política criada nas etapas anteriores. Por exemplo, JenkinsPolicy.
12. Marque a caixa de seleção ao lado da sua política e escolha Anexar políticas.

Use o seguinte procedimento para configurar a AWS CLI no seu servidor Jenkins.

Para configurar o servidor Jenkins para Automation

1. Connect ao servidor Jenkins na porta 8080 usando seu navegador preferido para acessar a interface de gerenciamento.
2. Digite a senha encontrada em `/var/lib/jenkins/secrets/initialAdminPassword`. Para exibir sua senha, execute o comando a seguir.

   ```
   sudo cat /var/lib/jenkins/secrets/initialAdminPassword
   ```
3. O script de instalação do Jenkins direciona você para a página Personalizar o Jenkins. Selecione Install suggested plugins (Instalar plugins sugeridos).
4. Uma vez concluída a instalação, escolha Credenciais de administrador, selecione Salvar credenciais e, depois, Começar a usar o Jenkins.
5. No painel de navegação à esquerda, escolha Gerenciar Jenkins e, em seguida, selecione Gerenciar plugins.
6. Selecione a guia Available (Disponível) e, em seguida, digite **`Amazon EC2 plugin`**.
7. Marque a caixa de seleção para **`Amazon EC2 plugin`**e, depois, selecione Install without restart (Instalar sem reiniciar).
8. Quando a instalação terminar, selecione Go back to the top page (Voltar para a página inicial).
9. Escolha Gerenciar Jenkins e, em seguida, Gerenciar nós e nuvens.
10. Na seção Configurar nuvens, selecione Adicionar uma nova nuvem e, em seguida, escolha Amazon EC2.
11. Insira suas informações nos campos restantes. Certifique-se de selecionar a opção Usar perfil de instância do EC2 para obter credenciais.

Use o procedimento a seguir para configurar o projeto do Jenkins para invocar o Automation.

Para configurar o servidor Jenkins para invocar o Automation

1. Abra o console do Jenkins em um navegador Web.
2. Escolha o projeto que deseja configurar com Automação e depois escolha Configure.
3. Na guia Build, escolha Add Build Step.
4. Escolha Execute shell ou Execute Windows batch command (dependendo do seu sistema operacional).

5. No campo Command (Comando), execute um comando da AWS CLI como o seguinte: Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

```
aws ssm start-automation-execution \
        --document-name runbook name \
        --region Região da AWS of your source AMI \
        --parameters runbook parameters
```

O comando de exemplo a seguir usa o runbook UpdateMyLatestWindowsAmi e o parâmetro `latestAmi` do Systems Manager criado em Atualize uma AMI dourada usando Automation, AWS Lambda e Parameter Store.

```
aws ssm start-automation-execution \
        --document-name UpdateMyLatestWindowsAmi \
        --parameters \
            "sourceAMIid='{{ssm:latestAmi}}'"
        --region region
```

No Jenkins, o comando se parece com exemplo mostrado na captura de tela a seguir.



6. No projeto do Jenkins, escolha Criar agora. Jenkins retorna uma saída semelhante ao exemplo a seguir.



Atualização de AMIs para grupos do Auto Scaling

O seguinte exemplo atualiza um grupo do Auto Scaling com uma AMI com patch recém-aplicado. Essa abordagem garante que as novas imagens sejam automaticamente disponibilizadas para diferentes ambientes de computação que usam grupos do Auto Scaling.

A etapa final da automação neste exemplo usa uma função do Python para criar um novo modelo de execução que usa a AMI que acabou de receber patch. Em seguida, o grupo do Auto Scaling é atualizado para usar o novo modelo de execução. Neste tipo de cenário de Auto Scaling, os usuários podem encerrar instâncias existentes no grupo do Auto Scaling para forçar a execução de uma nova instância que usa a nova imagem. Como alternativa, os usuários podem aguardar e permitir que eventos de aumento ou redução de escala aconteçam para executar instâncias mais recentes.

Antes de começar

Conclua as seguintes tarefas antes de começar este exemplo.

- Configure perfis do IAM para o Automation, uma ferramenta do AWS Systems Manager. O Systems Manager requer uma função de perfil da instância e um ARN da função de serviço para processar automações. Para ter mais informações, consulte Configurar a automação.

Criar o runbook PatchAMIAndUpdateASG

Use o procedimento a seguir para criar o runbook PatchAMIAndUpdateASG, que aplica patch à AMI que você especifica para o parâmetro SourceAMI. O runbook também atualiza o grupo do Auto Scaling para usar a AMI que recebeu o patch mais recente.

Para criar e executar o runbook

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.

2. No painel de navegação, escolha Documents.
3. Na lista suspensa Create document (Criar documento), escolha Automation (Automação).
4. No campo Name (Nome), insira **`PatchAMIAndUpdateASG`**.
5. Escolha a guia Editor e escolha Edit (Editar).
6. Escolha OK quando solicitado e exclua o conteúdo do espaço reservado no campo Document editor (Editor de documentos).
7. No campo Document editor (Editor de documentos), cole o seguinte conteúdo do runbook de exemplo YAML:

```
---
description: Systems Manager Automation Demo - Patch AMI and Update ASG
schemaVersion: '0.3'
assumeRole: '{{ AutomationAssumeRole }}'
parameters:
  AutomationAssumeRole:
    type: String
    description: '(Required) The ARN of the role that allows Automation to perform
 the actions on your behalf. If no role is specified, Systems Manager Automation
 uses your IAM permissions to execute this document.'
    default: ''
  SourceAMI:
    type: String
    description: '(Required) The ID of the AMI you want to patch.'
  SubnetId:
    type: String
    description: '(Required) The ID of the subnet where the instance from the
 SourceAMI parameter is launched.'
  SecurityGroupIds:
    type: StringList
    description: '(Required) The IDs of the security groups to associate with the
 instance launched from the SourceAMI parameter.'
  NewAMI:
    type: String
    description: '(Optional) The name of of newly patched AMI.'
    default: 'patchedAMI-{{global:DATE_TIME}}'
  TargetASG:
    type: String
    description: '(Required) The name of the Auto Scaling group you want to
 update.'
  InstanceProfile:
    type: String
```

```
    description: '(Required) The name of the IAM instance profile you want the
 source instance to use.'
  SnapshotId:
    type: String
    description: (Optional) The snapshot ID to use to retrieve a patch baseline
 snapshot.
    default: ''
  RebootOption:
    type: String
    description: '(Optional) Reboot behavior after a patch Install operation. If
 you choose NoReboot and patches are installed, the instance is marked as non-
compliant until a subsequent reboot and scan.'
    allowedValues:
      - NoReboot
      - RebootIfNeeded
    default: RebootIfNeeded
  Operation:
    type: String
    description: (Optional) The update or configuration to perform on the instance.
 The system checks if patches specified in the patch baseline are installed on the
 instance. The install operation installs patches missing from the baseline.
    allowedValues:
      - Install
      - Scan
    default: Install
mainSteps:
  - name: startInstances
    action: 'aws:runInstances'
    timeoutSeconds: 1200
    maxAttempts: 1
    onFailure: Abort
    inputs:
      ImageId: '{{ SourceAMI }}'
      InstanceType: m5.large
      MinInstanceCount: 1
      MaxInstanceCount: 1
      IamInstanceProfileName: '{{ InstanceProfile }}'
      SubnetId: '{{ SubnetId }}'
      SecurityGroupIds: '{{ SecurityGroupIds }}'
  - name: verifyInstanceManaged
    action: 'aws:waitForAwsResourceProperty'
    timeoutSeconds: 600
    inputs:
      Service: ssm
```

```
      Api: DescribeInstanceInformation
      InstanceInformationFilterList:
        - key: InstanceIds
          valueSet:
            - '{{ startInstances.InstanceIds }}'
      PropertySelector: '$.InstanceInformationList[0].PingStatus'
      DesiredValues:
        - Online
    onFailure: 'step:terminateInstance'
  - name: installPatches
    action: 'aws:runCommand'
    timeoutSeconds: 7200
    onFailure: Abort
    inputs:
      DocumentName: AWS-RunPatchBaseline
      Parameters:
        SnapshotId: '{{SnapshotId}}'
        RebootOption: '{{RebootOption}}'
        Operation: '{{Operation}}'
      InstanceIds:
        - '{{ startInstances.InstanceIds }}'
  - name: stopInstance
    action: 'aws:changeInstanceState'
    maxAttempts: 1
    onFailure: Continue
    inputs:
      InstanceIds:
        - '{{ startInstances.InstanceIds }}'
      DesiredState: stopped
  - name: createImage
    action: 'aws:createImage'
    maxAttempts: 1
    onFailure: Continue
    inputs:
      InstanceId: '{{ startInstances.InstanceIds }}'
      ImageName: '{{ NewAMI }}'
      NoReboot: false
      ImageDescription: Patched AMI created by Automation
  - name: terminateInstance
    action: 'aws:changeInstanceState'
    maxAttempts: 1
    onFailure: Continue
    inputs:
      InstanceIds:
```

```
        - '{{ startInstances.InstanceIds }}'
      DesiredState: terminated
  - name: updateASG
    action: 'aws:executeScript'
    timeoutSeconds: 300
    maxAttempts: 1
    onFailure: Abort
    inputs:
      Runtime: python3.8
      Handler: update_asg
      InputPayload:
        TargetASG: '{{TargetASG}}'
        NewAMI: '{{createImage.ImageId}}'
      Script: |-
        from __future__ import print_function
        import datetime
        import json
        import time
        import boto3

        # create auto scaling and ec2 client
        asg = boto3.client('autoscaling')
        ec2 = boto3.client('ec2')

        def update_asg(event, context):
            print("Received event: " + json.dumps(event, indent=2))

            target_asg = event['TargetASG']
            new_ami = event['NewAMI']

            # get object for the ASG we're going to update, filter by name of
 target ASG
            asg_query =
 asg.describe_auto_scaling_groups(AutoScalingGroupNames=[target_asg])
            if 'AutoScalingGroups' not in asg_query or not
 asg_query['AutoScalingGroups']:
                return 'No ASG found matching the value you specified.'

            # gets details of an instance from the ASG that we'll use to model the
 new launch template after
            source_instance_id = asg_query.get('AutoScalingGroups')[0]['Instances']
[0]['InstanceId']
            instance_properties = ec2.describe_instances(
                InstanceIds=[source_instance_id]
```

```
            )
            source_instance = instance_properties['Reservations'][0]['Instances']
[0]

            # create list of security group IDs
            security_groups = []
            for group in source_instance['SecurityGroups']:
                security_groups.append(group['GroupId'])

            # create a list of dictionary objects for block device mappings
            mappings = []
            for block in source_instance['BlockDeviceMappings']:
                volume_query = ec2.describe_volumes(
                    VolumeIds=[block['Ebs']['VolumeId']]
                )
                volume_details = volume_query['Volumes']
                device_name = block['DeviceName']
                volume_size = volume_details[0]['Size']
                volume_type = volume_details[0]['VolumeType']
                device = {'DeviceName': device_name, 'Ebs': {'VolumeSize':
 volume_size, 'VolumeType': volume_type}}
                mappings.append(device)

            # create new launch template using details returned from instance in
 the ASG and specify the newly patched AMI
            time_stamp = time.time()
            time_stamp_string =
 datetime.datetime.fromtimestamp(time_stamp).strftime('%m-%d-%Y_%H-%M-%S')
            new_template_name = f'{new_ami}_{time_stamp_string}'
            try:
                ec2.create_launch_template(
                    LaunchTemplateName=new_template_name,
                    LaunchTemplateData={
                        'BlockDeviceMappings': mappings,
                        'ImageId': new_ami,
                        'InstanceType': source_instance['InstanceType'],
                        'IamInstanceProfile': {
                            'Arn': source_instance['IamInstanceProfile']['Arn']
                        },
                        'KeyName': source_instance['KeyName'],
                        'SecurityGroupIds': security_groups
                    }
                )
            except Exception as e:
```

```
                return f'Exception caught: {str(e)}'
            else:
                # update ASG to use new launch template
                asg.update_auto_scaling_group(
                    AutoScalingGroupName=target_asg,
                    LaunchTemplate={
                        'LaunchTemplateName': new_template_name
                    }
                )
                return f'Updated ASG {target_asg} with new launch template
 {new_template_name} which uses AMI {new_ami}.'
outputs:
  - createImage.ImageId
```

8. Escolha Criar automação.
9. No painel de navegação, selecione Automation e Execute automation (Executar automação).
10. Na página Choose document (Escolher documento), escolha a guia Owned by me (Pertencem a mim).
11. Procure o runbook PatchAMIAndUpdateASG e selecione o botão no cartão PatchAMIAndUpdateASG.
12. Escolha Próximo.
13. Escolha Simple execution (Execução simples).
14. Especificar valores para os parâmetro de entrada. Verifique se o `SubnetId` e o `SecurityGroupIds` especificados permitem acesso aos endpoints públicos do Systems Manager ou aos seus endpoints de interface para o Systems Manager.
15. Clique em Executar.
16. Após a conclusão da automação, no console do Amazon EC2, escolha Auto Scaling e depois escolha Launch Templates (Modelos de execução). Verifique se você vê o novo modelo de execução e se ele usa a nova AMI.
17. Selecione Auto Scaling e, depois, escolhaAuto Scaling Groups (Grupos Auto Scaling). Verifique se o grupo do Auto Scaling usa o novo modelo de execução.
18. Encerre uma ou mais instâncias no grupo de Auto Scaling. As instâncias de substituição serão executadas usando a nova AMI.

Uso de runbooks de autoatendimento do AWS Support

Esta seção descreve como usar algumas das automações de autoatendimento criadas pela equipe do AWS Support. Essas automações ajudam você a gerenciar seus recursos da AWS.

Fluxos de trabalho do Automation de suporte

Os fluxos de trabalho do Automation de suporte (SAW) são runbooks de automação escritos e mantidos pela equipe do AWS Support. Esses runbooks ajudam a solucionar problemas comuns com seus recursos da AWS, monitoram e identificam proativamente os problemas da rede, coletam e analisam logs e muito mais.

Os runbooks SAW usam o prefixo do **`AWSSupport`**. Por exemplo, `AWSSupport-ActivateWindowsWithAmazonLicense`.

Além disso, os clientes de suporte Enterprise e Business da AWS também têm acesso a runbooks que usam o prefixo **`AWSPremiumSupport`**. Por exemplo, `AWSPremiumSupport-TroubleshootEC2DiskUsage`.

Para saber mais sobre o AWS Support, consulte Conceitos básicos do AWS Support.

Tópicos

- Executar a ferramenta EC2Rescue em instâncias inacessíveis
- Redefinir senhas e chaves SSH em instâncias do EC2

Executar a ferramenta EC2Rescue em instâncias inacessíveis

O EC2Rescue pode ajudar a diagnosticar e solucionar problemas em instâncias do Amazon Elastic Compute Cloud (Amazon EC2) para Linux e Windows Server. Você pode executar a ferramenta manualmente, conforme descrito em Usar o EC2Rescue para Linux Server e Usar o EC2Rescue para Windows Server. Ou você pode executar a ferramenta automaticamente usando o Systems Manager Automation e o runbook **`AWSSupport-ExecuteEC2Rescue`**. O Automation é uma ferramenta do AWS Systems Manager. O runbook **`AWSSupport-ExecuteEC2Rescue`** foi projetado para realizar uma combinação de ações do Systems Manager, ações do AWS CloudFormation e funções do Lambda que automatizam as etapas normalmente necessárias para usar o EC2Rescue.

Você pode usar o runbook **`AWSSupport-ExecuteEC2Rescue`** para solucionar problemas e potencialmente corrigir diferentes tipos de problemas do sistema operacional (SO). Não há suporte para instâncias com volumes raiz criptografados. Veja os seguintes tópicos para uma lista completa:

Windows: consulte Rescue Action em Usar o EC2Rescue para Windows Server com a linha de comando.

Linux e macOS: alguns módulos do EC2Rescue for Linux detectam e tentam corrigir problemas. Para obter mais informações, consulte a documentação `aws-ec2rescue-linux` para cada módulo no GitHub.

Como funciona

Solução de problemas de uma instância com o Automation e o runbook **`AWSSupport-ExecuteEC2Rescue`** funcionam da seguinte maneira:

- Você especifica o ID da instância inacessível e inicia o runbook.
- O sistema cria uma VPC temporária e, em seguida, executa uma série de funções do Lambda para configurar a VPC.
- O sistema identifica uma sub-rede para sua VPC temporária na mesma Zona de disponibilidade da sua instância original.
- O sistema executa uma instância temporária auxiliar do , habilitada para o SSM.
- O sistema interrompe sua instância original e cria um backup. Em seguida, atribui o volume raiz original à instância auxiliar.
- O sistema usa o Run Command para executar o EC2Rescue na instância auxiliar. O EC2Rescue identifica e tenta corrigir problemas no volume raiz original anexado. Ao terminar, o EC2Rescue anexa o volume raiz de volta à instância original.
- O sistema reinicia sua instância original e encerra a instância temporária. O sistema também encerra a VPC temporária e as funções Lambda criadas no início da automação.

Antes de começar

Antes de executar a automação a seguir:

- Copie o ID da instância inacessível. Você especificará esse ID no procedimento.
- Opcionalmente, colete o ID de uma sub-rede na mesma zona de disponibilidade como sua instância inacessível. A instância EC2Rescue será criada nessa sub-rede. Se você não especificar uma sub-rede, o Automation criará uma nova VPC temporária em sua Conta da AWS. Verifique se sua Conta da AWS tem pelo menos uma VPC disponível. Por padrão, você pode criar cinco VPCs em uma Região. Se você já tiver criado cinco VPCs na Região, a automação falhará sem

fazer alterações na sua instância. Para obter mais informações sobre as cotas da Amazon VPC, consulte VPC e sub-redes no Manual do usuário da Amazon VPC.

- Opcionalmente, você pode criar e especificar uma função do AWS Identity and Access Management (IAM) para o Automation. Se você não especificar essa função, a automação será executada no contexto do usuário que executou a automação.

Conceder permissões ao **AWSSupport-EC2Rescue** para realizar ações nas instâncias

O EC2Rescue precisa de permissão para realizar uma série de ações nas suas instâncias durante a automação. Essas ações invocam os serviços do AWS Lambda, IAM e Amazon EC2 para tentar corrigir problemas com as instâncias de forma segura. Se você tiver permissões em nível de administrador na sua Conta da AWS e/ou VPC, poderá executar a automação sem configurar permissões, conforme descrito nesta seção. Se não tiver permissões em nível de Administrador, você ou um administrador deverá configurar permissões usando uma das seguintes opções.

- Conceder permissões usando políticas do IAM
- Conceder permissões usando um modelo do AWS CloudFormation

Conceder permissões usando políticas do IAM

É possível anexar a política do IAM a seguir ao seu usuário, grupo ou perfil como uma política em linha, ou criar uma nova política gerenciada do IAM e anexá-la ao seu usuário, grupo ou perfil. Para obter mais informações sobre como adicionar uma política em linha ao seu usuário, grupo ou perfil, consulte Como trabalhar com políticas em linha. Para obter mais informações sobre como criar uma nova política gerenciada, consulte Como trabalhar com políticas gerenciadas.

> **Note**
>
> Se você criar uma nova política gerenciada do IAM, deverá também anexar a ela a política gerenciada AmazonSSMAutomationRole para que suas instâncias possam se comunicar com a API do Systems Manager.

Política do IAM para AWSSupport-EC2Rescue

Substitua *account ID* (ID da conta) por suas próprias informações.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "lambda:InvokeFunction",
                "lambda:DeleteFunction",
                "lambda:GetFunction"
            ],
            "Resource": "arn:aws:lambda:*:111122223333:function:AWSSupport-
EC2Rescue-*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::awssupport-ssm.*/*.template",
                "arn:aws:s3:::awssupport-ssm.*/*.zip"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "iam:CreateRole",
                "iam:CreateInstanceProfile",
                "iam:GetRole",
                "iam:GetInstanceProfile",
                "iam:PutRolePolicy",
                "iam:DetachRolePolicy",
                "iam:AttachRolePolicy",
                "iam:PassRole",
                "iam:AddRoleToInstanceProfile",
                "iam:RemoveRoleFromInstanceProfile",
                "iam:DeleteRole",
                "iam:DeleteRolePolicy",
                "iam:DeleteInstanceProfile"
            ],
            "Resource": [
```

```
                "arn:aws:iam::111122223333:role/AWSSupport-EC2Rescue-*",
                "arn:aws:iam::111122223333:instance-profile/AWSSupport-EC2Rescue-
*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "lambda:CreateFunction",
                "ec2:CreateVpc",
                "ec2:ModifyVpcAttribute",
                "ec2:DeleteVpc",
                "ec2:CreateInternetGateway",
                "ec2:AttachInternetGateway",
                "ec2:DetachInternetGateway",
                "ec2:DeleteInternetGateway",
                "ec2:CreateSubnet",
                "ec2:DeleteSubnet",
                "ec2:CreateRoute",
                "ec2:DeleteRoute",
                "ec2:CreateRouteTable",
                "ec2:AssociateRouteTable",
                "ec2:DisassociateRouteTable",
                "ec2:DeleteRouteTable",
                "ec2:CreateVpcEndpoint",
                "ec2:DeleteVpcEndpoints",
                "ec2:ModifyVpcEndpoint",
                "ec2:Describe*",
                "autoscaling:DescribeAutoScalingInstances"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Conceder permissões usando um modelo do AWS CloudFormation

O AWS CloudFormation automatiza o processo de criação de políticas e funções do IAM, usando um modelo pré-configurado. Use o procedimento a seguir para criar as funções e políticas do IAM necessárias para o Automation EC2Rescue, usando o AWS CloudFormation.

Para criar as funções e políticas do IAM necessárias para o EC2Rescue

1. Faça download de `AWSSupport-EC2RescueRole.zip` e extraia o arquivo `AWSSupport-EC2RescueRole.json` para um diretório em sua máquina local.
2. Se sua Conta da AWS estiver em uma partição especial, edite o modelo para alterar os valores do ARN para os valores da sua partição.

   Por exemplo, para as regiões da China, altere todos os casos de `arn:aws` para `arn:aws-cn`.
3. Faça login no AWS Management Console e abra o console AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
4. Escolha Create stack (Criar pilha), With new resources (Com novos recursos (padrão)).
5. Na página Create stack (Criar pilha), em Prerequisite - Prepare template (Pré-requisito – Preparar modelo), escolha Template is ready (O modelo está pronto).
6. Em Specify template (Especificar modelo), escolha Upload a template file (Fazer upload de um arquivo de modelo).
7. Escolha Choose file (Escolher arquivo), navegue até o arquivo `AWSSupport-EC2RescueRole.json` e selecione-o no diretório onde foi extraído.
8. Escolha Próximo.
9. Na página Specify stack details (Especificar detalhes da pilha), no campo Stack name (Nome da pilha), insira um nome para identificar essa pilha e escolha Next (Próximo).
10. (Opcional) Na área Tags, aplique um ou mais pares de nome/valor de chave de tag a pilha.

    Tags são metadados opcionais que você atribui a um recurso. Tags permitem categorizar um recurso de diferentes formas, como por finalidade, proprietário ou ambiente. Por exemplo, talvez você queira marcar uma pilha para identificar o tipo de tarefas que ela executa, os tipos de destinos ou outros recursos envolvidos e o ambiente em que ela é executada.
11. Escolha Next (Próximo).
12. Na página Review (Análise), role para baixo e escolha a opção I acknowledge that AWS CloudFormation might create IAM resources (Entendo que o poderá criar recursos do IAM).
13. Selecione Criar pilha.

    O AWS CloudFormation mostrará o status CREATE_IN_PROGRESS (CRIAÇÃO_EM_ANDAMENTO) por alguns minutos. O status mudará para CREATE_COMPLETE depois que a pilha tiver sido criada. Também é possível escolher o ícone de atualização para verificar o status do processo de criação.

14. Na lista de Stacks (Pilhas), escolha ao botão de opção ao lado da pilha que você acabou de criar e depois escolha a guia Outputs (Saídas).
15. Anote o Value (Valor). Este é o ARN de AssumeRole. Você especificará esse ARN ao executar a automação no próximo procedimento, Executar a Automação.

Executar a Automação

> **Important**
>
> A automação a seguir interrompe a instância inacessível. A interrupção da instância pode resultar em perda de dados em volumes de armazenamento de instâncias anexados (se presentes). A interrupção da instância também pode fazer com que o IP público seja alterado, caso nenhum IP elástico esteja associado.

Para executar a automação do **`AWSSupport-ExecuteEC2Rescue`**

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação à esquerda, escolha Automation (Automação).
3. Escolha Execute automation.
4. Na seção Automation document (Documento de automação), escolha Owned by Amazon (De propriedade da Amazon) na lista.
5. Na lista de runbooks, escolha o botão no cartão para o **`AWSSupport-ExecuteEC2Rescue`** e, depois, escolha Next (Próximo).
6. Na página Execute automation document (Executar documento de automação), escolha Simple execution (Execução simples).
7. Na seção Document details (Detalhes do documento), verifique se Versão do documento (Document version) está definida como a versão padrão mais recente. Por exemplo, $DEFAULT ou 3 (default) (3 (padrão)).
8. Na seção Input parameters, especifique os parâmetros a seguir:
   a. Em UnreachableInstanceId, especifique o ID da instância inacessível.
   b. (Opcional) Em EC2RescueInstanceType, especifique um tipo de instância para a instância EC2Rescue. O tipo de instância padrão é `t2.medium`.

c. Em AutomationAssumeRole, se você criou funções para esta automação usando o procedimento do AWS CloudFormation descrito anteriormente neste tópico, escolha o ARN de AssumeRole que você criou no console do AWS CloudFormation.

d. (Opcional) Em LogDestination, especifique um bucket do S3 se desejar coletar logs no nível do sistema operacional ao solucionar problemas da instância. Os logs são enviados automaticamente para o bucket especificado.

e. Em SubnetId, especifique uma sub-rede em uma VPC existente na mesma zona de disponibilidade da instância inacessível. Por padrão, o Systems Manager cria uma nova VPC, mas você pode especificar uma sub-rede em uma VPC existente, se quiser.

> **Note**
>
> Se a opção para especificar um bucket ou um ID de sub-rede não estiver disponível, verifique se você está usando a versão padrão mais recente do runbook.

9. (Opcional) Na área Tags aplique um ou mais pares de nome/valor de chave de tag para ajudar a identificar a automação, por exemplo, `Key=Purpose,Value=EC2Rescue`.
10. Clique em Executar.

O runbook cria uma AMI de backup como parte da automação. Todos os outros recursos criados pela automação são automaticamente excluídos, mas essa AMI permanece em sua conta. A AMI é chamada usando a convenção a seguir:

AMI de backup: AWSSupport-EC2Rescue:*UnreachableInstanceId*

Você pode localizar essa AMI no console do Amazon EC2 procurando o ID de execução do Automation.

Redefinir senhas e chaves SSH em instâncias do EC2

Você pode usar o runbook `AWSSupport-ResetAccess` para reativar automaticamente a geração de senhas do administrador local em instâncias do Amazon Elastic Compute Cloud (Amazon EC2) para o Windows Server e para gerar uma nova chave SSH nas instâncias do EC2 do Linux. O runbook `AWSSupport-ResetAccess` foi projetado para realizar uma combinação de ações do AWS Systems Manager, ações do AWS CloudFormation e funções do AWS Lambda que automatizam as etapas normalmente necessárias para redefinir a senha de administrador local.

Você pode usar o Automation, uma ferramenta do AWS Systems Manager, com o runbook `AWSSupport-ResetAccess` para resolver os seguintes problemas:

Windows

Você perdeu o par de chaves EC2: para resolver esse problema, você pode usar o runbook AWSSupport-ResetAccess para criar uma AMI ativada por senha de sua instância atual, inicie uma nova instância em uma AMI e selecione um par de chaves que você tenha.

Você perdeu a senha do administrador local: para resolver esse problema, você pode usar o runbook `AWSSupport-ResetAccess` para gerar uma nova senha que você pode descriptografar com o par de chaves EC2 atual.

Linux

Você perdeu seu par de chaves EC2 ou configurou o acesso SSH à instância com uma chave perdida: para resolver esse problema, você pode usar o runbook `AWSSupport-ResetAccess` para criar uma nova chave SSH para a instância atual, que permite se conectar à instância novamente.

> **Note**
>
> Se a instância do EC2 para Windows Server for configurada para o Systems Manager, você poderá também redefinir a senha de administrador local usando o EC2Rescue e AWS Systems Manager Run Command. Para obter mais informações, consulte Usar o EC2Rescue for Windows Server com o Run Command do Systems Manager no Guia do usuário do Amazon EC2.

Informações relacionadas

Conexão à sua instância do Linux via Windows usando o PuTTY no Guia do usuário do Amazon EC2

Como funciona

A solução de problemas em uma instância com o Automation e o runbook `AWSSupport-ResetAccess` funcionam da seguinte maneira:

- Você especifica o ID da instância e executa o runbook.

- O sistema cria uma VPC temporária e, em seguida, executa uma série de funções do Lambda para configurar a VPC.
- O sistema identifica uma sub-rede para sua VPC temporária na mesma Zona de disponibilidade da sua instância original.
- O sistema executa uma instância temporária auxiliar do , habilitada para o SSM.
- O sistema interrompe sua instância original e cria um backup. Em seguida, atribui o volume raiz original à instância auxiliar.
- O sistema usa o Run Command para executar o EC2Rescue na instância auxiliar. No Windows, o EC2Rescue permite a geração de senha para o administrador local usando o EC2Config ou EC2Launch no volume raiz original anexado. No Linux, o EC2Rescue gera e injeta uma nova chave SSH e salva a chave privada, criptografada em Parameter Store. Ao terminar, o EC2Rescue anexa o volume raiz de volta à instância original.
- O sistema cria uma nova Amazon Machine Image (AMI) de sua instância, agora que a geração de senha está habilitada. Você pode usar essa AMI para criar uma nova instância do EC2 e associar um novo par de chaves, se necessário.
- O sistema reinicia sua instância original e encerra a instância temporária. O sistema também encerra a VPC temporária e as funções Lambda criadas no início da automação.
- Windows: a instância gera uma nova senha que você pode decodificar no console do Amazon EC2 usando o par de chaves atual designado para a instância.

  Linux: você pode se conectar à instância via SSH usando a chave SSH armazenada no Systems Manager Parameter Store, como /ec2rl/openssh/***instance ID***/key.

Antes de começar

Antes de executar a automação a seguir:

- Copie o ID da instância na qual você deseja redefinir a senha de administrador. Você especificará esse ID no procedimento.
- Opcionalmente, colete o ID de uma sub-rede na mesma zona de disponibilidade como sua instância inacessível. A instância EC2Rescue será criada nessa sub-rede. Se você não especificar uma sub-rede, o Automation criará uma nova VPC temporária em sua Conta da AWS. Verifique se sua Conta da AWS tem pelo menos uma VPC disponível. Por padrão, você pode criar cinco VPCs em uma Região. Se você já tiver criado cinco VPCs na Região, a automação falhará sem fazer alterações na sua instância. Para obter mais informações sobre as cotas da Amazon VPC, consulte VPC e sub-redes no Manual do usuário da Amazon VPC.

- Opcionalmente, você pode criar e especificar uma função do AWS Identity and Access Management (IAM) para o Automation. Se você não especificar essa função, a automação será executada no contexto do usuário que executou a automação.

Conceder a AWSSupport-EC2Rescue permissões para realizar ações em suas instâncias

O EC2Rescue precisa de permissão para realizar uma série de ações nas suas instâncias durante a automação. Essas ações invocam os serviços do AWS Lambda, IAM e Amazon EC2 para tentar corrigir problemas com as instâncias de forma segura. Se você tiver permissões em nível de administrador na sua Conta da AWS e/ou VPC, poderá executar a automação sem configurar permissões, conforme descrito nesta seção. Se não tiver permissões em nível de Administrador, você ou um administrador deverá configurar permissões usando uma das seguintes opções.

- Conceder permissões usando políticas do IAM
- Conceder permissões usando um modelo do AWS CloudFormation

Conceder permissões usando políticas do IAM

É possível anexar a política do IAM a seguir ao seu usuário, grupo ou perfil como uma política em linha, ou criar uma nova política gerenciada do IAM e anexá-la ao seu usuário, grupo ou perfil. Para obter mais informações sobre como adicionar uma política em linha ao seu usuário, grupo ou perfil, consulte Como trabalhar com políticas em linha. Para obter mais informações sobre como criar uma nova política gerenciada, consulte Como trabalhar com políticas gerenciadas.

> **Note**
>
> Se você criar uma nova política gerenciada do IAM, deverá também anexar a ela a política gerenciada AmazonSSMAutomationRole para que suas instâncias possam se comunicar com a API do Systems Manager.

Política do IAM para**`AWSSupport-ResetAccess`**

Substitua *`account ID`* (ID da conta) por suas próprias informações.

JSON

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "lambda:InvokeFunction",
                "lambda:DeleteFunction",
                "lambda:GetFunction"
            ],
            "Resource": "arn:aws:lambda:*:111122223333:function:AWSSupport-
EC2Rescue-*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::awssupport-ssm.*/*.template",
                "arn:aws:s3:::awssupport-ssm.*/*.zip"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "iam:CreateRole",
                "iam:CreateInstanceProfile",
                "iam:GetRole",
                "iam:GetInstanceProfile",
                "iam:PutRolePolicy",
                "iam:DetachRolePolicy",
                "iam:AttachRolePolicy",
                "iam:PassRole",
                "iam:AddRoleToInstanceProfile",
                "iam:RemoveRoleFromInstanceProfile",
                "iam:DeleteRole",
                "iam:DeleteRolePolicy",
                "iam:DeleteInstanceProfile"
            ],
            "Resource": [
                "arn:aws:iam::111122223333:role/AWSSupport-EC2Rescue-*",
                "arn:aws:iam::111122223333:instance-profile/AWSSupport-EC2Rescue-
*"
            ],
```

```
            "Effect": "Allow"
        },
        {
            "Action": [
                "lambda:CreateFunction",
                "ec2:CreateVpc",
                "ec2:ModifyVpcAttribute",
                "ec2:DeleteVpc",
                "ec2:CreateInternetGateway",
                "ec2:AttachInternetGateway",
                "ec2:DetachInternetGateway",
                "ec2:DeleteInternetGateway",
                "ec2:CreateSubnet",
                "ec2:DeleteSubnet",
                "ec2:CreateRoute",
                "ec2:DeleteRoute",
                "ec2:CreateRouteTable",
                "ec2:AssociateRouteTable",
                "ec2:DisassociateRouteTable",
                "ec2:DeleteRouteTable",
                "ec2:CreateVpcEndpoint",
                "ec2:DeleteVpcEndpoints",
                "ec2:ModifyVpcEndpoint",
                "ec2:Describe*"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Conceder permissões usando um modelo do AWS CloudFormation

O AWS CloudFormation automatiza o processo de criação de políticas e funções do IAM, usando um modelo pré-configurado. Use o procedimento a seguir para criar as funções e políticas do IAM necessárias para o Automation EC2Rescue, usando o AWS CloudFormation.

Para criar as funções e políticas do IAM necessárias para o EC2Rescue

1. Faça download de AWSSupport-EC2RescueRole.zip e extraia o arquivo `AWSSupport-EC2RescueRole.json` para um diretório em sua máquina local.

2. Se sua Conta da AWS estiver em uma partição especial, edite o modelo para alterar os valores do ARN para os valores da sua partição.

   Por exemplo, para as regiões da China, altere todos os casos de `arn:aws` para `arn:aws-cn`.

3. Faça login no AWS Management Console e abra o console AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
4. Escolha Create stack (Criar pilha), With new resources (Com novos recursos (padrão)).
5. Na página Create stack (Criar pilha), em Prerequisite - Prepare template (Pré-requisito – Preparar modelo), escolha Template is ready (O modelo está pronto).
6. Em Specify template (Especificar modelo), escolha Upload a template file (Fazer upload de um arquivo de modelo).
7. Escolha Choose file (Escolher arquivo), navegue até o arquivo `AWSSupport-EC2RescueRole.json` e selecione-o no diretório onde foi extraído.
8. Escolha Próximo.
9. Na página Specify stack details (Especificar detalhes da pilha), no campo Stack name (Nome da pilha), insira um nome para identificar essa pilha e escolha Next (Próximo).
10. (Opcional) Na área Tags, aplique um ou mais pares de nome/valor de chave de tag a pilha.

    Tags são metadados opcionais que você atribui a um recurso. Tags permitem categorizar um recurso de diferentes formas, como por finalidade, proprietário ou ambiente. Por exemplo, talvez você queira marcar uma pilha para identificar o tipo de tarefas que ela executa, os tipos de destinos ou outros recursos envolvidos e o ambiente em que ela é executada.

11. Escolha Next (Próximo).
12. Na página Review (Análise), role para baixo e escolha a opção I acknowledge that AWS CloudFormation might create IAM resources (Entendo que o poderá criar recursos do IAM).
13. O AWS CloudFormation mostrará o status CREATE_IN_PROGRESS (CRIAÇÃO_EM_ANDAMENTO) por alguns minutos. O status mudará para CREATE_COMPLETE depois que a pilha tiver sido criada. Também é possível escolher o ícone de atualização para verificar o status do processo de criação.
14. Na lista de pilhas, escolha a opção ao lado da pilha que você acabou de criar e selecione a guia Outputs (Saídas).
15. Copie o conteúdo em Value. Este é o ARN de AssumeRole. Você especificará esse ARN quando executar a automação.

Executar a Automação

O procedimento a seguir descreve como executar o runbook `AWSSupport-ResetAccess` usando o console do AWS Systems Manager.

> **Important**
>
> A automação a seguir interrompe a instância. A interrupção da instância pode resultar em perda de dados em volumes de armazenamento de instâncias anexados (se presentes). A interrupção da instância também pode fazer com que o IP público seja alterado, caso nenhum IP elástico esteja associado. Para evitar essas alterações de configuração, use o Run Command para redefinir o acesso. Para obter mais informações, consulte Usar o EC2Rescue para Windows Server com o Run Command do Systems Manager no Guia do usuário do Amazon EC2.

Para executar a automação AWSSupport-ResetAccess

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação à esquerda, escolha Automation (Automação).
3. Escolha Execute automation.
4. Na seção Automation document (Documento de automação), escolha Owned by Amazon (De propriedade da Amazon) na lista.
5. Na lista de runbooks, escolha o botão no cartão para AWSSupport-ResetAccess e escolha Next (Próximo).
6. Na página Execute automation document (Executar documento de automação), escolha Simple execution (Execução simples).
7. Na seção Document details (Detalhes do documento), verifique se Versão do documento (Document version) está definida como a versão padrão mais recente. Por exemplo, $DEFAULT ou 3 (default) (3 (padrão)).
8. Na seção Input parameters, especifique os parâmetros a seguir:
   a. Em InstanceID, especifique o ID da instância inacessível.
   b. Em SubnetId, especifique uma sub-rede em uma VPC existente na mesma zona de disponibilidade da instância que você especificou. Por padrão, o Systems Manager cria uma nova VPC, mas você pode especificar uma sub-rede em uma VPC existente, se quiser.

> **Note**
>
> Se a opção para especificar um ID de sub-rede não estiver disponível, verifique se você está usando a versão padrão mais recente do runbook.

c. Em EC2RescueInstanceType, especifique um tipo de instância para a instância EC2Rescue. O tipo de instância padrão é `t2.medium`.

d. Em AssumeRole, se você criou funções para essa automação usando o procedimento do AWS CloudFormation descrito anteriormente neste tópico, especifique o ARN de AssumeRole que você anotou no console do AWS CloudFormation.

9. (Opcional) Na área Tags aplique um ou mais pares de nome/valor de chave de tag para ajudar a identificar a automação, por exemplo, `Key=Purpose,Value=ResetAccess`.
10. Clique em Executar.
11. Para monitorar o progresso da automação, escolha a automação em execução e depois escolha a guia Steps (Etapas). Quando a automação for concluída, escolha a guia Descriptions (Descrições) e, em seguida, View output (Exibir resultados) para visualizar os resultados. Para exibir a saída de etapas individuais, selecione a guia Steps (Etapas) e selecione View Outputs (Visualizar saídas) ao lado de uma etapa.

O runbook cria uma AMI de backup e uma AMI ativada por senha como parte da automação. Todos os outros recursos criados pela automação são automaticamente excluídos, mas essas AMIs permanecem em sua conta. As AMIs são nomeadas usando as seguintes convenções:

- AMI de backup: `AWSSupport-EC2Rescue:`*`InstanceID`*
- AMI ativada por senha: AWSSupport-EC2Rescue: AMI ativada por senha de *`Instance ID`*

Você pode localizar essas AMIs procurando o ID de execução do Automation.

No Linux, a nova chave privada SSH para sua instância é salva, criptografada, Parameter Store. O nome do parâmetro é /ec2rl/openssh/***`Instance ID`***/key.

Transferência de dados para o Automation usando transformadores de entrada

Este tutorial do AWS Systems Manager Automation mostra como usar o recurso transformador de entrada do Amazon EventBridge para extrair o `instance-id` de uma instância do Amazon Elastic Compute Cloud (Amazon EC2) de um evento de alteração de estado da instância. O Automation

é uma ferramenta do AWS Systems Manager. Usamos o transformador de entrada para passar esses dados ao destino do runbook do `AWS-CreateImage`, como o parâmetro de entrada do `InstanceId`. A regra será acionada quando qualquer instância for alterada para o estado `stopped`.

Para obter mais informações sobre como trabalhar com transformadores de entrada, consulte Tutorial: Use o transformador de entrada para personalizar o que o EventBridge passa para o destino do evento no Manual do usuário do Amazon EventBridge.

Antes de começar

Verifique se adicionou as permissões necessárias e a política de confiança para o EventBridge para a função de serviço do Systems Manager Automation. Para obter mais informações, consulte Visão geral do gerenciamento de permissões de acesso aos recursos do EventBridge no Manual do usuário do Amazon EventBridge.

Como usar transformadores de entrada com automação

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Regras.
3. Escolha Create rule.
4. Insira um nome e uma descrição para a regra.

   Uma regra não pode ter o mesmo nome que outra na mesma Região e barramento de eventos.
5. Em Barramento de eventos, selecione o barramento de eventos que você deseja associar a essa regra. Se você quiser que essa regra responda a eventos correspondentes provenientes da sua Conta da AWS, selecione default (padrão). Quando um AWS service (Serviço da AWS) na sua conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta.
6. Em Tipo de Regra, escolha Regra com Padrão de Evento.
7. Escolha Próximo.
8. Em Event source, escolha Eventos da AWS ou eventos de parceiro do EventBridge.
9. Na seção Event patter (Padrão de evento), selecione Event pattern form (Formulário de padrão de evento).
10. Em Event source (Origem do evento), escolha AWS services (Serviços da).
11. Em Serviço da AWS, escolha EC2.
12. Para Event Type (Tipo de evento), escolha EC2 Instance State-change Notification (Notificação de alteração de estado da instância do EC2).

13. Para Specific state(s) (Estados específicos), escolha stopped (parado).
14. Escolha Next (Avançar).
15. Em Tipos de destino, escolha Serviço da AWS.
16. Para Select a target (Selecionar um destino), escolha Systems Manager Automation (Automation do Systems Manager).
17. Em Document (Documento), escolha AWS-CreateImage.
18. Na seção Configure automation parameter(s) (Configurar parâmetros de automação), escolha Input Transformer (Transformador de entrada).
19. Para Input path (Caminho de entrada), insira **`{"instance":"$.detail.instance-id"}`**.
20. Para Template (Modelo), insira **`{"InstanceId":[<instance>]}`**.
21. Para Execution role, escolha Use existing role (Usar função existente) e escolha sua função de serviço do Automation.
22. Escolha Próximo.
23. (Opcional) Insira uma ou mais tags para a regra. Para obter mais informações, consulte Marcar recursos do Amazon EventBridge no Guia do usuário do Amazon EventBridge.
24. Escolha Próximo.
25. Analise os detalhes da regra e selecione Criar regra.

Saiba mais sobre os status retornados pelo Systems Manager Automation

O AWS Systems Manager Automation relata informações detalhadas de status sobre os vários status que uma ação ou etapa de automação passa quando você executa uma automação e para a automação geral. O Automation é uma ferramenta do AWS Systems Manager. Você pode monitorar os status da automação usando os seguintes métodos:

- Monitore o Execution status (Status da execução) no console do Systems Manager Automation.
- Use suas ferramentas de linha de comando preferidas. Para a AWS Command Line Interface (AWS CLI), você pode usar describe-automation-step-executions ou get-automation-execution. Para o AWS Tools for Windows PowerShell, você pode usar Get -SsMAutoMationStepExecution ou Get -SsMAutoMationExecution.
- Configure o Amazon EventBridge para responder a alterações de status de ação ou automação.

Para obter mais informações sobre como processar tempos limite em uma automação, consulte Gerenciar tempos limite em runbooks.

Sobre status de automação

Os relatórios de automação relatam detalhes das ações de automação individuais, além da automação geral.

O status geral da automação pode ser diferente do status relatado por uma ação individual ou etapa, conforme observado nas tabelas a seguir.

Status detalhado das ações

Status	Detalhes
Pendente	A execução da etapa ainda não começou. Se a automação usar ações condicionais, as etapas permanecerão nesse estado após a conclusão de uma automação se a condição não tiver sido atendida para executar a etapa. As etapas também permanecem nesse estado se a automação for cancelada antes da execução da etapa.
InProgress	A etapa está em execução.
Aguardando	A etapa está aguardando inserção de informações.
Bem-sucedida	A etapa foi concluída com êxito. Este é um estado terminal.
TimedOut	Uma etapa ou aprovação não foi concluída antes do período de tempo limite especificado. Este é um estado terminal.
Cancelando	A etapa está em processo de interrupção após ser cancelada por um solicitante.
Cancelado	A etapa foi interrompida por um solicitante antes de ser concluída. Este é um estado terminal.

Status	Detalhes
Falha	A etapa não foi concluída com êxito. Este é um estado terminal.
Exited	Retornado somente pela ação `aws:loop`. O loop não foi totalmente concluído. Uma etapa dentro do loop foi movida para uma etapa externa usando as propriedades `nextStep`, `onCancel` ou `onFailure` .

Status detalhado de uma automação

Status	Detalhes
Pendente	A automação ainda não começou a ser executada.
InProgress	A automação está em execução.
Aguardando	A automação está aguardando a entrada.
Bem-sucedida	A automação foi concluída com êxito. Este é um estado terminal.
TimedOut	Uma etapa ou aprovação não foi concluída antes do período de tempo limite especificado. Este é um estado terminal.
Cancelando	A automação está em processo de interrupção após ser cancelada por um solicitante.
Cancelado	A automação foi interrompida por um solicitan te antes de ser concluída. Este é um estado terminal.
Falha	A automação não foi concluída com êxito. Este é um estado terminal.

Solução de problemas do Systems Manager Automation

Use as informações a seguir para obter ajuda para solucionar problemas com o AWS Systems Manager Automation, uma ferramenta do AWS Systems Manager. Este tópico inclui tarefas específicas para resolver problemas com base em mensagens de erro de Automação.

Tópicos

- Erros comuns de automação
- Falha ao iniciar a execução da automação
- Execução iniciada, mas o status falhou
- Execução iniciada, mas tempo limite atingido

Erros comuns de automação

Esta seção inclui informações sobre erros comuns de Automação.

VPC não definida 400

Por padrão, quando o Automation executa o runbook `AWS-UpdateLinuxAmi` ou `AWS-UpdateWindowsAmi`, o sistema cria uma instância temporária na VPC padrão (172.30.0.0/16). Se tiver excluído a VPC padrão, você receberá o seguinte erro:

```
VPC not defined 400
```

Para resolver esse problema, você deve especificar um valor para o parâmetro de entrada `SubnetId`.

Falha ao iniciar a execução da automação

Uma automação pode apresentar falhas com um erro de acesso negado ou um erro de perfil assumido inválido, se você não tiver configurado corretamente as políticas e os perfis do AWS Identity and Access Management (IAM) para o Automation.

Acesso negado

Os exemplos a seguir descrevem situações em que uma automação não foi iniciada, sinalizando um erro de acesso negado.

Acesso negado à API do Systems Manager

A mensagem de erro: `User: user arn isn't authorized to perform: ssm:StartAutomationExecution on resource: document arn (Service: AWSSimpleSystemsManagement; Status Code: 400; Error Code: AccessDeniedException; Request ID: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx)`

- Causa possível 1: o usuário que está tentando iniciar a automação não tem permissões para invocar a API `StartAutomationExecution`. Para resolver esse problema, anexe a política do IAM requerida ao usuário que foi usado para iniciar a automação.
- Causa possível 2: o usuário que está tentando iniciar a automação tem permissões para invocar a API `StartAutomationExecution`, mas não tem permissões para invocar a API usando o runbook específico. Para resolver esse problema, anexe a política do IAM requerida ao usuário que foi usado para iniciar a automação.

Acesso negado por ausência de permissões PassRole

A mensagem de erro: `User: user arn isn't authorized to perform: iam:PassRole on resource: automation assume role arn (Service: AWSSimpleSystemsManagement; Status Code: 400; Error Code: AccessDeniedException; Request ID: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx)`

O usuário que está tentando iniciar a automação não tem permissões PassRole para assumir o perfil. Para resolver esse problema, anexe a política iam:PassRole ao perfil do usuário que está tentando iniciar a automação. Para ter mais informações, consulte Tarefa 2: Anexar a política iam:PassRole à função de automação.

Função assumida inválida

Quando você executa um Automation, uma função assumida é fornecida no runbook ou transmitida como um valor de parâmetro para o runbook. Diferentes tipos de erros poderão ocorrer se a função assumida não for especificada ou configurada corretamente.

Função de admissão malformada

Mensagem de erro: `The format of the supplied assume role ARN isn't valid.` A função de admissão está mal formatada. Para resolver esse problema, verifique se uma função assumida válida está especificada no seu runbook ou como um parâmetro em runtime ao executar a automação.

Não é possível assumir o perfil assumido

A mensagem de erro: `The defined assume role is unable to be assumed. (Service: AWSSimpleSystemsManagement; Status Code: 400; Error Code: InvalidAutomationExecutionParametersException; Request ID: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx)`

- Causa possível 1: a função assumida não existe. Para resolver esse problema, crie a função. Para ter mais informações, consulte the section called “Configurar a automação”. Detalhes específicos para a criação dessa função estão descritos no seguinte tópico, Tarefa 1: Criar uma função de serviço para a automação.
- Causa possível 2: a função assumida não possui uma relação de confiança com o serviço do Systems Manager. Para resolver esse problema, crie a relação de confiança. Para obter mais informações, consulte Não consigo assumir uma função no Manual do usuário do IAM.

Execução iniciada, mas o status falhou

Falhas específicas da ação

Runbooks contêm etapas e elas são executadas em ordem. Cada etapa invoca uma ou mais APIs de AWS service (Serviço da AWS). Essas APIs determinam as entradas, o comportamento e as saídas da etapa. Há vários locais em que um erro pode causar uma falha na etapa. As mensagens de falha indicam quando e onde um erro ocorreu.

Para ver uma mensagem de falha no console do Amazon Elastic Compute Cloud (Amazon EC2), escolha o link View Outputs (Exibir resultados) da etapa com falha. Para ver uma mensagem de falha da AWS CLI, chame `get-automation-execution` e procure o atributo `FailureMessage` em um `StepExecution` com falha.

Nos exemplos a seguir, uma etapa associada à ação `aws:runInstance` falhou. Cada exemplo explora um tipo diferente de erro.

Imagem ausente

A mensagem de erro: `Automation Step Execution fails when it's launching the instance(s). Get Exception from RunInstances API of ec2 Service. Exception Message from RunInstances API: [The image id '[ami id]' doesn't exist (Service: AmazonEC2; Status Code: 400; Error Code: InvalidAMIID.NotFound; Request ID: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx)]. Please refer to Automation Service Troubleshooting Guide for more diagnosis details.`

A ação `aws:runInstances` recebeu uma entrada para um `ImageId` que não existe. Para resolver esse problema, atualize o runbook ou os valores de parâmetros com o ID correto da AMI.

A política do perfil assumido não tem permissões suficientes

A mensagem de erro: `Automation Step Execution fails when it's launching the instance(s). Get Exception from RunInstances API of ec2 Service. Exception Message from RunInstances API: [You aren't authorized to perform this operation. Encoded authorization failure message: xxxxxxx (Service: AmazonEC2; Status Code: 403; Error Code: UnauthorizedOperation; Request ID: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx)]. Please refer to Automation Service Troubleshooting Guide for more diagnosis details.`

A função de assunção não tem permissão suficiente para invocar a API `RunInstances` em instâncias do EC2. Para resolver esse problema, anexe uma política do IAM à função assumida que tenha permissão para invocar a API `RunInstances`. Para obter mais informações, consulte Criar perfis de serviço para o Automation usando o console.

Estado inesperado

A mensagem de erro: `Step fails when it's verifying launched instance(s) are ready to be used. Instance i-xxxxxxxxx entered unexpected state: shutting-down. Please refer to Automation Service Troubleshooting Guide for more diagnosis details.`

- Causa possível 1: há um problema com a instância ou o serviço do Amazon EC2. Para resolver esse problema, faça login na instância ou revise o log do sistema da instância para entender por que ela iniciou o desligamento.
- Causa possível 2: o script de dados do usuário especificado para a ação `aws:runInstances` tem um problema ou uma sintaxe incorreta. Verifique a sintaxe do script de dados do usuário. Além disso, verifique se os scripts de dados do usuário não desligam a instância ou invocam outros scripts que desligam a instância.

Referência a falhas específicas de ação

Quando uma etapa falha, a mensagem de falha pode indicar qual serviço foi invocado quando a falha ocorreu. A tabela a seguir lista os serviços invocados por cada ação. Ela também fornece links para obter informações sobre cada serviço.

Ação	Serviços da AWS invocados por essa ação	Para obter informaçõ es sobre este serviço	Solucionar problemas de conteúdo
`aws:runInstances`	Amazon EC2	Guia do usuário do Amazon EC2	Solucionar problemas com instâncias do EC2
`aws:changeInstanceState`	Amazon EC2	Guia do usuário do Amazon EC2	Solucionar problemas com instâncias do EC2
`aws:runCommand`	Systems Manager	AWS Systems Manager Run Command	Solução de problemas do Run Commando do Systems Manager
`aws:createImage`	Amazon EC2	Amazon Machine Images	
`aws:createStack`	AWS CloudFormation	Guia do usuário do AWS CloudFormation	Resolução de problemasAWS CloudFormation
`aws:deleteStack`	AWS CloudFormation	Guia do usuário do AWS CloudFormation	Resolução de problemasAWS CloudFormation
`aws:deleteImage`	Amazon EC2	Imagens de máquina da Amazon	
`aws:copyImage`	Amazon EC2	Amazon Machine Images	
`aws:createTag`	Amazon EC2, Systems Manager	Recurso e etiquetas do EC2	
`aws:invokeLambdaFunction`	AWS Lambda	AWS Lambda Guia do desenvolvedor	Solução de problemas do Lambda

Erro interno do serviço de automação

A mensagem de erro: `Internal Server Error. Please refer to Automation Service Troubleshooting Guide for more diagnosis details.`

Um problema com o serviço Automation está impedindo que o runbook especificado seja executado corretamente. Para resolver esse problema, entre em contato com o AWS Support. Forneça o ID de execução e o ID de cliente, se disponíveis.

Execução iniciada, mas tempo limite atingido

A mensagem de erro: `Step timed out while step is verifying launched instance(s) are ready to be used. Please refer to Automation Service Troubleshooting Guide for more diagnosis details.`

Uma etapa na ação `aws:runInstances` atingiu o tempo limite. Isso pode acontecer se a ação da etapa demorar mais para ser executada do que o valor especificado para `timeoutSeconds` na etapa. Para resolver esse problema, especifique um valor mais longo para o parâmetro `timeoutSeconds` da ação `aws:runInstances`. Se isso não resolver o problema, investigue por que a etapa demora mais para ser executada do que o esperado

AWS Systems Manager Change Calendar

O Change Calendar, uma ferramenta do AWS Systems Manager, permite configurar intervalos de data e hora quando as ações especificadas (por exemplo, em runbooks do Systems Manager Automation) podem ou não ser executadas em sua Conta da AWS. No Change Calendar, esses intervalos são chamados de eventos. Ao criar uma entrada do Change Calendar, você está criando um documento do Systems Manager do tipo `ChangeCalendar`. No Change Calendar, o documento armazena dados do iCalendar 2.0 em formato de texto simples. Os eventos adicionados à entrada do Change Calendar tornam-se parte do documento. Para começar a usar o Change Calendar, abra o Systems Manager console (Console do gerenciador de sistemas). No painel de navegação, escolha Change Calendar.

Você pode criar um calendário e seus eventos no console do Systems Manager. Você pode também importar um arquivo do iCalendar (`.ics`) que você exportou de um provedor de calendário de terceiros compatível, para adicionar seus eventos ao calendário. Os provedores compatíveis incluem o Google Agenda, o Microsoft Outlook e o Calendário do iCloud.

Uma entrada do Change Calendar pode ser de dois tipos:

`DEFAULT_OPEN` ou Aberto por padrão

Todas as ações podem ser executadas por padrão, exceto durante eventos do calendário. Durante eventos, o estado de um calendário `DEFAULT_OPEN` é `CLOSED` e os eventos são bloqueados para execução.

`DEFAULT_CLOSED` ou Fechado por padrão

Todas as ações são bloqueadas por padrão, exceto durante eventos do calendário. Durante eventos, o estado de um calendário `DEFAULT_CLOSED` é `OPEN` e as ações podem ser executadas.

Você pode optar por ter todos os fluxos de trabalho do Automation, janelas de manutenção e associações do State Manager agendados adicionados automaticamente a um calendário. Também é possível remover qualquer um desses tipos individuais da exibição de calendário.

Quem deve usar o Change Calendar?

- Clientes da AWS que realizam os seguintes tipos de ação:
  - Crie ou execute runbooks do Automation.
  - Crie solicitações de alteração no Change Manager.
  - Executar janelas de manutenção.
  - Crie associações no State Manager.

  Automation, Change Manager, Maintenance Windows e State Manager são ferramentas do AWS Systems Manager. Ao integrar essas ferramentas ao Change Calendar, você poderá permitir ou bloquear esses tipos de ações dependendo do estado atual do calendário de alterações que associar a cada um deles.
- Administradores responsáveis por manter as configurações dos nós gerenciados pelo Systems Manager consistentes, estáveis e funcionais.

Benefícios do Change Calendar

Veja a seguir alguns benefícios do Change Calendar.

- Revisar as alterações antes de serem aplicadas

Uma entrada do Change Calendar pode ajudar a garantir que alterações potencialmente destrutivas em seu ambiente sejam analisadas antes de serem aplicadas.

- Aplicar alterações somente nas horas apropriadas

  O Change Calendar ajuda a manter seu ambiente estável durante os horários dos eventos. Por exemplo, é possível criar uma entrada do Change Calendar para bloquear alterações quando você espera uma alta demanda em seus recursos, como durante uma conferência ou promoção de marketing pública. Uma entrada de calendário também pode bloquear alterações quando você espera suporte limitado do administrador, como durante férias ou feriados. É possível usar uma entrada de calendário para permitir alterações, exceto em determinadas horas do dia ou da semana, quando há suporte limitado do administrador para solucionar problemas de ações ou implantações com falha.

- Obter o estado atual ou futuro do calendário

  Você pode executar a operação da API `GetCalendarState` do Systems Manager para mostrar o estado atual do calendário, o estado em uma hora especificada ou a próxima vez que o estado do calendário estiver programado para alterar.

  > **Note**
  >
  > A API `GetCalendarState` tem uma cota de dez solicitações por segundo. Para obter informações sobre as cotas do System Manager, consulte Systems Manager service quotas no Referência geral da Amazon Web Services.

- Suporte ao EventBridge

  Essa ferramenta do Systems Manager pode ser usada como um tipo de evento nas regras do Amazon EventBridge. Para obter informações, consulte Monitorar eventos do Systems Manager com o Amazon EventBridge e Referência: padrões e tipos de eventos do Amazon EventBridge para o Systems Manager.

Tópicos

- Configurar o Change Calendar
- Trabalhar com o Change Calendar
- Adicionar dependências do Change Calendar para runbooks do Automation
- Solução de problemas do Change Calendar

Configurar o Change Calendar

Conclua o seguinte antes de usar o Change Calendar, uma ferramenta do AWS Systems Manager.

Instalar as ferramentas de linha de comando mais recentes

Instale as ferramentas de linha de comando mais recentes para obter informações de estado sobre calendários.

Requisito	Descrição
AWS CLI	(Opcional) Para usar a AWS Command Line Interface (AWS CLI) para obter informações de estado sobre calendários, instale a versão mais recente da AWS CLI no computador local. Para obter mais informações sobre como instalar ou atualizar a CLI, consulte Instalar, atualizar e desinstalar a AWS CLI no Guia do usuário da AWS Command Line Interface.
Ferramentas da AWS para PowerShell	(Opcional) Para usar o Tools for PowerShel l para obter informações de estado sobre calendários, instale a versão mais recente do Tools for PowerShell no computador local. Para obter mais informações sobre como instalar ou atualizar a Tools for PowerShell, consulte Instalar a Ferramentas da AWS para PowerShell no Guia do usuário da Ferramentas da AWS para PowerShell.

Configurar permissões

Se seu usuário, grupo ou perfil tiver permissões de administrador atribuídas, você terá acesso total ao Change Calendar. Se você não tiver permissões de administrador, um administrador deverá conceder as permissões ao atribuir a política gerenciada `AmazonSSMFullAccess` ou ao atribuir uma política que forneça as permissões necessárias para seu usuário, grupo ou perfil.

As seguintes permissões são necessárias para trabalhar com o Change Calendar.

Entradas do Change Calendar

Para criar, atualizar ou excluir uma entrada do Change Calendar, incluindo adicionar e remover eventos da entrada, uma política anexada ao usuário, perfil ou grupo deve permitir as seguintes ações:

- `ssm:CreateDocument`
- `ssm:DeleteDocument`
- `ssm:DescribeDocument`
- `ssm:DescribeDocumentPermission`
- `ssm:GetCalendar`
- `ssm:ListDocuments`
- `ssm:ModifyDocumentPermission`
- `ssm:PutCalendar`
- `ssm:UpdateDocument`
- `ssm:UpdateDocumentDefaultVersion`

Estado do calendário

Para obter informações sobre o estado atual ou futuro do calendário, uma política anexada ao usuário, grupo ou perfil deve permitir a seguinte ação:

- `ssm:GetCalendarState`

Eventos operacionais

Para visualizar eventos operacionais, como janelas de manutenção, associações e automações planejadas, a política anexada ao usuário, grupo ou perfil deve permitir estas ações:

- `ssm:DescribeMaintenanceWindows`
- `ssm:DescribeMaintenanceWindowExecution`
- `ssm:DescribeAutomationExecutions`
- `ssm:ListAssociations`

> **Note**
>
> As entradas do Change Calendar que são de propriedade das (ou seja, criadas por) contas que não sejam suas são somente leitura, mesmo que compartilhadas com sua conta. Janelas de manutenção, associações do State Manager e automações não são compartilhadas.

Trabalhar com o Change Calendar

É possível usar o console do AWS Systems Manager para adicionar, gerenciar ou excluir entradas no Change Calendar, uma ferramenta do AWS Systems Manager. Você também pode importar eventos de provedores de calendário de terceiros compatíveis, importando um arquivo do iCalendar (`.ics`) exportado do calendário de origem. E você pode usar a operação API `GetCalendarState` ou o comando `get-calendar-state` da AWS Command Line Interface (AWS CLI) para obter informações sobre o estado do Change Calendar em um horário específico.

Tópicos

- Criar um calendário de alterações
- Criar e gerenciar eventos no Change Calendar
- Importar e gerenciar eventos de calendários de terceiros
- Atualizar um calendário de alterações
- Compartilhar um calendário de alterações
- Excluir um calendário de alterações
- Obter o estado de um calendário de alterações

Criar um calendário de alterações

Quando você cria uma entrada no Change Calendar, uma ferramenta do AWS Systems Manager, você está criando um documento do Systems Manager (documento SSM) que usa o formato `text`.

Para criar um calendário de alterações

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Change Calendar.
3. Escolha Create calendar (Criar calendário).

- ou -

Se a página inicial Change Calendar abrir primeiro, escolha Create change calendar (Criar calendário de alterações).

4. Na página Create calendar (Criar calendário) em Calendar details (Detalhes do calendário), insira um nome para a entrada do calendário. Nomes de entradas do calendário podem conter letras, números, pontos, traços e sublinhados. O nome deve ser específico o suficiente para identificar rapidamente a finalidade da entrada do calendário. Um exemplo é **`support-off-hours`**. Não é possível atualizar esse nome depois de criar a entrada do calendário.
5. Em Description (Descrição), insira uma descrição para a sua entrada do calendário.
6. (Opcional) Na área Import calendar (Importar calendário), selecione Choose file (Escolher arquivo) para selecionar um arquivo do iCalendar (`.ics`) exportado de um provedor de calendário terceirizado. Importar o arquivo adicionará seus eventos ao seu calendário.

   Os provedores compatíveis incluem o Google Agenda, o Microsoft Outlook e o Calendário do iCloud.

   Para ter mais informações, consulte Importar eventos de provedores de calendário de terceiros.
7. Em Calendar type (Tipo de calendário), escolha uma das seguintes opções.

   - Open by default (Aberto por padrão): o calendário é aberto (as ações de automação podem ser executadas até que um evento seja iniciado) e fechado logo depois pela duração de um evento associado.
   - Closed by default (Fechado por padrão) - o calendário está fechado (as ações do Automation não podem ser executadas até que um evento seja iniciado), mas é aberto durante um evento associado.
8. (Opcional) Em Eventos de gerenciamento de alterações, selecione Adicionar eventos de gerenciamento de alterações ao calendário. Essa seleção exibe todas as janelas de manutenção programada, associações do State Manager, fluxos de trabalho do Automation e solicitações de alterações do Change Manager na exibição do calendário mensal.

> **Tip**
>
> Depois, caso você queira remover permanentemente esses tipos de eventos da exibição do calendário, edite o calendário, desmarque essa caixa de seleção e escolha Salvar.

9. Escolha Create calendar (Criar calendário).

   Depois que a entrada do calendário for criada, o Systems Manager exibirá a entrada do calendário na lista Change Calendar. As colunas exibem a versão do calendário e o número de conta da Conta da AWS do proprietário do calendário. Sua entrada do calendário não pode prevenir ou permitir nenhuma ação até que você adicione pelo menos um evento. Para obter mais informações sobre como criar um evento, consulte Criar um evento do Change Calendar. Para obter mais informações sobre a importação de eventos, consulte Importar eventos de provedores de calendário de terceiros.

Criar e gerenciar eventos no Change Calendar

Depois de criar um calendário no AWS Systems Manager Change Calendar, você poderá criar, atualizar e excluir eventos incluídos no calendário aberto ou fechado. O Change Calendar é uma ferramenta do AWS Systems Manager.

> **Tip**
>
> Como alternativa à criação de eventos diretamente no console do Systems Manager, você pode importar um iCalendar (`.ics`) de uma aplicação de calendário de terceiros compatível. Para ter mais informações, consulte Importar e gerenciar eventos de calendários de terceiros.

Tópicos

- Criar um evento do Change Calendar
- Atualizar um evento do Change Calendar
- Excluir um evento do Change Calendar

Criar um evento do Change Calendar

Ao adicionar um evento a uma entrada do Change Calendar, uma ferramenta do AWS Systems Manager, você está especificando um período durante o qual a ação padrão da entrada do calendário está suspensa. Por exemplo, se o tipo de entrada do calendário for fechado por padrão, o calendário estará aberto para alterações durante eventos. (Alternativamente, você pode criar um evento consultivo, que cumpre uma função informativa apenas no calendário.)

Atualmente, só é possível pode criar um evento do Change Calendar usando o console. Os eventos são adicionados ao documento do Change Calendar criado quando você cria uma entrada do Change Calendar.

Para criar um evento do Change Calendar

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Change Calendar.
3. Na lista de calendários, escolha o nome da entrada do calendário à qual você deseja adicionar um evento.
4. Na página de detalhes da entrada do calendário, escolha Create event (Criar evento).
5. Na página Create scheduled event (Criar evento programado) em Event details (Detalhes do evento), insira um nome de exibição para seu evento. Nomes de eventos podem conter letras, números, pontos, traços e sublinhados. O nome deve ser específico o suficiente para identificar a finalidade do evento. Um exemplo é **`nighttime-hours`**.
6. Em Description (Descrição), insira uma descrição para o seu evento. Por exemplo, **`The support team isn't available during these hours`**.
7. (Opcional) Se você quiser que este evento sirva como uma notificação visual ou lembrete apenas, selecione a opção Advisory (Consultivo). Eventos consultivos não desempenham nenhum papel funcional em seu calendário. Eles servem para fins informativos apenas para aqueles que visualizam seu calendário.
8. Em Event start date (Data de início do evento), insira ou escolha um dia no formato `MM/DD/YYYY` para iniciar o evento e insira uma hora no dia especificada no formato `hh:mm:ss` (horas, minutos e segundos) para iniciar o evento.
9. Em Event end date (Data de término do evento), insira ou escolha um dia no formato `MM/DD/YYYY` para encerrar o evento e insira uma hora no dia especificada no formato `hh:mm:ss` (horas, minutos e segundos) para encerrar o evento.
10. Em Schedule time zone (Fuso horário da programação), escolha um fuso horário que se aplique às horas de início e término do evento. É possível inserir parte de um nome de cidade ou diferença de fuso horário de Greenwich Mean Time (GMT) para encontrar um fuso horário mais rapidamente. O padrão é o Tempo Universal Coordenado (UTC).
11. (Opcional) Para criar um evento que se repita diariamente, semanalmente ou mensalmente, ative Recurrence (Recorrência) e, em seguida, especifique a frequência e a data final opcional para a recorrência.

12. Escolha Create scheduled event (Criar evento programado). O novo evento é adicionado à entrada do calendário e é exibido na guia Events (Eventos) da página de detalhes da entrada do calendário.

Atualizar um evento do Change Calendar

Use o procedimento a seguir para atualizar um evento do Change Calendar no console do AWS Systems Manager. O Change Calendar é uma ferramenta do AWS Systems Manager.

Para atualizar um evento do Change Calendar

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Change Calendar.
3. Na lista de calendários, escolha o nome da entrada do calendário para a qual você deseja editar um evento.
4. Na página de detalhes da entrada do calendário, escolha Events (Eventos).
5. Na página do calendário, escolha o evento que deseja editar.

> **Tip**
>
> Use os botões no canto superior esquerdo para se mover para atrás ou para frente de um ano e de um mês. Altere o fuso horário, se necessário, escolhendo o fuso horário correto na lista no canto superior direito.

6. Em Event details (Detalhes do evento), escolha Edit (Editar).

   Para alterar o nome e a descrição do evento, adicione ou substitua os valores de texto atuais.
7. Para alterar o valor da Event start date (Data de início do evento), escolha a data de início atual e escolha uma nova data no calendário. Para alterar a hora de início, escolha a hora de início atual e escolha uma nova hora na lista.
8. Para alterar o valor da Event end date (Data de término do evento), escolha a data de término atual e escolha uma nova data no calendário. Para alterar a hora de término, escolha a hora de término atual e escolha uma nova hora na lista.
9. Para alterar o valor de Schedule time zone (Fuso horário da programação), escolha um fuso horário que se aplique às horas de início e término do evento. É possível inserir parte de um nome de cidade ou diferença de fuso horário de Greenwich Mean Time (GMT) para encontrar um fuso horário mais rapidamente. O padrão é o Tempo Universal Coordenado (UTC).

10. (Opcional) Se você quiser que este evento sirva como uma notificação visual ou lembrete apenas, selecione a opção Advisory (Consultivo). Eventos consultivos não desempenham nenhum papel funcional em seu calendário. Eles servem para fins informativos apenas para aqueles que visualizam seu calendário.
11. Escolha Salvar. Suas alterações são exibidas na guia Events (Eventos) da página de detalhes da entrada do calendário. Escolha o evento atualizado para exibir suas alterações.

Excluir um evento do Change Calendar

É possível excluir um evento de cada vez no Change Calendar, uma ferramenta do AWS Systems Manager, usando o AWS Management Console.

> **Tip**
>
> Caso selecione Adicionar eventos de gerenciamento de alterações ao calendário ao criar o calendário, você poderá fazer o seguinte:
>
> - Para ocultar um tipo de evento de gerenciamento de alterações na exibição do calendário temporariamente, escolha o X para o tipo na parte superior da pré-visualização mensal.
> - Para remover esses tipos da exibição do calendário permanentemente, edite o calendário, desmarque a caixa de seleção Adicionar eventos de gerenciamento de alterações ao calendário, e escolha Salvar. Remover os tipos da exibição do calendário não os excluirá da conta.

Para excluir um evento do Change Calendar

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Change Calendar.
3. Na lista de calendários, escolha o nome da entrada do calendário da qual você deseja excluir um evento.
4. Na página de detalhes da entrada do calendário, escolha Events (Eventos).
5. Na página do calendário, escolha o evento que você deseja excluir.

> **Tip**
>
> Use os botões no canto superior esquerdo para retroceder ou avançar o calendário em um ano ou um mês. Altere o fuso horário, se necessário, escolhendo o fuso horário correto na lista no canto superior direito.

6. Na página Event details (Detalhes do evento), escolha Delete (Excluir). Quando for solicitado, confirme que você deseja excluir o evento, escolhendo Confirm (Confirmar).

Importar e gerenciar eventos de calendários de terceiros

Como alternativa à criação de eventos diretamente no console do AWS Systems Manager, você pode importar um iCalendar (`.ics`) de uma aplicação de calendário de terceiros compatível. Seu calendário pode incluir eventos importados e eventos que você cria no Change Calendar, uma ferramenta do AWS Systems Manager.

Antes de começar

Antes de tentar importar um arquivo de calendário, revise os seguintes requisitos e restrições:

Formato de arquivo do calendário

Somente arquivos iCalendar válidos (`.ics`) são compatíveis.

Fornecedores de calendário compatíveis

Somente os arquivos `.ics` exportados dos provedores de calendário de terceiros a seguir são compatíveis.

- Calendário do Google (Instruções de exportação)
- Microsoft Outlook (Instruções de exportação)
- Calendário do iCloud (Instruções de exportação)

Tamanho do arquivo

Você pode importar qualquer número de arquivos `.ics` válidos. No entanto, o tamanho total de todos os arquivos importados para cada calendário não pode exceder 64 KB.

> **Tip**
>
> Para minimizar o tamanho do arquivo `.ics`, verifique se você está exportando somente detalhes básicos sobre suas entradas do calendário. Se necessário, reduza a duração do período de exportação.

Fuso horário

Além de um nome de calendário, um provedor de calendário e pelo menos um evento, o arquivo `.ics` exportado também deve indicar o fuso horário do calendário. Se isso não acontecer, ou se houver um problema ao identificar o fuso horário, você será solicitado a especificar um depois de importar o arquivo.

Limitação de eventos recorrentes

O arquivo exportado `.ics` pode incluir eventos recorrentes. No entanto, se uma ou mais ocorrências de um evento recorrente tiverem sido excluídas no calendário de origem, a importação falhará.

Tópicos

- Importar eventos de provedores de calendário de terceiros
- Atualizar todos os eventos de um provedor de calendário de terceiros
- Excluir todos os eventos importados de um calendário de terceiros

Importar eventos de provedores de calendário de terceiros

Use o procedimento a seguir para importar um iCalendar (`.ics`) de uma aplicação de calendário de terceiros compatível. Os eventos contidos no arquivo são incorporados às regras do seu calendário aberto ou fechado. Você pode importar um arquivo para um novo calendário que está criando com o Change Calendar (uma ferramenta do AWS Systems Manager) ou para um calendário existente.

Depois de importar o arquivo `.ics`, você poderá remover eventos individuais dele usando a interface Change Calendar. Para ter mais informações, consulte Excluir um evento do Change Calendar. Você também pode excluir todos os eventos do calendário de origem excluindo o arquivo `.ics`. Para ter mais informações, consulte Excluir todos os eventos importados de um calendário de terceiros.

Para importar eventos de provedores de calendário de terceiros

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Change Calendar.
3. Para começar com um novo calendário, escolha Create calendar (Criar calendário). Na área Import calendar (Importar calendário), selecione Choose file (Escolher arquivo). Para obter informações sobre outras etapas para criar um novo calendário, consulte Criar um calendário de alterações.

   - ou -

   Para importar eventos de terceiros para um calendário existente, escolha o nome de um calendário existente para abri-lo.
4. Selecione Actions, Edit (Ações, Editar), e depois, na área Import calendar (Importar calendário), escolha Choose file (Escolher arquivo).
5. Acesse e selecione o arquivo `.ics` exportado em seu computador local.
6. Caso solicitado, para Select a time zone (Selecione um fuso horário), selecione o fuso horário que se aplica ao calendário.
7. Escolha Salvar.

Atualizar todos os eventos de um provedor de calendário de terceiros

Se vários eventos forem adicionados ou removidos do calendário de origem depois de importar o arquivo `.ics` do iCalendar, você poderá refletir essas alterações no Change Calendar. Primeiro, exporte novamente o calendário de origem e, em seguida, importe o novo arquivo para o Change Calendar, que é uma ferramenta do AWS Systems Manager. Os eventos no calendário de alterações serão atualizados para refletir o conteúdo do arquivo mais recente.

Para atualizar todos os eventos de um provedor de calendário de terceiros

1. No calendário de terceiros, adicione ou remova eventos conforme você deseja que eles sejam refletidos no Change Calendar e, em seguida, reexporte o calendário para um novo arquivo `.ics`.
2. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
3. No painel de navegação, escolha Change Calendar.
4. Na lista de calendários, escolha o nome do calendário na lista.

5. Selecione Escolher arquivo e, em seguida, acesse e selecione o arquivo `.ics` de substituição.
6. Em resposta à notificação sobre a substituição do arquivo existente, escolhaConfirm (Confirme).

Excluir todos os eventos importados de um calendário de terceiros

Se você não quiser mais nenhum dos eventos importados de um provedor de terceiros incluído no calendário, exclua o arquivo `.ics` importado do iCalendar.

Para excluir todos os eventos importados de um calendário de terceiros

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Change Calendar.
3. Na lista de calendários, escolha o nome do calendário na lista.
4. Na área Import calendar (Importar calendário), em My imported calendars (Meus calendários importados), localize o nome do calendário importado e marque X no cartão dele.
5. Escolha Salvar.

Atualizar um calendário de alterações

Você pode atualizar a descrição de um calendário de alterações, mas não o nome dele. Embora seja possível alterar o estado padrão de uma entrada de calendário, esteja ciente de que isso reverte o comportamento das ações de alteração durante eventos associados à entrada do calendário. Por exemplo, se você alterar o estado de um calendário de Open by default (Aberto por padrão) para Closed by default(Fechado por padrão), as alterações indesejadas poderão ser feitas durante períodos do evento em que os usuários que criaram os eventos associados não estiverem esperando alterações.

Ao atualizar um calendário de alterações, você está editando o documento do Change Calendar criado quando você criou a entrada. O Change Calendar é uma ferramenta do AWS Systems Manager.

Para atualizar um calendário de alterações

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Change Calendar.
3. Na lista de calendários, escolha o nome do calendário que você deseja atualizar.
4. Na página de detalhes do calendário, escolha Actions, Edit (Ações, Editar).

5. Em Description (Descrição), é possível alterar o texto da descrição. Não é possível editar o nome de um calendário de alteração.
6. Para alterar o estado do calendário, em Calendar type (Tipo de calendário), escolha um valor diferente. Observe que isso reverte o comportamento das ações de alteração durante eventos associados ao calendário. Antes de alterar o tipo do calendário, verifique com outros usuários do Change Calendar se a alteração do tipo do calendário não permite alterações indesejadas durante os eventos criados.

   - Open by default (Aberto por padrão): o calendário é aberto (as ações do Automation podem ser executadas até que um evento seja iniciado) e fechado logo depois pela duração de um evento associado.
   - Closed by default (Fechado por padrão) - o calendário está fechado (as ações do Automation não podem ser executadas até que um evento seja iniciado), mas é aberto durante um evento associado.
7. Escolha Salvar.

   Sua entrada do calendário não pode prevenir ou permitir ações até que você adicione pelo menos um evento. Para obter informações sobre como adicionar um evento, consulte Criar um evento do Change Calendar.

Compartilhar um calendário de alterações

Você pode compartilhar um calendário no Change Calendar, uma ferramenta do AWS Systems Manager, com outras Contas da AWS usando o console do AWS Systems Manager. Ao compartilhar um calendário, o calendário é somente leitura para usuários na conta compartilhada. Janelas de manutenção, associações do State Manager e automações não são compartilhadas.

Para compartilhar um calendário de alterações

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Change Calendar.
3. Na lista de calendários, escolha o nome do calendário que você deseja compartilhar.
4. Na página de detalhes do calendário, escolha a guia Sharing (Compartilhamento).
5. Escolha Actions, Share (Ações, Compartilhar).
6. Em Share calendar (Compartilhar calendário), para Account ID (ID da conta), insira o número de ID de uma Conta da AWS válida e escolha Share (Compartilhar).

Os usuários da conta compartilhada podem ler o calendário de alterações, mas não podem fazer alterações.

Excluir um calendário de alterações

Você pode excluir um calendário no Change Calendar, uma ferramenta do AWS Systems Manager, usando o console do Systems Manager ou a AWS Command Line Interface (AWS CLI). A exclusão de um calendário de alterações exclui todos os eventos associados.

Para excluir um calendário de alterações

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Change Calendar.
3. Na lista de calendários, escolha o nome do calendário que você deseja excluir.
4. Na página de detalhes do calendário, escolha Actions, Delete (Ações, Excluir). Quando for solicitado, confirme que você deseja excluir a entrada do calendário escolhendo Delete (Excluir).

Obter o estado de um calendário de alterações

Você pode obter o estado geral de um calendário ou o estado de um calendário em um horário específico no Change Calendar, uma ferramenta do AWS Systems Manager. Também é possível mostrar a próxima vez em que o estado do calendário muda de `OPEN` para `CLOSED` ou vice-versa.

> **Note**
>
> Para obter informações sobre a integração do Change Calendar com o Amazon EventBridge para monitoramento automatizado das mudanças de estado do calendário, consulte the section called “Integração do Change Calendar com o Amazon EventBridge”. A integração com o EventBridge fornece notificações orientadas por eventos quando os estados do calendário fazem a transição, complementando a abordagem baseada em polling da ação da API `GetCalendarState`.

Você só pode fazer essa tarefa usando a API `GetCalendarState`. O procedimento nesta seção usa a AWS Command Line Interface (AWS CLI).

Para obter o estado de um calendário de alterações

- Execute o seguinte comando para mostrar o estado de um ou mais calendários em um horário específico. O parâmetro `--calendar-names` é obrigatório, mas `--at-time` é opcional. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

  Linux & macOS

  ```
  aws ssm get-calendar-state \
      --calendar-names "Calendar_name_or_document_ARN_1"
   "Calendar_name_or_document_ARN_2" \
      --at-time "ISO_8601_time_format"
  ```

  Veja um exemplo a seguir.

  ```
  aws ssm get-calendar-state \
      --calendar-names "arn:aws:ssm:us-east-2:123456789012:document/
  MyChangeCalendarDocument" "arn:aws:ssm:us-east-2:123456789012:document/
  SupportOffHours" \
      --at-time "2020-07-30T11:05:14-0700"
  ```

  Windows

  ```
  aws ssm get-calendar-state ^
      --calendar-names "Calendar_name_or_document_ARN_1"
   "Calendar_name_or_document_ARN_2" ^
      --at-time "ISO_8601_time_format"
  ```

  Veja um exemplo a seguir.

  ```
  aws ssm get-calendar-state ^
      --calendar-names "arn:aws:ssm:us-east-2:123456789012:document/
  MyChangeCalendarDocument" "arn:aws:ssm:us-east-2:123456789012:document/
  SupportOffHours" ^
      --at-time "2020-07-30T11:05:14-0700"
  ```

  O comando retorna informações como as seguintes.

```
{
    "State": "OPEN",
    "AtTime": "2020-07-30T16:18:18Z",
    "NextTransitionTime": "2020-07-31T00:00:00Z"
}
```

Os resultados mostram o estado do calendário (se o calendário é do tipo DEFAULT_OPEN ou DEFAULT_CLOSED) para as entradas de calendário especificadas que são propriedade ou compartilhadas com sua conta, na hora especificada como o valor `--at-time` e a hora da próxima transição. Se você não adicionar o parâmetro `--at-time`, a hora atual será usada.

> **Note**
>
> Se você especificar mais de um calendário em uma solicitação, o comando retornará o status OPEN somente se todos os calendários na solicitação estiverem abertos. Se um ou mais calendários na solicitação forem fechados, o status retornado será CLOSED.

Adicionar dependências do Change Calendar para runbooks do Automation

Para garantir que as ações do Automation sigam o Change Calendar, uma ferramenta do AWS Systems Manager, adicione uma etapa em um runbook do Automation que use a ação `aws:assertAwsResourceProperty`. Configure a ação para executar `GetCalendarState` a fim de verificar se uma entrada de calendário especificada está no estado desejado (OPEN ou CLOSED). O runbook do Automation só poderá passar para a próxima etapa se o estado do calendário for OPEN. Veja a seguir um trecho de exemplo baseado em YAML de um runbook do Automation que não pode prosseguir para a próxima etapa, `LaunchInstance`, a menos que o estado do calendário corresponda a OPEN, o estado especificado em `DesiredValues`.

Veja um exemplo a seguir.

```
mainSteps:
  - name: MyCheckCalendarStateStep
    action: 'aws:assertAwsResourceProperty'
    inputs:
      Service: ssm
      Api: GetCalendarState
      CalendarNames: ["arn:aws:ssm:us-east-2:123456789012:document/SaleDays"]
```

```
      PropertySelector: '$.State'
      DesiredValues:
      - OPEN
    description: "Use GetCalendarState to determine whether a calendar is open or
 closed."
    nextStep: LaunchInstance
  - name: LaunchInstance
    action: 'aws:executeScript'
    inputs:
      Runtime: python3.8
...
```

Solução de problemas do Change Calendar

Use as informações a seguir para obter ajuda para solucionar problemas com o Change Calendar, uma ferramenta do AWS Systems Manager.

Tópicos

- Erro 'Falha na importação do calendário'

Erro 'Falha na importação do calendário'

Problema: ao importar um arquivo do iCalendar (`.ics`), o sistema informa que a importação do calendário falhou.

- Solução 1: verifique se você está importando um arquivo que foi exportado de um provedor de calendário de terceiros compatível, que inclua o seguinte:
  - Calendário do Google (Instruções de exportação)
  - Microsoft Outlook (Instruções de exportação)
  - Calendário do iCloud (Instruções de exportação)
- Solução 2: se o calendário de origem incluir eventos recorrentes, verifique se nenhuma ocorrência individual do evento foi cancelada ou excluída. No momento, o Change Calendar não suporta a importação de eventos recorrentes com cancelamentos individuais. Para resolver o problema, remova o evento recorrente do calendário de origem, exporte novamente o calendário, importe-o novamente para o Change Calendar e, em seguida, adicione o evento recorrente usando a interface do Change Calendar. Para ter mais informações, consulte Criar um evento do Change Calendar.

- Solução 3: verifique se o calendário de origem contém pelo menos um evento. Os carregamentos de arquivos `.ics` que não contêm eventos não têm êxito.
- Solução 4: se o sistema relatar que a importação falhou porque a propriedade `.ics` é muito grande, verifique se você está exportando somente os detalhes básicos sobre suas entradas de calendário. Se necessário, reduza a duração do período de exportação.
- Solução 5: se o Change Calendar não conseguir determinar o fuso horário do calendário exportado quando você tenta importá-lo da guia Events (Eventos), você poderá receber esta mensagem: “Falha na importação do calendário. O Change Calendar não localizou um fuso horário válido’’. Você pode importar o calendário do menu Edit (Editar). Nesse caso, escolha Actions, Edit (Ações, Editar) e, em seguida, tente importar o arquivo da página Edit calendar (Editar calendário).
- Solução 6: não edite o arquivo `.ics` antes da importação. A tentativa de modificar o conteúdo do arquivo pode corromper os dados do calendário. Se você modificou o arquivo antes de tentar a importação, exporte o calendário do calendário de origem novamente e tente fazer o upload novamente.

AWS Systems Manager Change Manager

O Change Manager, uma ferramenta do AWS Systems Manager, é um framework de gerenciamento de alterações corporativas para solicitar, aprovar, implementar e emitir relatórios sobre alterações operacionais em sua configuração e infraestrutura de aplicações. Em uma única Conta de administrador delegado, se você usar o AWS Organizations, poderá gerenciar alterações em várias Contas da AWS e Regiões da AWS. Como alternativa, usando um conta local, você pode gerenciar alterações para uma única Conta da AWS. Use o Change Manager para gerenciar alterações em recursos da AWS e recursos on-premises. Para começar a usar o Change Manager, abra o Systems Manager console (Console do gerenciador de sistemas). No painel de navegação, escolha Change Manager.

Com o Change Manager, você pode usar modelos de alterações pré-aprovado para ajudar a automatizar os processos de alteração para seus recursos e ajudar a evitar resultados não intencionais ao fazer alterações operacionais. Cada modelo de alteração especifica o seguinte:

- Um ou mais runbooks do Automation para um usuário escolher ao criar uma solicitação de alteração. As alterações feitas nos recursos são definidas nos runbooks do Automation. Você pode incluir runbooks personalizados ou runbooks gerenciados da AWS nos modelos de alteração criados. Quando um usuário cria uma solicitação de alteração, ele pode escolher qual dos runbooks disponíveis incluir na solicitação. Além disso, você pode criar modelos de alteração

que permitem que o usuário que faz a solicitação especifique qualquer runbook na solicitação de alteração.

- Os usuários da conta que devem revisar solicitações de alteração feitas usando esse modelo de alteração.
- O tópico do Amazon Simple Notification Service (Amazon SNS) usado para notificar os aprovadores atribuídos de que uma solicitação de alteração está pronta para análise.
- O alarme do Amazon CloudWatch usado para monitorar o fluxo de trabalho do runbook.
- O tópico do Amazon SNS usado para enviar notificações sobre alterações de status para solicitações de alteração criadas usando o modelo de alteração.
- As tags a serem aplicadas ao modelo de alteração para uso na categorização e filtragem dos modelos de alteração.
- Se as solicitações de alteração criadas com o modelo de alteração podem ser executadas sem uma etapa de aprovação (solicitações aprovadas automaticamente).

Através da sua integração com o Change Calendar, que é outra ferramenta do Systems Manager, o Change Manager também ajudará você a implementar alterações com segurança, evitando conflitos de agendamento com eventos comerciais importantes. A integração do Change Manager com o AWS Organizations e AWS IAM Identity Center ajuda você a gerenciar alterações em toda a organização em uma única conta usando seu sistema de gerenciamento de identidades existente. Você pode monitorar o progresso das alterações no Change Manager e auditar alterações operacionais em toda a organização, proporcionando visibilidade e responsabilidade aprimoradas.

O Change Manager complementa os controles de segurança das práticas de integração contínua (CI) e a metodologia de fornecimento contínuo (CD). O Change Manager não se destina a alterações feitas como parte de um processo de lançamento automatizado, como um pipeline CI/CD, a menos que haja uma exceção ou aprovação necessária.

Como a Change Manager funciona

Quando a necessidade de uma alteração operacional padrão ou de emergência for identificada, alguém na organização criará uma solicitação de alteração baseada em um dos modelos de alteração criados para uso em sua organização ou conta.

Se a alteração solicitada exigir aprovações manuais, o Change Manager notificará os aprovadores designados por meio de uma notificação do Amazon SNS de que uma solicitação de alteração está pronta para sua revisão. É possível designar aprovadores para solicitações de alteração no modelo de alteração ou permitir que os usuários designem aprovadores em sua própria solicitação

de alteração. Você pode atribuir diferentes revisores a diferentes modelos. Por exemplo, atribua um usuário, grupo de usuários ou função do AWS Identity and Access Management (IAM) que devem aprovar as solicitações de alterações em nós gerenciados e outro usuário, grupo ou função do IAM para alterações no banco de dados. Se o modelo de alteração permitir aprovações automáticas e a política de usuário de um solicitante não as proibir, o usuário também poderá optar por executar o runbook do Automation para sua solicitação sem uma etapa de análise (com exceção dos eventos de congelamento de alterações).

Para cada modelo de alteração, você pode adicionar até cinco níveis de aprovadores. Por exemplo, você pode exigir que os revisores técnicos aprovem primeiro uma solicitação de alteração criada a partir de um modelo de alteração e, em seguida, solicitar um segundo nível de aprovações de um ou mais gerentes.

O Change Manager é integrado ao AWS Systems Manager Change Calendar. Quando uma alteração solicitada é aprovada, o sistema primeiro determina se a solicitação entra em conflito com outras atividades comerciais agendadas. Se um conflito for detectado, o Change Manager poderá bloquear a alteração ou exigir aprovações adicionais antes de iniciar o fluxo de trabalho do runbook. Por exemplo, você pode permitir alterações somente durante o horário comercial para garantir que as equipes estejam disponíveis para gerenciar problemas inesperados. Para quaisquer alterações solicitadas para serem executadas fora desse horário, você pode solicitar a aprovação da gerência de nível superior no formato de change freeze approvers (alterar aprovadores de congelamento). Para mudanças de emergência, o Change Manager pode ignorar a etapa de verificação Change Calendar para conflitos ou eventos de bloqueio após a aprovação de uma solicitação de alteração.

Quando for a hora de implementar uma alteração aprovada, o Change Manager executará o runbook do Automation especificado na solicitação de alteração associada. Somente as operações definidas em solicitações de alteração aprovadas serão permitidas quando os fluxos de trabalho de runbook forem executados. Essa abordagem ajuda você a evitar resultados não intencionais enquanto as mudanças estiverem sendo implementadas.

Além de restringir as alterações que podem ser feitas quando um fluxo de trabalho de runbook for executado, o Change Manager também ajuda a controlar a simultaneidade e os limites de erros. Você escolhe quantos recursos um fluxo de trabalho de runbook pode executar de uma só vez, quantas contas a alteração pode executar de uma só vez e quantas falhas serão permitidas antes do processo ser interrompido e (se o runbook incluir um script de reversão) revertido. Você também pode monitorar o andamento das alterações feitas usando os alarmes do CloudWatch.

Após a conclusão de um fluxo de trabalho do runbook, você pode revisar os detalhes sobre as alterações feitas. Esses detalhes incluem o motivo de uma solicitação de alteração, qual modelo

de alteração foi usado, quem solicitou e aprovou as alterações e como as alterações foram implementadas.

Mais informações

Apresentação do AWS Systems ManagerChange Manager no Blog de notícias da AWS

Como o Change Manager beneficia minhas operações?

Os benefícios do Change Manager incluem o seguinte:

- Reduza o risco de interrupção do serviço e tempo de inatividade

  O Change Manager pode tornar as alterações operacionais mais seguras, garantindo que apenas as alterações aprovadas sejam implementadas quando um fluxo de trabalho de runbook for executado. Você pode bloquear alterações não planejadas e não revisadas. O Change Manager ajuda você a evitar os tipos de resultados não intencionais causados por erro humano que exigem horas dispendiosas de pesquisa e retrocesso.
- Obtenha auditoria e relatórios detalhados sobre os históricos de alterações

  O Change Manager fornece responsabilidade com uma maneira consistente de relatar e auditar alterações feitas em toda a organização, a intenção das alterações e detalhes sobre quem as aprovou e implementou.
- Evite conflitos ou violações de agendamento

  O Change Manager pode detectar conflitos de agendamento, como eventos de feriados ou lançamentos de novos produtos, com base no calendário de alterações ativo da sua organização. Você pode permitir que fluxos de trabalho do runbook sejam executados somente durante o horário comercial ou permiti-los somente com aprovações adicionais.
- Adapte os requisitos de mudança aos seus negócios em mudança

  Durante diferentes períodos comerciais, você poderá implementar diferentes requisitos de gerenciamento de alterações. Por exemplo, durante a geração de relatórios de fim de mês, época fiscal ou outros períodos comerciais críticos, você pode bloquear alterações ou exigir aprovação em nível de diretoria para alterações que possam introduzir riscos operacionais desnecessários.
- Gerencie centralmente as alterações nas contas

Através da sua integração com o Organizations, o Change Manager permite que você gerencie alterações em todas as unidades organizacionais (UOs) em uma única conta de administrador delegado. É possível ativar o Change Manager para uso com toda a sua organização ou com apenas algumas das suas UOs.

Quem deve usar o Change Manager?

O Change Manager é apropriado para o seguinte clientes e organizações da AWS:

- Qualquer cliente da AWS que quiser melhorar a segurança e a governança das alterações operacionais feitas em seus ambientes na nuvem ou on-premises.
- As organizações que desejam aumentar a colaboração e a visibilidade entre as equipes, melhorar a disponibilidade das aplicações evitando o tempo de inatividade e reduzir o risco associado a tarefas manuais e repetitivas.
- Organizações que devem estar em conformidade com as práticas recomendadas para o gerenciamento de alterações.
- Clientes que precisam de um histórico totalmente auditável das alterações feitas em sua configuração e infraestrutura de aplicações.

Quais são os principais recursos do Change Manager?

Os principais recursos do Change Manager incluem o seguinte:

- Suporte integrado para as práticas recomendadas de gerenciamento de alterações

  Com o Change Manager, você pode aplicar práticas recomendadas de gerenciamento de alterações selecionadas às suas operações. Você pode ativar uma das seguintes opções:

  - Confira o Change Calendar para ver se os eventos estão restritos de forma que as alterações sejam feitas somente durante períodos de calendário abertos.
  - Permitir alterações durante eventos restritos com aprovações extras de aprovadores de congelamento de alterações.
  - Exija que os alarmes do CloudWatch sejam especificados para todos os modelos de alteração.
  - Exija que todos os modelos de alteração criados em sua conta sejam revisados e aprovados antes que possam ser usados para criar solicitações de alteração.

- Caminhos diferentes de aprovação para períodos de calendário fechados e solicitações de alteração de emergência

  Você pode permitir que uma opção escolha o Change Calendar para eventos restritos e bloqueie solicitações de alteração aprovadas, até que o evento seja concluído. No entanto, você também pode designar um segundo grupo de aprovadores, alterar aprovadores congelados, que podem permitir que a alteração seja feita mesmo que o calendário esteja fechado. Você também pode criar modelos de alteração de emergência. As solicitações de alteração criadas a partir de um modelo de alteração de emergência ainda exigem aprovações regulares, mas não estão sujeitas às restrições do calendário e não exigem aprovações de congelamento de alterações.

- Controle como e quando os fluxos de trabalho do runbook são iniciados

  Os fluxos de trabalho do runbook podem ser iniciados de acordo com uma programação ou assim que as aprovações forem concluídas (sujeito a regras de restrição de calendário).

- Suporte de notificação integrado

  Especifique quem na sua organização deve revisar e aprovar modelos e solicitações de alteração. Atribua um tópico do Amazon SNS a um modelo de alteração para enviar notificações aos assinantes do tópico sobre alterações de status para solicitações de alteração criadas com esse modelo de alteração.

- Integração com AWS Systems Manager Change Calendar

  O Change Manager permite que os administradores restrinjam alterações de agendamento durante períodos de tempo especificados. Por exemplo, você pode criar uma política que permita alterações somente durante o horário comercial para garantir que a equipe esteja disponível para lidar com quaisquer problemas. Você também pode restringir alterações durante eventos comerciais importantes. Por exemplo, as empresas de varejo podem restringir as alterações durante grandes eventos de vendas. Você também pode exigir aprovações adicionais durante períodos restritos.

- Integração com o AWS IAM Identity Center e suporte ao Active Directory

  Com a integração ao IAM Identity Center, os membros da sua organização podem acessar Contas da AWS e gerenciar seus recursos usando o Systems Manager, com base em uma identidade de um usuário comum. Usando o IAM Identity Center, você pode atribuir acesso de usuários a contas na AWS.

A integração com o Active Directory torna possível atribuir usuários em sua conta do Active Directory como aprovadores para modelos de alteração criados para as operações do Change Manager.

- Integração com alarmes do Amazon CloudWatch

  O Change Manager é integrado aos alarmes do CloudWatch. O Change Manager escuta os alarmes do CloudWatch durante o fluxo de trabalho do runbook e executa todas as ações, incluindo o envio de notificações definidas para o alarme.

- Integração com o AWS CloudTrail Lake

  Ao criar um armazenamento de dados de eventos no AWS CloudTrail Lake, é possível visualizar informações auditáveis sobre as alterações feitas pelas solicitações de alteração executadas em sua conta ou organização. As informações do evento armazenadas incluem detalhes como os seguintes:

  - As ações de API que foram executadas
  - Os parâmetros de solicitação incluídos para essas ações
  - O usuário que executou a ação
  - Os recursos que foram atualizados durante o processo

- Integração com AWS Organizations

  Usando os recursos de contas cruzadas fornecidos pelo Organizations, você pode usar uma conta de administrador delegado para gerenciar as operações do Change Manager em UOs da sua organização. Na conta de gerenciamento do Organizations, você pode especificar qual conta será a conta de administrador delegado. Você também pode controlar em quais das UOs o Change Manager pode ser usado.

Há cobrança pelo uso do Change Manager?

Sim. O preço do Change Manager é calculado com base no pagamento conforme o uso. Você paga somente pelo que usar. Para obter mais informações, consulte Preços do AWS Systems Manager.

Quais são os componentes primários do Change Manager?

Os componentes do Change Manager que você usa para gerenciar o processo de alteração em sua organização ou conta incluem o seguinte:

Conta de administrador delegado

Se você usar o Change Manager em uma organização, você usará uma conta de administrador delegado. Esta é a Conta da AWS designada como a conta para gerenciar atividades de operações no Systems Manager, incluindo o Change Manager. A conta de administrador delegado gerencia as atividades de alteração em toda a organização. Quando você configura sua organização para usar o Change Manager, você especifica qual das suas contas desempenhará essa função. A conta de administrador delegado deve ser o único membro da unidade organizacional (UO) à qual está atribuída. A conta de administrador delegado não será necessária se você usar o Change Manager apenas com uma única Conta da AWS.

> **Important**
>
> Se você usar o Change Manager em uma organização, recomendamos sempre fazer as alterações na conta de administrador delegado. Embora seja possível fazer alterações de outras contas na organização, essas alterações não serão relatadas ou visíveis na conta do administrador delegado.

Modelo de alteração

Um modelo de alteração é uma coleção de definições de configuração no Change Managerque definem itens como aprovações obrigatórias, runbooks disponíveis e opções de notificação para solicitações de alteração.

Você pode exigir que os modelos de alteração criados pelos usuários em sua organização ou conta passem por um processo de aprovação antes que possam ser usados.

O Change Manager oferece suporte a dois tipos de modelos de alteração. Para uma solicitação de alteração aprovada que se baseia em um modelo de alteração de emergência, a alteração solicitada pode ser feita mesmo que existam eventos de bloqueio no Change Calendar. Para uma solicitação de alteração aprovada que se baseia em um modelo de alteração padrão, a alteração solicitada não poderá ser feita se houver eventos de bloqueio no Change Calendar a menos que aprovações adicionais sejam recebidas dos aprovadores designados em change freeze event (alterar evento de congelamento).

Solicitação de alteração

Uma solicitação de alteração é uma solicitação no Change Manager para executar um runbook do Automation que atualiza um ou mais recursos na AWS ou em ambientes on-premises. Uma solicitação de alteração é criada usando um modelo de alteração.

Quando você cria uma solicitação de alteração, um ou mais aprovadores em sua organização ou conta devem revisar e aprovar a solicitação. Sem as aprovações necessárias, o fluxo de trabalho do runbook, que aplica as alterações solicitadas, não tem permissão para ser executado.

No sistema, as solicitações de alteração são um tipo de OpsItem no AWS Systems Manager OpsCenter. No entanto, OpsItems do tipo `/aws/changerequest` não são exibidos no OpsCenter. Como OpsItems, as solicitações de alteração estão sujeitas às mesmas cotas impostas que os outros tipos de OpsItems.

Além disso, para criar uma solicitação de alteração programaticamente, não chame a operação da API `CreateOpsItem`, Use a operação `StartChangeRequestExecution` da API. Mas, em vez de ser executada imediatamente, a solicitação de alteração deve ser aprovada, e não deve haver nenhum evento de bloqueio no Change Calendar para impedir que o fluxo de trabalho seja executado. Quando as aprovações tiverem sido recebidas e o calendário não estiver bloqueado (ou tiver permissão para ignorar eventos de bloqueio do calendário), a ação `StartChangeRequestExecution` poderá ser concluída.

Fluxo de trabalho do runbook

Um fluxo de trabalho de runbook é o processo de alterações solicitadas que estão sendo feitas nos recursos direcionados no ambiente de nuvem ou on-premises. Cada solicitação de alteração designa um único runbook do Automation a ser usado para fazer a alteração solicitada. O fluxo de trabalho do runbook ocorre depois que todas as aprovações necessárias foram concedidas e não houver eventos de bloqueio no Change Calendar. Se a alteração tiver sido agendada para uma data e hora específicas, o fluxo de trabalho do runbook não começará até o horário agendado, mesmo que todas as aprovações tenham sido recebidas e o calendário não esteja bloqueado.

Tópicos

- Configurar o Change Manager
- Trabalhar com o Change Manager
- Auditar e registrar atividades do Change Manager em log
- Solução de problemas do Change Manager

Configurar o Change Manager

Você pode usar o Change Manager, uma ferramenta do AWS Systems Manager, para gerenciar alterações de uma organização inteira, conforme configurado no AWS Organizations, ou para uma única Conta da AWS.

Se você estiver usando o Change Manager com uma organização, comece com o tópico Configure o Change Manager para uma organização (conta de gerenciamento) e, em seguida, vá para Configurar as opções e práticas recomendadas do Change Manager.

Se você estiver usando Change Manager com uma única conta, vá diretamente para Configurar as opções e práticas recomendadas do Change Manager.

> **Note**
>
> Se começar você a usar o Change Manager com uma única conta, mas essa conta é posteriormente adicionada a uma unidade organizacional para a qual o Change Manager for permitido, as configurações de conta única serão desconsideradas.

Tópicos

- Configure o Change Manager para uma organização (conta de gerenciamento)
- Configurar as opções e práticas recomendadas do Change Manager
- Configurar perfis e permissões para o Change Manager
- Controlar o acesso a fluxos de trabalho do runbook de aprovação automática

Configure o Change Manager para uma organização (conta de gerenciamento)

As tarefas neste tópico se aplicam se você estiver usando o Change Manager, uma ferramenta do AWS Systems Manager, com uma organização configurada no AWS Organizations. Se você quiser usar o Change Manager apenas com uma única Conta da AWS, vá para o tópico Configurar as opções e práticas recomendadas do Change Manager.

Execute as tarefas nesta seção em uma Conta da AWS que estiver atuando como a Conta de gerenciamento no Organizations. Para obter informações sobre a conta de gerenciamento e outros conceitos do Organizations, consulte Terminologia e conceitos do AWS Organizations.

Se você precisar ativar o Organizations e especificar sua conta como conta de gerenciamento antes de prosseguir, consulte Criar e gerenciar uma organização no Manual do usuário do AWS Organizations.

> **Note**
>
> Esse processo de configuração não pode ser executado na seguinte Regiões da AWS:
>
> - UE (Milão) (eu-south-1)
> - Oriente Médio (Bahrein) (me-south-1)
> - África (Cidade do Cabo) (af-south-1)
> - Ásia-Pacífico (Hong Kong) (ap-east-1)
>
> Verifique se você está trabalhando em uma região diferente em sua conta de gerenciamento para este procedimento.

Durante o procedimento de instalação, você realizará as seguintes tarefas principais no Quick Setup, uma ferramenta do AWS Systems Manager.

- Tarefa 1: Registrar a conta do administrador delegado do para a sua organização

  As tarefas relacionadas à alteração que são executadas usando o Change Manager são gerenciadas em uma de suas contas de membro, que você especifica como sendo a conta do administrador delegado. A conta de administrador delegado na qual você registra o Change Manager torna-se a conta de administrador delegado para todas as operações do Systems Manager. (Você pode ter contas de administrador delegado para outros Serviços da AWS). A conta de administrador delegado do Change Manager, que não é o mesmo que sua conta de gerenciamento, gerencia atividades de alteração em toda a organização, incluindo modelos de alteração, solicitações de alteração e aprovações para cada uma delas. Na conta de administrador delegado, você também especifica outras opções de configuração para o as operações do Change Manager.

  > **Important**
  >
  > A conta de administrador delegado deve ser o único membro da unidade organizacional (UO) ao qual está atribuída no Organizations.

- Tarefa 2: Definir e especificar as políticas de acesso do runbook para funções do solicitante de alterações ou funções de trabalho personalizadas, que você quiser usar para as operações do Change Manager

  Para criar solicitações de alteração no Change Manager, os usuários em suas contas de membros devem ter permissões do AWS Identity and Access Management (IAM) para acessar somente os runbooks do Automation e alterar os modelos que você disponibilizar para eles.

  > **Note**
  >
  > Quando um usuário cria uma solicitação de alteração, ele primeiro seleciona um modelo de alteração. Esse modelo de alteração pode disponibilizar vários runbooks, mas o usuário pode selecionar apenas um runbook para cada solicitação de alteração. Os modelos de alteração também podem ser configurados para permitir que os usuários incluam qualquer runbook disponível em suas solicitações.

  Para conceder as permissões necessárias, o Change Manager usa o conceito de funções de trabalho, que também é usado pelo IAM. No entanto, ao contrário das Políticas gerenciadas pela AWS para as funções de trabalho no IAM, você especifica os nomes das suas funções de trabalho do Change Manager e as permissões do IAM para elas.

  Quando você configura uma função de trabalho, recomendamos criar uma política personalizada e fornecer apenas as permissões necessárias para executar tarefas de gerenciamento de alterações. Por exemplo, é possível especificar permissões que limitem os usuários a esse conjunto específico de runbooks com base nas funções de trabalho que você definir.

  Por exemplo, você pode criar uma função de trabalho com o nome `DBAdmin`. Para essa função de trabalho, você pode conceder apenas permissões necessárias para runbooks relacionados aos bancos de dados do Amazon DynamoDB, como `AWS-CreateDynamoDbBackup` e `AWSConfigRemediation-DeleteDynamoDbTable`.

  Como outro exemplo, você pode conceder a alguns usuários apenas as permissões necessárias para trabalhar com runbooks relacionados aos buckets do Amazon Simple Storage Service (Amazon S3), como `AWS-ConfigureS3BucketLogging` e `AWSConfigRemediation-ConfigureS3BucketPublicAccessBlock`.

O processo de configuração no Quick Setup para o Change Manager também disponibiliza um conjunto de permissões administrativas completas do Systems Manager para aplicar a uma função administrativa criada.

Cada configuração do Change Manager Quick Setup que você implanta cria uma função de trabalho em sua conta de administrador delegado com permissões para executar modelos do Change Manager e runbooks do Automation nas unidades organizacionais que você selecionou. Você pode criar até 15 configurações do Quick Setup para o Change Manager.

- Tarefa 3: Escolher quais contas de membro em sua organização usar com o Change Manager

  Você pode usar o Change Manager com todas as contas de membros em todas as unidades organizacionais que estiverem configuradas no Organizations e em todas as Regiões da AWS em que eles operam. Se você preferir, use o Change Manager com apenas algumas de suas unidades organizacionais.

> **Important**
>
> Recomendamos enfaticamente que, antes de iniciar este procedimento, você leia as etapas para entender as opções de configuração que você está escolhendo e as permissões que está concedendo. Planeje principalmente as funções de trabalho personalizadas que você criará e as permissões atribuídas a cada função de trabalho. Isso garante que, posteriormente, você anexar as políticas de função de trabalho criadas a usuários individuais, grupos de usuários ou funções do IAM, eles estejam recebendo somente as permissões que você quer que eles tenham.
> Como prática recomendada, comece configurando a conta de administrador delegado usando o login de um administrador da Conta da AWS. Em seguida, configure as funções de trabalho e suas permissões depois de criar os modelos de alteração e identificar os runbooks que cada um usa.

Para configurar o Change Manager para uso com uma organização, execute a seguinte tarefa na área Quick Setup do console do Systems Manager.

Repita essa tarefa para cada função de trabalho que você deseja criar para sua organização. Cada função de trabalho criada pode ter permissões para um conjunto diferente de unidades organizacionais.

Para configurar uma organização no Change Manager na conta de gerenciamento de uma organização

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Quick Setup.
3. No cartão do Change Manager, escolha Create (Criar).
4. Para a Delegated administrator account (Conta de administrador delegado), digite o ID da Conta da AWS que você quer usar para gerenciar modelos de alteração, solicitações de alteração e fluxos de trabalho de runbook no Change Manager.

   Se você tiver especificado anteriormente uma conta de administrador delegado para o Systems Manager, seu ID já será relatado neste campo.

   > **Important**
   >
   > A conta de administrador delegado deve ser o único membro da unidade organizacional (UO) ao qual está atribuída no Organizations.
   > Se a conta de administrador delegado que você registrar for posteriormente cancelada dessa função, o sistema removerá suas permissões para gerenciar operações do Systems Manager ao mesmo tempo. Tenha em mente que será necessário que você retorne ao Quick Setup, designe uma conta de administrador delegado diferente e especifique todas as funções e permissões de trabalho novamente.
   > Se você usar o Change Manager em uma organização, recomendamos sempre fazer as alterações na conta de administrador delegado. Embora seja possível fazer alterações de outras contas na organização, essas alterações não serão relatadas ou visíveis na conta do administrador delegado.

5. Na seção Permissions to request and make changes (Permissões para solicitar e fazer alterações), faça o seguinte:

   > **Note**
   >
   > Cada configuração de implantação criada fornece a política de permissões para apenas uma função de trabalho. Você pode retornar ao Quick Setup mais tarde para criar mais funções de trabalho quando você tiver criado modelos de alteração para usar em suas operações.

Para criar uma função administrativa: para uma função de trabalho de administrador que tenha permissões do IAM para todos as ações da AWS, faça o seguinte:

> **Important**
>
> A concessão de permissões administrativas completas aos usuários deve ser feita com moderação e somente se suas funções exigirem acesso total ao Systems Manager. Para obter informações importantes sobre considerações de segurança para acesso ao Systems Manager, consulte Gerenciamento de identidade e acesso para o AWS Systems Manager e Melhores práticas de segurança do Systems Manager.

1. Para Job function (Função de trabalho), insira um nome para identificar essa função e suas permissões, como **`MyAWSAdmin`**.
2. Para Role and permissions option (Opção Função e permissões), escolha Administrator permissions (Permissões de administrador).

Para criar outras funções de trabalho: para criar uma função não administrativa, faça o seguinte:

1. Para Job function (Função de trabalho), insira um nome para identificar essa função e sugira as permissões. O nome escolhido deverá representar o escopo dos runbooks para os quais você fornecerá permissões, como `DBAdmin` ou `S3Admin`.
2. Para Role and permissions option (Opção Função e permissões), escolha Custom permissions (Permissões personalizadas).
3. Em Permissions policy editor (Editor de políticas de permissões), insira as permissões do IAM, no formato JSON, para conceder a essa função de trabalho.

> **Tip**
>
> Recomendamos que você use o editor de políticas do IAM para construir sua política e, em seguida, cole a política JSON no campo Permissions policy (Política de permissões).

Exemplo de política: gerenciamento de banco de dados do DynamoDB

Por exemplo, você pode começar com o conteúdo da política que fornece permissões para trabalhar com os documentos do Systems Manager (documentos SSM), os quais a função de trabalho precisa acessar. Aqui está um exemplo de conteúdo de política que concede acesso a todos os runbooks do Automation gerenciados pela AWS relacionados aos bancos de dados do DynamoDB e dois modelos de alteração que foram criados no exemplo Conta da AWS `123456789012`, na região Leste dos EUA (Ohio) (`us-east-2`).

A política também inclui permissão para a operação StartChangeRequestExecution, que é necessária para criar uma solicitação de alteração no Change Calendar.

> **Note**
>
> Este exemplo não é abrangente. Permissões adicionais podem ser necessárias para trabalhar com outros recursos da AWS, como bancos de dados e nós.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:CreateDocument",
                "ssm:DescribeDocument",
                "ssm:DescribeDocumentParameters",
                "ssm:DescribeDocumentPermission",
                "ssm:GetDocument",
                "ssm:ListDocumentVersions",
                "ssm:ModifyDocumentPermission",
                "ssm:UpdateDocument",
                "ssm:UpdateDocumentDefaultVersion"
            ],
            "Resource": [
                "arn:aws:ssm:region:*:document/AWS-CreateDynamoDbBackup",
                "arn:aws:ssm:region:*:document/AWS-AWS-DeleteDynamoDbBackup",
                "arn:aws:ssm:region:*:document/AWS-
DeleteDynamoDbTableBackups",
```

```
                "arn:aws:ssm:region:*:document/AWSConfigRemediation-
DeleteDynamoDbTable",
                "arn:aws:ssm:region:*:document/AWSConfigRemediation-
EnableEncryptionOnDynamoDbTable",
                "arn:aws:ssm:region:*:document/AWSConfigRemediation-
EnablePITRForDynamoDbTable",
                "arn:aws:ssm:region:111122223333:document/
MyFirstDBChangeTemplate",
                "arn:aws:ssm:region:111122223333:document/
MySecondDBChangeTemplate"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "ssm:ListDocuments",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ssm:StartChangeRequestExecution",
            "Resource": "arn:aws:ssm:region:111122223333:automation-
definition/*:*"
        }
    ]
}
```

Para obter mais informações sobre políticas do IAM, consulte Gerenciamento de acesso para recursos do AWS e Criar políticas do IAM no Manual do usuário do IAM.

6. Na seção Targets (Destinos), escolha se deseja conceder permissões para a função de trabalho que você está criando para toda a organização ou apenas algumas de suas unidades organizacionais.

   Se você escolher Entire organization (Toda a organização), continue na etapa 9.

   Se escolher Custom (Personalizado), continue na etapa 8.

7. Na seção Target OUs (UOs de destino), marque as caixas de seleção das unidades organizacionais a serem usadas com o Change Manager.

8. Escolha Criar.

Depois que o sistema terminar a configuração do Change Manager para sua organização, ele exibirá um resumo das implantações. Essas informações de resumo incluem o nome do perfil criado para a função de trabalho que você configurou. Por exemplo, `.AWS-QuickSetup-SSMChangeMgr-DBAdminInvocationRole`

> **Note**
>
> O Quick Setup usa StackSets do AWS CloudFormation para implantar suas configurações. Você também pode visualizar informações sobre uma configuração de implantação concluída no console do AWS CloudFormation. Para obter mais informações sobre o StackSets, consulte Trabalhar com o StackSets do AWS CloudFormation no Manual do usuário do AWS CloudFormation.

O próximo passo é configurar outras opções do Change Manager. Você pode concluir essa tarefa em sua conta de administrador delegado ou em qualquer conta em uma unidade organizacional que você tenha permitido para uso com o Change Manager. Você configura opções como escolher uma opção de gerenciamento de identidade de usuário, especificar quais usuários podem revisar e aprovar ou rejeitar modelos de alteração e solicitações de alteração, além de escolher quais opções de práticas recomendadas devem ser permitidas para sua organização. Para mais informações, consulte Configurar as opções e práticas recomendadas do Change Manager.

Configurar as opções e práticas recomendadas do Change Manager

As tarefas nesta seção deverão ser executadas se você estiver usando o Change Manager, uma ferramenta do AWS Systems Manager, em uma organização ou em uma Conta da AWS individual.

Se você estiver usando o Change Manager para uma organização, você poderá executar as tarefas a seguir em sua conta de administrador delegado ou em qualquer conta em uma unidade organizacional que você tenha permissão para uso com o Change Manager.

Tópicos

- Tarefa 1: Configurar o gerenciamento de identidade de usuário e revisores de modelo do Change Manager
- Tarefa 2: Configurar os aprovadores de eventos de congelamento de alterações e práticas recomendadas do Change Manager
- Configurar os tópicos do Amazon SNS para as notificações do Change Manager

Tarefa 1: Configurar o gerenciamento de identidade de usuário e revisores de modelo do Change Manager

Execute a tarefa neste procedimento na primeira vez que você acessar Change Manager. Você pode atualizar essas configurações mais tarde, retornando ao Change Manager e escolhendo Edit (Editar) na guia Settings (Configurações).

Para configurar o gerenciamento de identidade de usuário e revisores de modelo do Change Manager

1. Faça login no AWS Management Console.

   Se você estiver usando o Change Manager para uma organização, entre usando suas credenciais da sua conta de administrador delegado. O usuário deve ter as permissões necessárias do AWS Identity and Access Management (IAM) para realizar atualizações nas configurações do Change Manager.
2. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
3. No painel de navegação, escolha Change Manager.
4. Na página inicial do serviço, dependendo das opções disponíveis, siga um destes procedimentos:

   - Se você estiver usando o Change Manager com o AWS Organizations, escolha Set up delegated account (Configurar conta delegada).
   - Se você estiver usando o Change Manager com uma única Conta da AWS, escolha Setup Change Manager (Configurar o Change Manager).

     - ou -

     Selecione Create sample change request (Criar solicitação de alteração de exemplo), Skip (Ignorar) e, em seguida, escolha a guia Settings (Configurações).
5. Para User identity management (Gerenciamento de identidade do usuário), escolha uma das seguintes opções:

   - AWS Identity and Access Management (IAM): identifique os usuários que realizam e aprovam solicitações e executam outras ações no Change Manager ao usar seus usuários, grupos e perfis existentes.

- AWS IAM Identity Center (IAM Identity Center): permite que o IAM Identity Center crie e gerencie identidades ou se conecte à sua fonte de identidade existente para identificar os usuários que executam ações no Change Manager.

6. Na seção Template reviewer notification (Notificação do revisor de modelo), especifique os tópicos do Amazon Simple Notification Service (Amazon SNS) a serem usados para notificar os revisores de modelos de que há um novo modelo de alteração ou versão de modelo de alteração pronto para análise. Verifique se o tópico do Amazon SNS escolhido está configurado para enviar notificações aos revisores do modelo.

   Para obter informações sobre como criar e configurar tópicos do Amazon SNS para notificações de revisão de modelo de alteração, consulte Configurar os tópicos do Amazon SNS para as notificações do Change Manager.

   1. Para especificar o tópico do Amazon SNS para notificação do revisor de modelo, escolha uma das opções a seguir:

      - Insira um nome do recurso da Amazon (ARN) – Para ARN do tópico, insira o ARN de um tópico existente do Amazon SNS. Esse tópico pode estar em qualquer uma das contas da sua organização.
      - Selecione um tópico do SNS existente: para Target notification topic (Tópico de notificação de destino), selecione o ARN de um tópico existente do Amazon SNS em sua Conta da AWS atual. (Essa opção não estará disponível se você ainda não tiver criado nenhum tópico do Amazon SNS na sua Conta da AWS e Região da AWS.)

      > **Note**
      >
      > O tópico do Amazon SNS selecionado deve ser configurado para especificar as notificações que ele envia e os assinantes para os quais eles são enviados. Sua política de acesso também deve conceder permissões ao Systems Manager, para que o Change Manager possa enviar notificações. Para obter mais informações, consulte Configurar os tópicos do Amazon SNS para as notificações do Change Manager.

   2. Escolha Adicionar notificação.

7. Na seção Change template reviewers (Revisores de modelos de alterações), selecione os usuários em sua organização ou conta para revisar novos modelos de alteração ou alterar versões de modelo antes que eles possam ser usados em suas operações.

Os revisores do modelo de alteração são responsáveis por verificar a adequação e a segurança dos modelos que outros usuários enviaram para uso nos fluxos de trabalho do runbook do Change Manager.

Selecione revisores de modelo de alteração fazendo o seguinte:

1. Escolha Adicionar.
2. Marque a caixa de seleção ao lado do nome de cada usuário, grupo ou função do IAM que você deseja atribuir como um revisor do modelo de alteração.
3. Selecione Add approvers (Adicionar aprovadores).

8. Selecione Enviar.

Depois de concluir este processo de configuração inicial, defina as configurações e práticas recomendadas do Change Manager seguindo as etapas em Tarefa 2: Configurar os aprovadores de eventos de congelamento de alterações e práticas recomendadas do Change Manager.

Tarefa 2: Configurar os aprovadores de eventos de congelamento de alterações e práticas recomendadas do Change Manager

Após concluir as etapas em Tarefa 1: Configurar o gerenciamento de identidade de usuário e revisores de modelo do Change Manager, você poderá designar revisores adicionais para solicitações de alteração durante os eventos de alteração do congelamento e especificar quais as práticas recomendadas disponíveis que você deseja permitir para as operações do Change Manager.

Um evento de congelamento de alterações significa que as restrições estão em vigor no calendário de alterações atual (o estado do calendário em AWS Systems Manager Change Calendar é `CLOSED`). Nesses casos, além de aprovadores regulares para solicitações de alteração, ou se a solicitação de alteração for criada usando um modelo que permita aprovações automáticas, os aprovadores do congelamento de alterações devem conceder permissão para que essa solicitação de alteração seja executada. Se não o fizerem, a alteração não será processada até que o estado do calendário seja novamente `OPEN`.

Para configurar os aprovadores de eventos de congelamento de alterações e práticas recomendadas do Change Manager

1. No painel de navegação, escolha Change Manager.

2. Escolha a guia Web Settings (Configurações da Web) e escolha Edit (Editar).
3. Na seção Approvers for change freeze events (Aprovadores para eventos de congelamento de alterações), selecione os usuários em sua organização ou conta que podem aprovar alterações para execução, mesmo quando o calendário em uso no Change Calendar estiver atualmente FECHADO.

   > **Note**
   >
   > Para permitir revisões do congelamento de alterações, você deve ativar a opção Check Change Calendar for restricted change events (Verificar o calendário de alterações para eventos de alteração restritos) emBest practices (Práticas recomendadas).

   Selecione aprovadores para eventos de congelamento de alteração fazendo o seguinte:

   1. Escolha Adicionar.
   2. Marque a caixa de seleção ao lado do nome de cada usuário, grupo ou função do IAM que você deseja atribuir como um aprovador de eventos de congelamento de alterações.
   3. Selecione Add approvers (Adicionar aprovadores).
4. Na seção Best practices (Práticas recomendadas), próxima à parte inferior da página, ative as práticas recomendadas que você deseja impor para cada uma das opções a seguir.

   - Opção: verifique o Calendário de alterações para eventos de alteração restritos

     Para especificar que o Change Manager verifique um calendário no Change Calendar para garantir que as alterações não sejam bloqueadas por eventos agendados, selecione primeiro a caixa de seleção Enabled (Habilitado) e, em seguida, selecione o calendário para verificar se há eventos restritos na lista Change Calendar (Alterar calendário).

     Para obter mais informações sobre o Change Calendar, consulte AWS Systems Manager Change Calendar.

   - Opção: tópico SNS para aprovadores em eventos fechados

     1. Escolha uma das opções a seguir para especificar o tópico Amazon Simple Notification Service (Amazon SNS) em sua conta a ser usado para enviar notificações aos aprovadores durante os eventos de congelamento de alterações. Observe que você também deve especificar os aprovadores na seção Approvers for change freeze events (Aprovadores

para eventos de congelamento de alterações), acima de Best practices (Práticas recomendadas)

- Insira um nome do recurso da Amazon (ARN) – Para ARN do tópico, insira o ARN de um tópico existente do Amazon SNS. Esse tópico pode estar em qualquer uma das contas da sua organização.
- Selecione um tópico do SNS existente: para Target notification topic (Tópico de notificação de destino), selecione o ARN de um tópico existente do Amazon SNS em sua Conta da AWS atual. (Essa opção não estará disponível se você ainda não tiver criado nenhum tópico do Amazon SNS na sua Conta da AWS e Região da AWS.)

> **Note**
>
> O tópico do Amazon SNS selecionado deve ser configurado para especificar as notificações que ele envia e os assinantes para os quais eles são enviados. Sua política de acesso também deve conceder permissões ao Systems Manager, para que o Change Manager possa enviar notificações. Para obter mais informações, consulte Configurar os tópicos do Amazon SNS para as notificações do Change Manager.

2. Escolha Adicionar notificação.

- Opção: exija monitores para todos os modelos

  Se você quiser garantir que todos os modelos da sua organização ou conta especifiquem um alarme do Amazon CloudWatch para monitorar a operação de alteração, marque a caixa de seleção Enabled (Habilitado).

- Opção: exija revisão e aprovação do modelo antes de usar

  Para garantir que nenhuma solicitação de alteração seja criada e que nenhum fluxo de trabalho do runbook seja executado, sem se basear em um modelo que tenha sido revisado e aprovado, selecione a opção Enabled (Habilitado).

5. Escolha Salvar.

Configurar os tópicos do Amazon SNS para as notificações do Change Manager

Você pode configurar o Change Manager, uma ferramenta do AWS Systems Manager, para enviar notificações a um tópico do Amazon Simple Notification Service (Amazon SNS) para eventos

relacionados a solicitações e modelos de alteração. Conclua as tarefas a seguir para receber notificações dos eventos do Change Manageraos quais você adiciona um tópico.

Tópicos

- Tarefa 1: Criar e assinar um tópico do Amazon SNS
- Tarefa 2: Atualizar a política de acesso do Amazon SNS
- Tarefa 3: Atualizar a política de acesso do AWS Key Management Service (opcional)

Tarefa 1: Criar e assinar um tópico do Amazon SNS

Primeiro, crie e se inscreva em um tópico do Amazon SNS. Para obter informações, consulte Criar um tópico do Amazon SNS e Assinar tópico do Amazon SNS no Guia do desenvolvedor do Amazon Simple Notification Service.

> **Note**
>
> Para receber notificações, você deverá especificar o nome do recurso da Amazon (ARN) de um tópico do Amazon SNS que estiver na mesma Região da AWS e Conta da AWS que a conta do administrador delegado.

Tarefa 2: Atualizar a política de acesso do Amazon SNS

Use o procedimento a seguir para atualizar a política de acesso do Amazon SNS, para que o Systems Manager possa publicar notificações do Change Manager no tópico do Amazon SNS que você criou na tarefa 1. Sem concluir esta tarefa, o Change Manager não terá permissão para enviar notificações para os eventos aos quais você adicionou o tópico.

1. Faça login no AWS Management Console e abra o console do Amazon SNS em https://console.aws.amazon.com/sns/v3/home.
2. No painel de navegação, escolha Tópicos.
3. Selecione o tópico criado na tarefa 1 e escolha Edit (Editar).
4. Expanda Access policy (Política de acesso).
5. Adicione e atualize o seguinte bloco do `Sid` à política existente e substitua cada *`espaço reservado para entrada do usuário`* pelas suas próprias informações.

```
{
```

```
    "Sid": "Allow Change Manager to publish to this topic",
    "Effect": "Allow",
    "Principal": {
        "Service": "ssm.amazonaws.com"
    },
    "Action": "sns:Publish",
    "Resource": "arn:aws:sns:region:account-id:topic-name",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": [
                "account-id"
            ]
        }
    }
}
```

Insira esse bloco após o bloco `Sid` existente e substitua *`region`*, *`account-id`* e *`topic_name`* pelos valores apropriados para o tópico que você criou.

6. Escolha Salvar alterações.

O sistema agora envia notificações para o tópico do Amazon SNS quando o tipo de evento que você adicionar ao tópico ocorrer.

> **Important**
>
> Se você configurou o tópico do Amazon SNS com uma chave de criptografia do AWS Key Management Service (AWS KMS) no lado do servidor, conclua a tarefa 3.

Tarefa 3: Atualizar a política de acesso do AWS Key Management Service (opcional)

Se você ativou a criptografia do lado do servidor do AWS Key Management Service (AWS KMS) para seu tópico do Amazon SNS, atualize também a política de acesso da AWS KMS key escolhida ao configurar o tópico. Use o procedimento a seguir para atualizar a política de acesso, de forma que o Systems Manager possa publicar as notificações de aprovação do Change Manager no tópico do Amazon SNS que você criou na tarefa 1.

1. Abra o console do AWS KMS em https://console.aws.amazon.com/kms.
2. No painel de navegação, escolha Chaves gerenciadas pelo cliente.

3. Selecione o ID da chave gerenciada do cliente que você escolheu ao criar o tópico.
4. Na seção Key Policy (Política de chaves), selecione Switch to policy view (Alternar para a visualização da política).
5. Escolha Editar.
6. Insira o seguinte bloco `Sid` após um dos blocos `Sid` existentes na política existente. Substitua cada *`espaço reservado para entrada do usuário`* por suas próprias informações.

```
{
    "Sid": "Allow Change Manager to decrypt the key",
    "Effect": "Allow",
    "Principal": {
        "Service": "ssm.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey*"
    ],
    "Resource": "arn:aws:kms:region:account-id:key/key-id",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": [
                "account-id"
            ]
        }
    }
}
```

7. Insira o seguinte bloco `Sid` após um dos blocos `Sid` existentes na política de recursos para ajudar a evitar o problema confused deputy entre serviços.

   Esse bloco usa as chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` com o objetivo de limitar as permissões concedidas pelo Systems Manager para outro serviço ao recurso.

   Substitua cada *`espaço reservado para entrada do usuário`* por suas próprias informações.

   JSON

```
{
```

```
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "Configure confused deputy protection for AWS KMS keys
 used in Amazon SNS topic when called from Systems Manager",
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": [
                "sns:Publish"
            ],
            "Resource": "arn:aws:sns:us-east-1:111122223333:topic-name",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:ssm:us-east-1:111122223333:*"
                },
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                }
            }
        }
    ]
}
```

8. Escolha Salvar alterações.

Configurar perfis e permissões para o Change Manager

Por padrão, o Change Manager não tem permissão para executar ações em suas instâncias. Você deve conceder acesso usando uma AWS Identity and Access ManagementFunção de serviço (IAM) ou Função assumida. Essa função permite Change Manager executar com segurança os fluxos de trabalho de runbook especificados em uma solicitação de alteração aprovada em seu nome. Essa função concede a AWS Security Token Service (AWS STS) AssumeRole (Função assuida) de Change Manager.

Ao fornecer essas permissões a uma função para agir em nome de usuários em uma organização, os usuários não precisam receber essa matriz de permissões. As ações permitidas pelas permissões são limitadas apenas a operações aprovadas.

Quando os usuários em sua conta ou organização criam uma solicitação de alteração, eles podem selecionar essa função para realizar as operações de alteração.

Você pode criar uma nova função assumida para Change Manager ou atualizar uma função existente com as permissões necessárias.

Se você precisar criar uma função de serviço para o Change Manager, conclua as tarefas a seguir.

Tarefas

- Tarefa 1: Criar uma política de função assumida para Change Manager
- Tarefa 2: Criando uma função assumida para Change Manager
- Tarefa 3: Anexar a política iam:PassRole a outras funções
- Tarefa 4: Adicionar políticas em linha a um papel assumido para invocar outros Serviços da AWS
- Tarefa 5: Configurar o acesso do usuário para Change Manager

Tarefa 1: Criar uma política de função assumida para Change Manager

Use o procedimento a seguir para criar a política que você anexará à sua função assumida Change Manager.

Para criar uma política de função assumida para Change Manager

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Policies e, em seguida, Create Policy.
3. Na página Criar política, escolha a guia JSON e substitua o conteúdo padrão pelo seguinte, que você modificará para suas próprias Change Manager operações nas etapas a seguir.

> **Note**
>
> Se você estiver criando uma política para usar com uma única Conta da AWS e não uma organização com várias contas e Regiões da AWS, é possível omitir o primeiro bloco de instrução. A `iam:PassRole` permissão não é necessária no caso de uma única conta usando Change Manager.

JSON

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

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/AWS-
SystemsManager-job-functionAdministrationRole",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ssm.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeDocument",
                "ssm:GetDocument",
                "ssm:StartChangeRequestExecution"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:automation-
definition/template-name:$DEFAULT",
                "arn:aws:ssm:us-east-1::document/template-name"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:ListOpsItemEvents",
                "ssm:GetOpsItem",
                "ssm:ListDocuments",
                "ssm:DescribeOpsItems"
            ],
            "Resource": "*"
        }
    ]
}
```

4. Para a ação `iam:PassRole`, atualize o valor `Resource` para incluir os ARNs de todas as funções de trabalho definidas para sua organização que você deseja conceder permissões para iniciar fluxos de trabalho de runbook.
5. Substitua os espaços reservados de *`region`*, *`account-id`*, *`template-name`*, *`delegated-admin-account-id`* e *`job-function`* por valores para suas operações Change Manager.

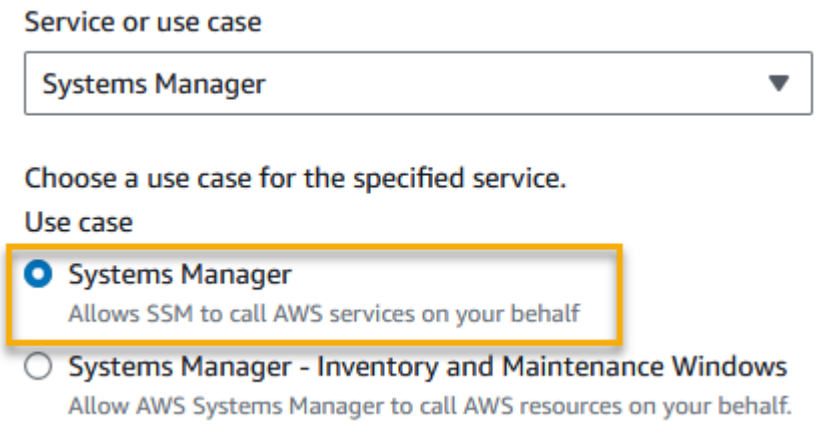
6. Para a segunda instrução `Resource`, modifique a lista para incluir todos os modelos de alteração para os quais você deseja conceder permissões. Como alternativa, especifique `"Resource": "*"` para conceder permissões para todos os modelos de alteração em sua organização.
7. Escolha Próximo: tags.
8. (Opcional) Adicione um ou mais pares de chave-valor de etiqueta para organizar, monitorar ou controlar acesso para essa política.
9. Escolha Próximo: revisar.
10. Na página Review policy (Revisar política), insira um nome na caixa Name (Nome), como **`MyChangeManagerAssumeRole`**, e insira uma descrição opcional.
11. Escolha Create policy (Criar política) e continue para Tarefa 2: Criando uma função assumida para Change Manager.

Tarefa 2: Criando uma função assumida para Change Manager

Use o procedimento a seguir para criar uma função assumida Change Manager, um tipo de função de serviço, para Change Manager.

Para criar uma função assumida para Change Manager

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Roles e Create role.
3. Em Select trusted entity (Selecionar entidade confiável), faça as seguintes escolhas:
    1. Em Trusted entity type (Tipo de entidade confiável), escolha service (Serviço da AWS)
    2. Em Casos de uso para outros Serviços da AWS, escolha Systems Manager.
    3. Escolha Systems Manager, como mostrado na imagem a seguir.



4. Escolha Próximo.
5. Na página Attached permissions policy (Política de permissões anexadas), procure a política de função assumida que você criou em Tarefa 1: Criar uma política de função assumida para Change Manager, por exemplo,**`MyChangeManagerAssumeRole`**.
6. Marque a caixa de seleção ao lado do nome da política assumir função e escolha Próximo: tags..
7. Em Role name (Nome da função), insira um nome para seu novo perfil da instância, como **`MyChangeManagerAssumeRole`**.
8. (Opcional) Em Description (Descrição), atualize a descrição deste perfil de instância.
9. (Opcional) Adicione um ou mais pares de chave-valor de etiqueta para organizar, monitorar ou controlar acesso para essa função.
10. Escolha Próximo: revisar.
11. (Opcional) Em Tags (Etiquetas), adicione um ou mais pares de valores etiqueta-chave para organizar, monitorar ou controlar o acesso para esse perfil e, em seguida, escolha Create role (Criar perfil). O sistema faz com que você retorne para a página Roles.
12. Selecione Create role (Criar função). O sistema faz com que você retorne para a página Roles.
13. Na página Roles (Funções), escolha a função que você acabou de criar para abrir a página Summary (Resumo).

Tarefa 3: Anexar a política **`iam:PassRole`** a outras funções

Use o procedimento a seguir para anexar a política `iam:PassRole` a um perfil de instância do IAM ou função de serviço do IAM. (O serviço Systems Manager usa perfis de instância do IAM para se comunicar com instâncias do EC2. Para nós gerenciados que não são do EC2 em um ambiente híbrido e multinuvem, utiliza-se um perfil de serviço do IAM.)

Ao anexar o política `iam:PassRole`, o serviço Change Manager pode passar permissões de função para outros serviços ou ferramentas do Systems Manager (Gerenciador de sistemas) ao executar fluxos de trabalho do runbook.

Para anexar a política **`iam:PassRole`** a um perfil de instância ou perfil de serviço do IAM

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Perfis.
3. Procure pela função assumida Change Manager que você criou, como **`MyChangeManagerAssumeRole`**, e escolha seu nome.
4. Na página Summary (Resumo) da função assumida, escolha a guia Permissions (Permissões).
5. Escolha Add permissions, Create inline policy (Adicionar permissões, Criar política em linha).
6. Na página Create policy (Criar política), selecione a guia Visual editor (Editor visual).
7. Selecione Service (Serviço) e, em seguida, selecione IAM.
8. Na caixa de texto Filter actions (Filtrar ações), insira **`PassRole`** e selecione a opção PassRole.
9. Ampliar os Resources (Recursos). Verifique se Specific (Específico) está selecionado e, em seguida, selecione Add ARN (Adicionar ARN).
10. No campo Specify ARN for role (Especificar ARN para função), insira o ARN da função de perfil da instância do IAM ou da função de serviço do IAM para a qual você deseja passar permissões de função assumida. O sistema preenche os campos Account (Conta) e Role name with path (Nome da função com caminho).
11. Escolha Add (Adicionar).
12. Escolha Review policy (Revisar política).
13. Em Name (Nome), insira um nome para identificar essa política e escolha Create policy (Criar política).

Mais informações

- Configurar permissões de instância obrigatórias para o Systems Manager
- Criar o perfil de serviço do IAM obrigatório para o Systems Manager em ambientes híbridos e multinuvem

Tarefa 4: Adicionar políticas em linha a um papel assumido para invocar outros Serviços da AWS

Quando uma solicitação de alteração invoca outros Serviços da AWS usando a função assumida Change Manager, a função assume deve ser configurada com permissão para invocar esses serviços. Esse requisito se aplica a todos os AWSRunbooks do Automation (runbooks AWS-*) que podem ser usados em uma solicitação de alteração, como os Runbooks `AWS-ConfigureS3BucketLogging`, `AWS-CreateDynamoDBBackup` e `AWS-RestartEC2Instance`. Esse requisito também se aplica a todos os runbooks personalizados criados que invoquem outros Serviços da AWS, usando ações que chamam outros serviços. Por exemplo, se você usar as ações `aws:executeAwsApi`, `aws:CreateStack` ou `aws:copyImage`, então você deve configurar um perfil de serviço com permissão para invocar esses serviços. É possível habilitar permissões para outros Serviços da AWS adicionando uma política em linha do IAM à função.

Para adicionar uma política em linha a uma função de assunção para invocar outros Serviços da AWS (console do IAM)

1. Faça login no AWS Management Console e abra o console do IAM, em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Perfis.
3. Na lista, escolha o nome da função assumida que você deseja atualizar, como `MyChangeManagerAssumeRole`.
4. Escolha a aba Permissões.
5. Escolha Add permissions, Create inline policy (Adicionar permissões, Criar política em linha).
6. Selecione a guia JSON.
7. Insira um documento de política JSON para os Serviços da AWS que você deseja chamar. Veja a seguir dois exemplos de documentos de política JSON

   Simple Storage Service (Amazon S3) **`PutObject`** e **`GetObject`** exemplo

   JSON

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

```
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

Amazon EC2 **CreateSnapshot** e **DescribeSnapShots** exemplo

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshot",
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":"ec2:DescribeSnapshots",
         "Resource":"*"
      }
   ]
}
```

Para obter detalhes sobre a linguagem da política do IAM, consulte a Referência da política JSON do IAM no Guia do usuário do IAM.

8. Ao concluir, selecione Review policy (Revisar política). O Validador de política indica se há qualquer erro de sintaxe.
9. Em Name (Nome), insira um nome para identificar a política que você está criando. Revise o Resumo da política para ver as permissões que são concedidas pela política. Em seguida, escolha Criar política para salvar seu trabalho.
10. Após a criação de uma política em linha, ela é automaticamente incorporada à sua função.

Tarefa 5: Configurar o acesso do usuário para Change Manager

Se seu usuário, grupo ou perfil tiver permissões de administrador atribuídas, você terá acesso ao Change Manager. Se você não tiver permissões de administrador, um administrador deverá atribuir a política gerenciada `AmazonSSMFullAccess` ou uma política que forneça permissões comparáveis ao seu usuário, grupo ou perfil.

Use o procedimento a seguir para configurar um usuário para usar o Change Manager. O usuário escolhido terá permissões para configurar e executar o Change Manager.

Dependendo da aplicação de identidade que você usa em sua organização, é possível selecionar qualquer uma das três opções disponíveis para configurar o acesso do usuário. Ao configurar o acesso do usuário, atribua ou adicione o seguinte:

1. Atribua a política `AmazonSSMFullAccess` ou uma política comparável que forneça permissões para acessar o Systems Manager.
2. Atribua a política `iam:PassRole`.
3. Adicione o ARN para o perfil assumido do Change Manager copiado no final de Tarefa 2: Criando uma função assumida para Change Manager.

Para conceder acesso, adicione as permissões aos seus usuários, grupos ou perfis:

- Usuários e grupos no AWS IAM Identity Center:

  Crie um conjunto de permissões. Siga as instruções em Criação de um conjunto de permissões no Guia do usuário do AWS IAM Identity Center.

- Usuários gerenciados no IAM com provedor de identidades:

  Crie um perfil para a federação de identidades. Siga as instruções em Criando um perfil para um provedor de identidades de terceiros (federação) no Guia do Usuário do IAM.

- Usuários do IAM:
  - Crie um perfil que seu usuário possa assumir. Siga as instruções em Criação de um perfil para um usuário do IAM no Guia do usuário do IAM.
  - (Não recomendado) Vincule uma política diretamente a um usuário ou adicione um usuário a um grupo de usuários. Siga as instruções em Adição de permissões a um usuário (console) no Guia do usuário do IAM.

Você terminou de configurar as funções necessárias para Change Manager. Agora é possível usar a função assumida ARN Change Manager em suas operações Change Manager.

Controlar o acesso a fluxos de trabalho do runbook de aprovação automática

Em cada modelo de alteração criado para sua organização ou conta, você pode especificar se as solicitações de alteração criadas com esse modelo podem ser executadas como solicitações de alteração aprovadas automaticamente, o que significa que elas são executadas automaticamente sem uma etapa de revisão (com exceção dos eventos de congelamento de alterações).

No entanto, você pode querer impedir determinados usuários, grupos ou grupos funções do AWS Identity and Access Management (IAM) de executar solicitações de alteração aprovadas automaticamente, mesmo que um modelo de alteração permita. Você pode fazer isso usando a chave de condição `ssm:AutoApprove` para a operação `StartChangeRequestExecution`em uma política do IAM atribuída ao usuário, grupo ou função do IAM.

Você pode adicionar a seguinte política como uma política em linha, onde a condição é especificada como `false` para impedir que os usuários executem solicitações de alteração aprovadas automaticamente.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
            {
            "Effect": "Allow",
            "Action": "ssm:StartChangeRequestExecution",
            "Resource": "*",
            "Condition": {
                "BoolIfExists": {
                    "ssm:AutoApprove": "false"
                }
            }
        }
    ]
}
```

Para obter informações sobre como especificar políticas em linha, consulte Políticas em linha e Adicionar e remover permissões de identidade do IAM no Manual do usuário do IAM.

Para obter mais informações sobre chaves de condição das políticas do Systems Manager, consulteChaves de condição do Systems Manager.

Trabalhar com o Change Manager

Com o Change Manager, uma ferramenta do AWS Systems Manager, os usuários em toda a sua organização ou em uma única Conta da AWS podem executar tarefas relacionadas a alterações para as quais receberam as permissões necessárias. As tarefas do Change Manager incluem o seguinte:

- Crie, revise e aprove ou rejeite os modelos de alteração.

  Um modelo de alteração é uma coleção de definições de configuração no Change Managerque definem itens como aprovações obrigatórias, runbooks disponíveis e opções de notificação para solicitações de alteração.
- Crie, revise e aprove ou rejeite as solicitações de alteração.

  Uma solicitação de alteração é uma solicitação no Change Manager para executar um runbook do Automation que atualiza um ou mais recursos na AWS ou em ambientes on-premises. Uma solicitação de alteração é criada usando um modelo de alteração.
- Especifique quais usuários em sua organização ou conta podem ser revisores para modelos e solicitações de alteração.
- Edite as definições de configuração, por exemplo, a forma como as identidades do usuário são gerenciadas no Change Manager e qual das opções de práticas recomendadas são impostas nas operações do Change Manager. Para obter mais informações sobre como definir essas configurações consulte Configurar as opções e práticas recomendadas do Change Manager.

Tópicos

- Trabalhar com modelos de alteração
- Trabalhar com solicitações de alteração
- Revisar detalhes, tarefas e cronogramas das solicitações de alteração (console)
- Visualizar contagens agregadas de solicitações de alteração (linha de comando)

Trabalhar com modelos de alteração

Um modelo de alteração é uma coleção de definições de configuração no Change Managerque definem itens como aprovações obrigatórias, runbooks disponíveis e opções de notificação para solicitações de alteração.

> **Note**
>
> A AWS fornece um exemplo de modelo de alteração Hello World que você pode usar para testar o Change Manager, uma ferramenta do AWS Systems Manager. No entanto, você cria seus próprios modelos de alteração para definir as alterações que deseja permitir aos recursos em sua organização ou conta.

As alterações feitas quando um fluxo de trabalho do runbook é executado são baseadas no conteúdo de um runbook do Automation. Em cada modelo de alteração criado, você pode incluir um ou mais runbooks do Automation, que o usuário que faz uma solicitação de alteração poderá escolher para executar durante a atualização. Você também pode criar modelos de alteração que permitem que os solicitantes escolham qualquer runbook do Automation disponível para a solicitação de alteração.

Para criar um modelo de alteração, você pode usar a opção Builder na página do console Create template (Criar modelo). Como alternativa, use a opção Editor. Crie manualmente conteúdo JSON ou YAML com a configuração desejada para o fluxo de trabalho do runbook. Você também pode usar uma ferramenta de linha de comando para criar um modelo de alteração com conteúdo JSON para o modelo de alteração armazenado em um arquivo externo.

Tópicos

- Experimente o modelo de alteração Hello World gerenciado pela AWS
- Criar modelos de alteração
- Revisar e aprovar ou rejeitar solicitações de alteração (modelos)
- Excluir modelos de alteração

Experimente o modelo de alteração **`Hello World`** gerenciado pela AWS

Você pode usar o modelo de alteração de exemplo `AWS-HelloWorldChangeTemplate`, que usa o runbook do Automation de exemplo `AWS-HelloWorld`, para testar o processo de revisão e aprovação depois de concluir a configuração do Change Manager, uma ferramenta do AWS Systems Manager. Esse modelo foi projetado para testar ou verificar suas permissões configuradas, atribuições de aprovador e processo de aprovação. A aprovação para usar esse modelo de alteração em sua organização ou conta já foi fornecida pela AWS. Porém, qualquer solicitação de alteração com base nesse modelo de alteração ainda deve ser aprovada pelos revisores em sua organização ou conta.

Em vez de fazer alterações em um recurso, o resultado do fluxo de trabalho do runbook associado a esse modelo é a impressão de uma mensagem na saída de uma etapa do Automation.

Antes de começar

Antes de começar, realize as seguintes tarefas:

- Se você estiver usando o AWS Organizations para gerenciar alterações em uma organização, conclua as tarefas de configuração da organização descritas em Configure o Change Manager para uma organização (conta de gerenciamento).
- Configure o Change Manager para a conta de administrador delegada ou conta única, conforme descrito em Configurar as opções e práticas recomendadas do Change Manager.

> **Note**
>
> Se você ativou a opção de prática recomendada Require monitors for all templates (Exigir monitores para todos os modelos) nas configurações do Change Manager, desative-a temporariamente enquanto testa o modelo de alteração Hello World.

Para experimentar o modelo de alteração Hello World gerenciado pela AWS

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Change Manager.
3. Selecione Create request (Criar solicitação).
4. Selecione o modelo de alteração chamado `AWS-HelloWorldChangeTemplate` e, em seguida, escolha Next (Próximo).
5. Para Name (Nome), insira um nome para a solicitação de alteração que facilite a identificação de sua finalidade, como **`MyChangeRequestTest`**.
6. Para que o restante das etapas crie sua solicitação de alteração, consulte Criar solicitações de alteração.

Próximas etapas

Para obter informações sobre como aprovar solicitações de alteração, consulte Revisar e aprovar ou rejeitar solicitações de alteração.

Para exibir o status e os resultados da solicitação de alteração, escolha o nome da solicitação de alteração na guia Requests (Solicitações) no Change Manager.

Criar modelos de alteração

Um modelo de alteração é uma coleção de definições de configuração no Change Managerque definem itens como aprovações obrigatórias, runbooks disponíveis e opções de notificação para solicitações de alteração.

É possível criar modelos de alteração para suas operações no Change Manager, uma ferramenta do AWS Systems Manager usando o console, que inclui opções do Builder e Editor ou ferramentas da linha de comando.

Tópicos

- Sobre as aprovações em seus modelos de alteração
- Criar modelos de alteração usando o Builder
- Criando modelos de alteração usando o Editor
- Criar modelos de alteração usando ferramentas de linha de comando

Sobre as aprovações em seus modelos de alteração

Para cada modelo de alteração criado, é possível especificar até cinco níveis de aprovação para as solicitações de alteração criadas nele. Para cada um desses níveis, você pode designar até cinco possíveis aprovadores. Um aprovador não está limitado a um único usuário. Também é possível especificar um grupo do IAM ou um perfil do IAM como um aprovador individual. Para grupos do IAM e perfis do IAM, um ou mais usuários pertencentes ao grupo ou perfil podem fornecer aprovações para receber o número total de aprovações requeridas para uma solicitação de alteração. Você também pode especificar mais aprovadores do que o requerido pelo seu modelo de alteração.

O Change Manager oferece suporte para duas abordagens principais para aprovações: aprovações por nível e aprovações por linha. Em algumas situações, também é possível realizar uma combinação dos dois tipos. Recomendamos usar somente aprovações por nível em suas operações do Change Manager.

Per-level approvals

Recomendado. A partir de 23 de janeiro de 2023, o Change Manager oferecerá suporte para aprovações por nível. Nesse modelo, para cada nível de aprovação em seu modelo de alteração, primeiro é necessário especificar quantas aprovações são requeridas para esse nível.

Em seguida, você especifica, no mínimo, esse número de aprovadores para o nível e pode especificar mais aprovadores. No entanto, somente o número de aprovadores por nível que você especificar precisará aprovar a solicitação de alteração. Por exemplo, é possível especificar cinco aprovadores, mas requerer três aprovações.

Para obter amostras de visualização do console e JSON desse tipo de aprovação, consulte the section called “Amostra de configuração de aprovação por nível”.

Per-line approvals

Com suporte para compatibilidade com versões anteriores. A versão original do Change Manager oferecia suporte somente para aprovações por linha. Nesse modelo, cada aprovador especificado para um nível de aprovação é representado como uma linha de aprovação. Cada aprovador precisava aprovar uma solicitação de alteração para que ela fosse aprovada nesse nível. Antes de 23 de janeiro de 2023, este era o único modelo com suporte para aprovações. Os modelos de alteração criados antes dessa data continuam a oferecer suporte às aprovações por linha, mas recomendamos usar as aprovações por nível.

Para obter amostras de visualização do console e JSON desse tipo de aprovação, consulte the section called “Amostra de configuração de aprovação por linha”.

Combined per-line and per-level approvals

Não recomendado. No console, a guia Builder não oferece mais suporte para a adição de aprovações por linha. No entanto, em alguns casos, é possível ter aprovações por linha e por nível em um modelo de alteração. Isso pode ocorrer caso você atualize um modelo de alteração criado antes de 23 de janeiro de 2023 ou crie ou atualize um modelo de alteração ao editar seu conteúdo YAML manualmente.

Para obter amostras de visualização do console e JSON desse tipo de aprovação, consulte the section called “Amostra de configuração de aprovação combinada por nível e por linha”.

Important

Embora seja possível criar um modelo de alteração que combine aprovações por linha e por nível, essa configuração não é recomendada ou necessária. Qualquer tipo de aprovação que requeira mais aprovações (aprovações por linha ou por nível) tem precedência. Por exemplo:

- Se um modelo de alteração especificar três aprovações por nível, mas cinco aprovações por linha, serão requeridas cinco aprovações.

- Se um modelo de alteração especificar quatro aprovações por nível, mas duas aprovações por linha, serão requeridas quatro aprovações.

É possível criar um nível que inclua aprovações por linha e por nível ao editar o conteúdo YAML ou JSON manualmente. Em seguida, a guia Builder exibirá os controles para especificação do número requerido de aprovações para o nível e para as linhas individuais. No entanto, os novos níveis adicionados usando o console ainda oferecem suporte somente às configurações de aprovação por nível.

Notificações e rejeições das solicitações de alteração

Notificações do Amazon SNS

Quando uma solicitação de alteração é criada usando seu modelo de alteração, as notificações são enviadas aos assinantes do tópico do Amazon Simple Notification Service (Amazon SNS) que foi designado para notificações de aprovação nesse nível. É possível especificar o tópico de notificação no modelo de alteração ou permitir que o usuário que está criando a solicitação de alteração o especifique.

Após o número mínimo de aprovações requeridas ser recebido em um nível, as notificações serão enviadas aos aprovadores inscritos no tópico do Amazon SNS para o próximo nível, e assim sucessivamente.

> **Important**
>
> Certifique-se de que os grupos, usuários e perfis do IAM designados em conjunto forneçam aprovadores suficientes para atender ao número requerido de aprovações especificado. Por exemplo, se você designar como aprovador um único grupo do IAM que contém três usuários, não poderá especificar que cinco aprovações sejam requeridas nesse nível, somente três ou menos.

Rejeições de solicitações de alteração

Não importa quantos níveis de aprovação e aprovadores você especificar, somente uma rejeição a uma solicitação de alteração é requerida para evitar que o fluxo de trabalho do runbook para essa solicitação ocorra.

Exemplos de tipos de aprovação do Change Manager

As amostras a seguir demonstram a visualização do console e o conteúdo JSON para os três tipos de tipos de aprovação no Change Manager.

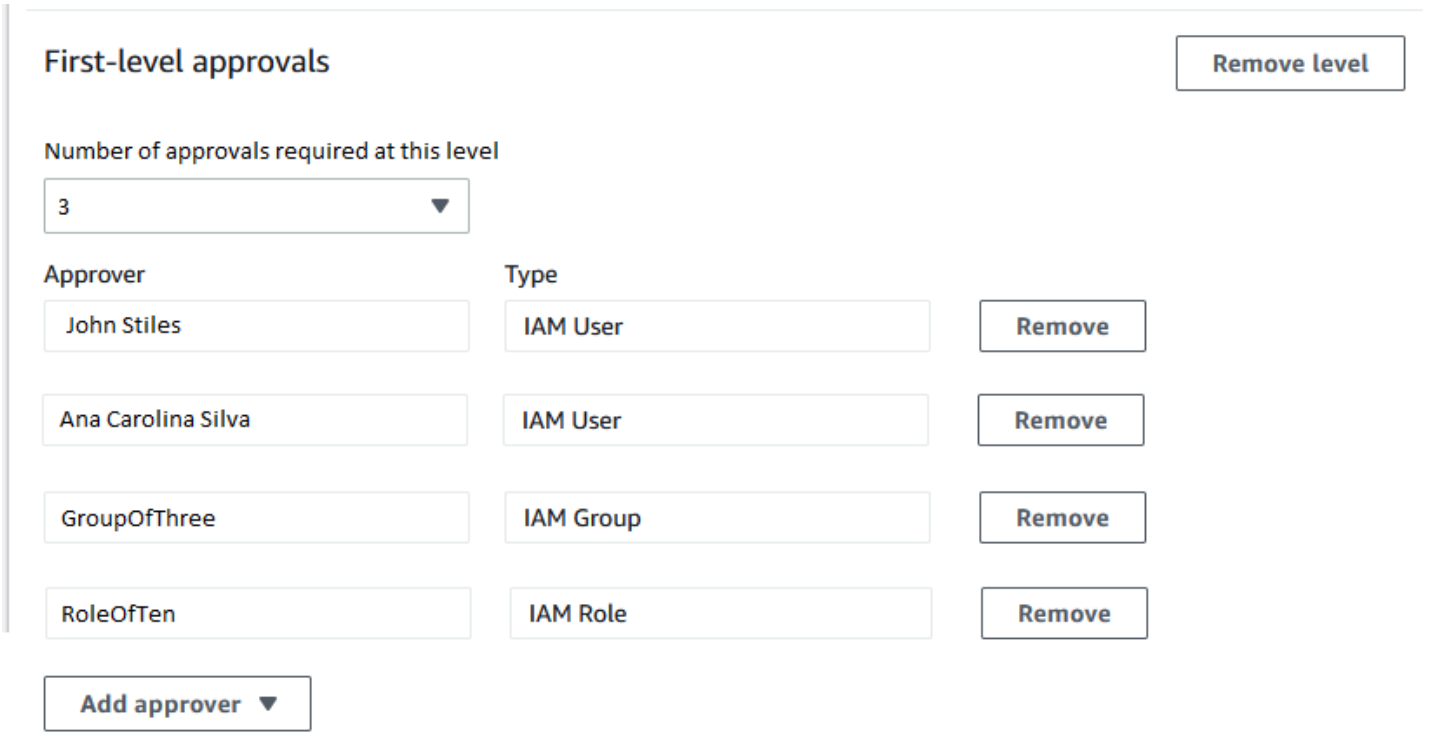
Tópicos

- Amostra de configuração de aprovação por nível
- Amostra de configuração de aprovação por linha
- Amostra de configuração de aprovação combinada por nível e por linha

Amostra de configuração de aprovação por nível

Na configuração do nível de aprovação por nível mostrada na imagem a seguir, são requeridas três aprovações. Essas aprovações poderão ser originadas de qualquer combinação de usuários, grupos e perfis do IAM que são especificados como aprovadores. Os aprovadores especificados incluem dois usuários do IAM (John Stiles e Ana Carolina Silva), um grupo de usuários que contém três membros (`GroupOfThree`) e um perfil de usuário que representa dez usuários (`RoleOfTen`).

Se todos os três usuários do grupo `GroupOfThree` aprovarem a solicitação de alteração, ela será aprovada para esse nível. Não é necessário receber uma aprovação de cada usuário, grupo ou perfil. O número mínimo de aprovações pode ser originado de qualquer combinação de aprovadores especificados. Recomendamos aprovações por nível para suas operações do Change Manager.



A amostra a seguir ilustra parte do código YAML para essa configuração.

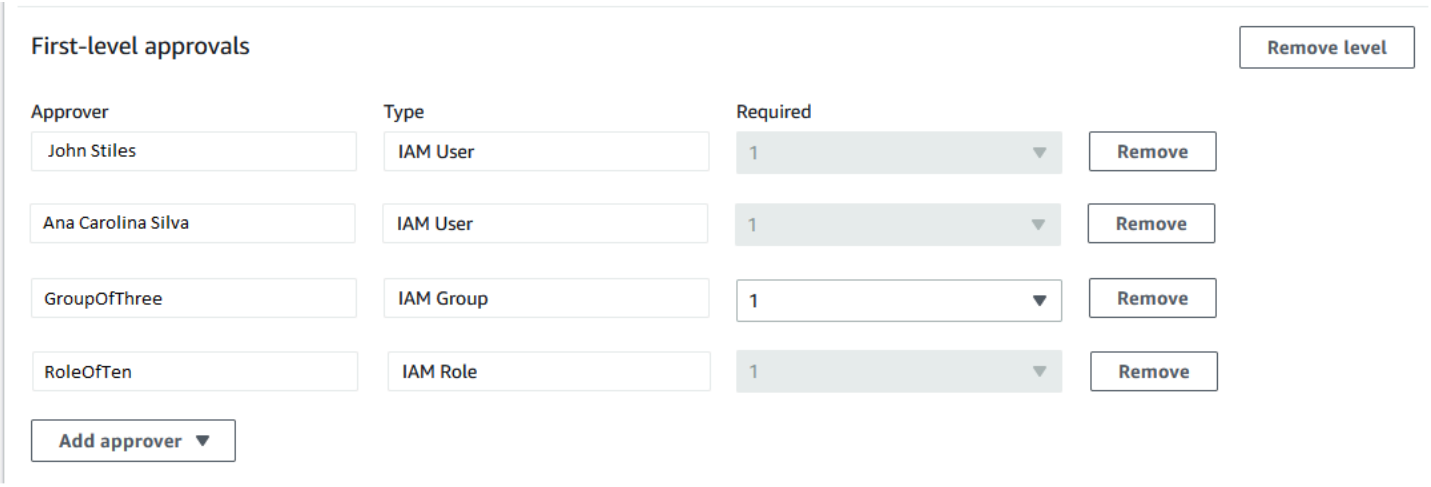
> **Note**
>
> Essa versão do código YAML inclui uma entrada adicional, `MinRequiredApprovals` (com uma letra maiúscula inicial `M`). O valor para essa entrada indica quantas aprovações são requeridas entre todos os revisores disponíveis. Observe também que o valor `minRequiredApprovals` (com uma letra minúscula inicial `m`) para cada aprovador na lista `Approvers` é `0` (zero). Isso indica que o aprovador poderá contribuir com as aprovações gerais, mas não é obrigado a fazê-lo.

```
schemaVersion: "0.3"
emergencyChange: false
autoApprovable: false
mainSteps:
  - name: ApproveAction1
    action: aws:approve
    timeoutSeconds: 604800
    inputs:
      Message: Please approve this change request
```

```
      MinRequiredApprovals: 3
      EnhancedApprovals:
        Approvers:
          - approver: John Stiles
            type: IamUser
            minRequiredApprovals: 0
          - approver: Ana Carolina Silva
            type: IamUser
            minRequiredApprovals: 0
          - approver: GroupOfThree
            type: IamGroup
            minRequiredApprovals: 0
          - approver: RoleOfTen
            type: IamRole
            minRequiredApprovals: 0
templateInformation: >
  #### What is the purpose of this change?
    //truncated
```

Amostra de configuração de aprovação por linha

Na configuração do nível de aprovação mostrada na imagem a seguir, quatro aprovadores estão especificados. Destes, temos dois usuários do IAM (John Stiles e Ana Carolina Silva), um grupo de usuários que contém três membros (`GroupOfThree`) e um perfil de usuário que representa dez usuários (`RoleOfTen`). As aprovações por linha oferecem suporte para a compatibilidade com versões anteriores, mas não são recomendadas.



Para que a solicitação de alteração seja aprovada nessa configuração de aprovação por linha, ela deve ser aprovada por todas as linhas de aprovadores: John Stiles, Ana Carolina Silva, um membro do grupo `GroupOfThree` e um membro do perfil `RoleOfTen`.

A amostra a seguir ilustra parte do código YAML para essa configuração.

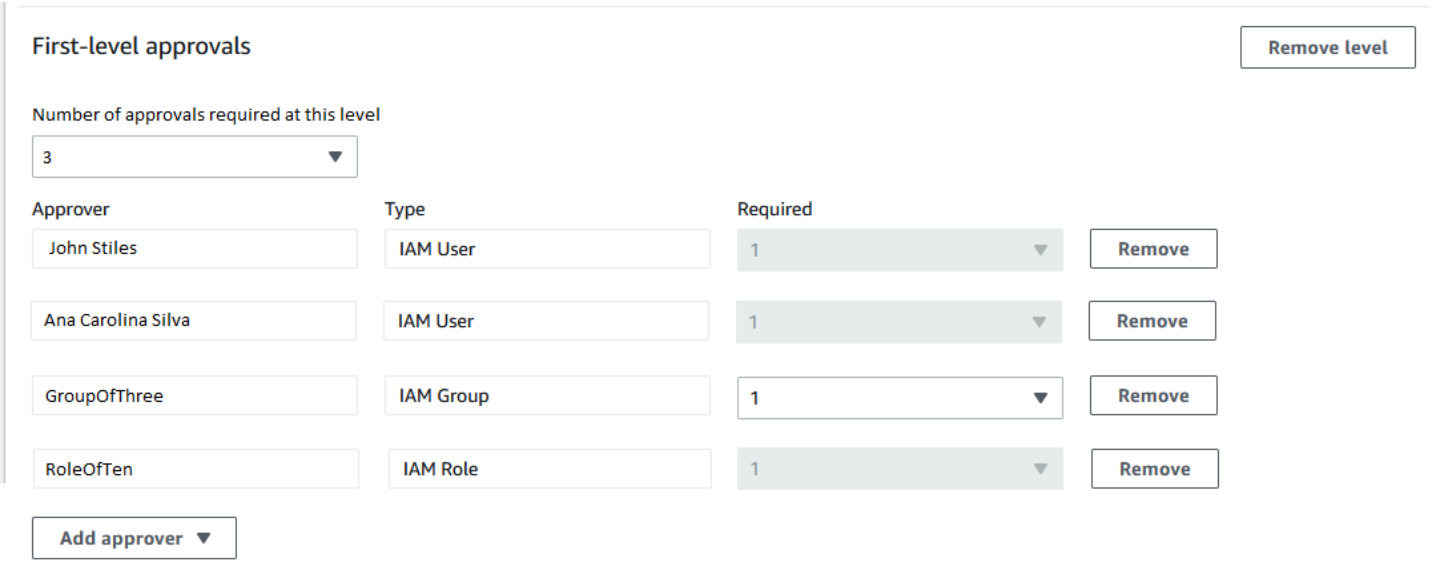
> **Note**
>
> Observe que o valor para cada aprovador `minRequiredApprovals` é `1`. Isso indica que uma aprovação é requerida de cada aprovador.

```
schemaVersion: "0.3"
emergencyChange: false
autoApprovable: false
mainSteps:
  - name: ApproveAction1
    action: aws:approve
    timeoutSeconds: 10000
    inputs:
      Message: Please approve this change request
      EnhancedApprovals:
        Approvers:
          - approver: John Stiles
            type: IamUser
            minRequiredApprovals: 1
          - approver: Ana Carolina Silva
            type: IamUser
            minRequiredApprovals: 1
          - approver: GroupOfThree
            type: IamGroup
            minRequiredApprovals: 1
          - approver: RoleOfTen
            type: IamRole
            minRequiredApprovals: 1
executableRunBooks:
  - name: AWS-HelloWorld
    version: $DEFAULT
templateInformation: >
  #### What is the purpose of this change?
    //truncated
```

Amostra de configuração de aprovação combinada por nível e por linha

Na configuração de aprovação combinada por nível e por linha mostrada na imagem a seguir, três aprovações são especificadas para o nível, mas quatro aprovações são especificadas para as

aprovações de itens de linha. O tipo de aprovação que requer mais aprovações tem precedência sobre o outro, portanto, quatro aprovações são requeridas para essa configuração. A aprovação combinada por nível e por linha não é recomendada.




```
schemaVersion: "0.3"
emergencyChange: false
autoApprovable: false
mainSteps:
  - name: ApproveAction1
    action: aws:approve
    timeoutSeconds: 604800
    inputs:
      Message: Please approve this change request
      MinRequiredApprovals: 3
      EnhancedApprovals:
        Approvers:
          - approver: John Stiles
            type: IamUser
            minRequiredApprovals: 1
          - approver: Ana Carolina Silva
            type: IamUser
            minRequiredApprovals: 1
          - approver: GroupOfThree
            type: IamGroup
            minRequiredApprovals: 1
          - approver: RoleOfTen
            type: IamRole
            minRequiredApprovals: 1
```

```
templateInformation: >
  #### What is the purpose of this change?
    //truncated
```

Tópicos

- Criar modelos de alteração usando o Builder
- Criando modelos de alteração usando o Editor
- Criar modelos de alteração usando ferramentas de linha de comando

Criar modelos de alteração usando o Builder

Usando o Builder para modelos de alteração no Change Manager, uma ferramenta do AWS Systems Manager, é possível configurar o fluxo de trabalho do runbook definido no modelo de alteração, sem precisar usar a sintaxe JSON ou YAML. Depois de especificar suas opções, o sistema converterá sua entrada no formato YAML que o Systems Manager pode usar para executar fluxos de trabalho do runbook.

Para criar um modelo de alteração usando o Builder

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Change Manager.
3. Selecione Criar modelo.
4. Para Name (Nome), insira um nome para o modelo que facilite a identificação de sua finalidade, como **`UpdateEC2LinuxAMI`**.
5. Na seção Change template details (Detalhes do modelo de alteração), faça o seguinte:

   - Em Descrição, forneça uma breve explicação de como e quando o modelo de alteração que você está criando deve ser usado.

     Esta descrição ajuda os usuários que criam solicitações de alteração a determinar se estão usando o modelo de alteração correto. Isso ajuda aqueles que revisam solicitações de alteração a entender se a solicitação deve ser aprovada.
   - Para Change template type (Alterar tipo de modelo), especifique se você está criando um modelo de alteração padrão ou um modelo de alteração de emergência.

     Um modelo de alteração de emergência é usado para situações em que uma alteração deve ser feita, mesmo que as alterações sejam bloqueadas por um evento no calendário em uso

pelo AWS Systems Manager Change Calendar. As solicitações de alteração criadas a partir de um modelo de alteração de emergência ainda devem ser aprovadas pelos aprovadores designados, mas as alterações solicitadas ainda podem ser executadas mesmo quando o calendário estiver bloqueado.

- Para Runbook options (Opções do Runbook), especifique os runbooks que os usuários podem escolher ao criar uma solicitação de alteração. Você pode adicionar um único runbook ou vários runbooks. Como alternativa, você pode permitir que os solicitantes especifiquem qual runbook usar. Em qualquer um desses casos, apenas um runbook pode ser incluído na solicitação de alteração.
- Para Runbook, selecione os nomes e as versões dos runbooks que os usuários podem escolher para as solicitações de alteração. Não importa quantos runbooks você adicionar ao modelo de alteração, somente um poderá ser selecionado em cada solicitação de alteração.

  Não especifique um runbook se escolher Any runbook can be used (Qualquer runbook pode ser usado).

> **Tip**
>
> Selecione um runbook e uma versão dele e escolha View (Exibir) para examinar o conteúdo do runbook na interface de documentos do Systems Manager.

6. Na seção Template information (Informações do modelo), use Markdown para inserir informações para usuários que criam solicitações de alteração com esse modelo de alteração. Fornecemos um conjunto de perguntas que você pode incluir para os usuários que criam solicitações de alteração, ou você pode adicionar outras informações e perguntas.

> **Note**
>
> Markdown é uma linguagem de marcação, que permite adicionar descrições de documentos no estilo wiki e etapas individuais dentro do documento. Para obter mais informações sobre como usar Markdown, consulte Usar Markdown na AWS.

Recomendamos fornecer perguntas para que os usuários respondam sobre suas solicitações de alteração para ajudar os aprovadores a decidir se devem ou não conceder cada solicitação de alteração, como listar as etapas manuais necessárias para serem executadas como parte da alteração e um plano de reversão.

> **Tip**
>
> Alterne entre Hide preview (Ocultar visualização) e Show preview(Exibir visualização) para ver o conteúdo de descrição à medida que você o cria.

7. Na seção Change request approvals (Aprovações de solicitação de alteração), faça o seguinte:

- (Opcional) Se você quiser permitir que as solicitações de alteração criadas neste modelo de alteração sejam executadas automaticamente, sem revisão por nenhum aprovador (com exceção dos eventos de congelamento de alterações), selecione Enable auto-approval (Habilitar aprovação automática).

> **Note**
>
> Ativar aprovações automáticas em um modelo de alteração fornece aos usuários a opção de ignorar os revisores. Eles ainda podem optar por especificar revisores ao criar uma solicitação de alteração. Portanto, você ainda deve especificar as opções do revisor no modelo de alteração.

> **Important**
>
> Se você ativar a aprovação automática para um modelo de alteração, os usuários poderão enviar solicitações de alteração usando esse modelo que não exige revisão antes de serem executados (com exceção dos aprovadores de eventos de congelamento de alterações). Se você quiser restringir um determinado usuário, grupo ou função do IAM de enviar solicitações de aprovação automática, você poderá usar uma condição em uma política do IAM para essa finalidade. Para obter mais informações, consulte Controlar o acesso a fluxos de trabalho do runbook de aprovação automática.

- Em Número de aprovações requeridas nesse nível, escolha o número de aprovações que as solicitações de alteração criadas nesse modelo de alteração devem receber para esse nível.
- Para adicionar aprovadores obrigatórios de primeiro nível, escolha Add approver (Adicionar aprovador) e, em seguida, escolha uma das seguintes opções:

- Aprovadores especificados pelo modelo: escolha um ou mais usuários, grupos ou funções do AWS Identity and Access Management (IAM) da sua conta para aprovar solicitações de alteração criadas com este modelo de alteração. Quaisquer solicitações de alteração criadas usando este modelo devem ser revisadas e aprovadas por cada aprovador especificado.
- Request specified approvers (Solicitar aprovadores especificados): o usuário que fizer a solicitação de alteração especificará os revisores no momento em que fizer a solicitação, e poderá escolher em uma lista de usuários da sua conta.

  O número que você inserir na coluna Required (Obrigatório) determinará quantos revisores devem ser especificados por uma solicitação de alteração que usar esse modelo de alteração.

  > **Important**
  >
  > Antes de 23 de janeiro de 2023, a guia Builder oferecia suporte à especificação somente de aprovações por linha. Os novos modelos de alteração e os novos níveis adicionados aos modelos de alteração existentes usando a guia Builder oferecem suporte somente a aprovações por nível. Recomendamos usar somente aprovações por nível em suas operações do Change Manager.
  > Para obter mais informações, consulte Sobre as aprovações em seus modelos de alteração.

- Para SNS topic to notify approvers (Tópico do SNS para notificar os aprovadores), faça o seguinte:

  1. Escolha uma das opções a seguir para especificar o tópico Amazon Simple Notification Service (Amazon SNS) em sua conta a ser usado para enviar notificações aos aprovadores avisando que a solicitação de alteração está pronta para avaliação.
     - Insira um nome do recurso da Amazon (ARN) – Para ARN do tópico, insira o ARN de um tópico existente do Amazon SNS. Esse tópico pode estar em qualquer uma das contas da sua organização.
     - Selecione um tópico do SNS existente: para Target notification topic (Tópico de notificação de destino), selecione o ARN de um tópico existente do Amazon SNS em sua Conta da AWS atual. (Essa opção não estará disponível se você ainda não tiver criado nenhum tópico do Amazon SNS na sua Conta da AWS e Região da AWS.)

- Especifique o tópico SNS quando a solicitação de alteração for criada: o usuário que cria uma solicitação de alteração pode especificar o tópico do Amazon SNS a ser usado para as notificações.

> **Note**
>
> O tópico do Amazon SNS selecionado deve ser configurado para especificar as notificações que ele envia e os assinantes para os quais eles são enviados. Sua política de acesso também deve conceder permissões ao Systems Manager, para que o Change Manager possa enviar notificações. Para obter mais informações, consulte Configurar os tópicos do Amazon SNS para as notificações do Change Manager.

2. Escolha Adicionar notificação.

8. (Opcional) Para adicionar outro nível de aprovadores, escolha Add approval level (Adicionar nível de aprovação) e escolha entre os aprovadores especificados pelo modelo e os aprovadores especificados pela solicitação para esse nível. Em seguida, escolha um tópico do SNS para notificar esse nível de aprovadores.

   Depois de todas as aprovações terem sido recebidas pelos aprovadores de primeiro nível, os aprovadores de segundo nível são notificados, e assim por diante.

   Você pode adicionar um máximo de cinco níveis de aprovadores em cada modelo. Você pode, por exemplo, exigir aprovações de usuários em funções técnicas para o primeiro nível e, em seguida, aprovação gerencial para o segundo nível.

9. Na seção Monitoring (Monitoramento), para CloudWatch alarm to monitor (Alarme do CloudWatch a ser monitorado), insira o nome de um alarme do Amazon CloudWatch na conta atual para monitorar o andamento dos fluxos de trabalho do runbook baseados nesse modelo.

> **Tip**
>
> Para criar um novo alarme, ou para rever as configurações de um alarme que deseja especificar, escolha Open the Amazon CloudWatch console (Abrir o console do Amazon CloudWatch). Para obter informações sobre o trabalho com os alarmes do CloudWatch, consulte Usar alarmes do CloudWatch no Guia do usuário do Amazon CloudWatch.

10. Na seção Notifications (Notificações), faça o seguinte:

1. Escolha uma das opções a seguir para especificar o tópico do Amazon SNS em sua conta a ser usado para enviar notificações sobre solicitações de alteração criadas usando este modelo de alteração:
   - Insira um nome do recurso da Amazon (ARN) – Para ARN do tópico, insira o ARN de um tópico existente do Amazon SNS. Esse tópico pode estar em qualquer uma das contas da sua organização.
   - Selecione um tópico do SNS existente: para Target notification topic (Tópico de notificação de destino), selecione o ARN de um tópico existente do Amazon SNS em sua Conta da AWS atual. (Essa opção não estará disponível se você ainda não tiver criado nenhum tópico do Amazon SNS na sua Conta da AWS e Região da AWS.)

   > **Note**
   >
   > O tópico do Amazon SNS selecionado deve ser configurado para especificar as notificações que ele envia e os assinantes para os quais eles são enviados. Sua política de acesso também deve conceder permissões ao Systems Manager, para que o Change Manager possa enviar notificações. Para obter mais informações, consulte Configurar os tópicos do Amazon SNS para as notificações do Change Manager.

2. Escolha Adicionar notificação.

11. (Opcional) Na seção Tags, aplique um ou mais pares de nome/valor de chave de tag ao modelo de alteração.

    Tags são metadados opcionais que você atribui a um recurso. Usando tags, você pode categorizar um recurso de diferentes formas, como por finalidade, proprietário ou ambiente. Por exemplo, você poderá marcar um modelo de alteração para identificar o tipo de alteração que ele faz e o ambiente em que ele é executado. Nesse caso, você pode especificar os seguintes pares de nome/valor:

    - `Key=TaskType,Value=InstanceRepair`
    - `Key=Environment,Value=Production`

12. Selecione Save and preview (Salvar e pré-visualizar).
13. Analise os detalhes do modelo de alteração que você estiver criando.

Se você quiser alterar o modelo de alteração antes de enviá-lo para análise, escolha Actions, Edit (Ações, Editar).

Se você estiver satisfeito com o conteúdo do modelo de alteração, escolhaSubmit for review (Enviar para análise). Os usuários em sua organização ou conta que foram especificados como revisores de modelo na guia Settings (Configurações) do Change Manager serão notificados de que um novo modelo de alteração está com a revisão pendente.

Se um tópico do Amazon SNS tiver sido especificado para modelos de alteração, as notificações serão enviadas quando o modelo de alteração for rejeitado ou aprovado. Se você não receber notificações relacionadas a este modelo de alteração, retorne ao Change Manager mais tarde para verificar seu status.

Criando modelos de alteração usando o Editor

Use as etapas neste tópico para configurar um modelo de alteração no Change Manager, uma ferramenta do AWS Systems Manager, inserindo JSON ou YAML em vez de usar os controles do console.

Para criar um modelo de alteração usando o Editor

1. No painel de navegação, escolha Change Manager.
2. Selecione Criar modelo.
3. Para Name (Nome), insira um nome para o modelo que facilite a identificação de sua finalidade, como **`RestartEC2LinuxInstance`**.
4. Acima de Change template details (Alterar detalhes do modelo), escolha Editor.
5. Na seção Document editor (Editor de documentos), selecione Edit (Editar) e insira o conteúdo JSON ou YAML para o modelo de alteração.

   Veja um exemplo a seguir.

   > **Note**
   >
   > O parâmetro `minRequiredApprovals` é usado para especificar quantos revisores em um nível especificado devem aprovar uma solicitação de alteração criada usando esse modelo.

Este exemplo demonstra dois níveis de aprovações. Você pode especificar até cinco níveis de aprovações, mas somente um nível é necessário.
No primeiro nível, o usuário específico "John-Doe" deve aprovar cada solicitação de alteração. Depois disso, quaisquer três membros da função do IAM `Admin` devem aprovar a solicitação de alteração.
Para obter mais informações sobre aprovações para modelos de alteração, consulte Sobre as aprovações em seus modelos de alteração.

YAML

```
description: >-
  This change template demonstrates the feature set available for creating
  change templates for Change Manager. This template starts a Runbook workflow
  for the Automation runbook called AWS-HelloWorld.
templateInformation: >
  ### Document Name: HelloWorldChangeTemplate

  ## What does this document do?

  This change template demonstrates the feature set available for creating
  change templates for Change Manager. This template starts a Runbook workflow
  for the Automation runbook called AWS-HelloWorld.

  ## Input Parameters

  * ApproverSnsTopicArn: (Required) Amazon Simple Notification Service ARN for
  approvers.

  * Approver: (Required) The name of the approver to send this request to.

  * ApproverType: (Required) The type of reviewer.
    * Allowed Values: IamUser, IamGroup, IamRole, SSOGroup, SSOUser

  ## Output Parameters

  This document has no outputs
schemaVersion: '0.3'
parameters:
  ApproverSnsTopicArn:
    type: String
    description: Amazon Simple Notification Service ARN for approvers.
```

```
  Approver:
    type: String
    description: IAM approver
  ApproverType:
    type: String
    description: >-
      Approver types for the request. Allowed values include IamUser, IamGroup,
      IamRole, SSOGroup, and SSOUser.
executableRunBooks:
  - name: AWS-HelloWorld
    version: '1'
emergencyChange: false
autoApprovable: false
mainSteps:
  - name: ApproveAction1
    action: 'aws:approve'
    timeoutSeconds: 3600
    inputs:
      Message: >-
        A sample change request has been submitted for your review in Change
        Manager. You can approve or reject this request.
      EnhancedApprovals:
        NotificationArn: '{{ ApproverSnsTopicArn }}'
        Approvers:
          - approver: John-Doe
            type: IamUser
            minRequiredApprovals: 1
  - name: ApproveAction2
    action: 'aws:approve'
    timeoutSeconds: 3600
    inputs:
      Message: >-
        A sample change request has been submitted for your review in Change
        Manager. You can approve or reject this request.
      EnhancedApprovals:
        NotificationArn: '{{ ApproverSnsTopicArn }}'
        Approvers:
          - approver: Admin
            type: IamRole
            minRequiredApprovals: 3
```

JSON

```
{
   "description": "This change template demonstrates the feature set available
 for creating
  change templates for Change Manager. This template starts a Runbook workflow
  for the Automation runbook called AWS-HelloWorld",
   "templateInformation": "### Document Name: HelloWorldChangeTemplate\n\n
    ## What does this document do?\n
    This change template demonstrates the feature set available for creating
 change templates for Change Manager.
    This template starts a Runbook workflow for the Automation runbook called
 AWS-HelloWorld.\n\n
    ## Input Parameters\n* ApproverSnsTopicArn: (Required) Amazon Simple
 Notification Service ARN for approvers.\n
    * Approver: (Required) The name of the approver to send this request to.\n
    * ApproverType: (Required) The type of reviewer.  * Allowed Values: IamUser,
 IamGroup, IamRole, SSOGroup, SSOUser\n\n
    ## Output Parameters\nThis document has no outputs\n",
   "schemaVersion": "0.3",
   "parameters": {
      "ApproverSnsTopicArn": {
         "type": "String",
         "description": "Amazon Simple Notification Service ARN for approvers."
      },
      "Approver": {
         "type": "String",
         "description": "IAM approver"
      },
      "ApproverType": {
         "type": "String",
         "description": "Approver types for the request. Allowed values include
 IamUser, IamGroup, IamRole, SSOGroup, and SSOUser."
      }
   },
   "executableRunBooks": [
      {
         "name": "AWS-HelloWorld",
         "version": "1"
      }
   ],
   "emergencyChange": false,
   "autoApprovable": false,
```

```
  "mainSteps": [
     {
        "name": "ApproveAction1",
        "action": "aws:approve",
        "timeoutSeconds": 3600,
        "inputs": {
           "Message": "A sample change request has been submitted for your
review in Change Manager. You can approve or reject this request.",
           "EnhancedApprovals": {
              "NotificationArn": "{{ ApproverSnsTopicArn }}",
              "Approvers": [
                 {
                    "approver": "John-Doe",
                    "type": "IamUser",
                    "minRequiredApprovals": 1
                 }
              ]
           }
        }
     },
       {
        "name": "ApproveAction2",
        "action": "aws:approve",
        "timeoutSeconds": 3600,
        "inputs": {
           "Message": "A sample change request has been submitted for your
review in Change Manager. You can approve or reject this request.",
           "EnhancedApprovals": {
              "NotificationArn": "{{ ApproverSnsTopicArn }}",
              "Approvers": [
                 {
                    "approver": "Admin",
                    "type": "IamRole",
                    "minRequiredApprovals": 3
                 }
              ]
           }
        }
     }
  ]
}
```

6. Selecione Save and preview (Salvar e pré-visualizar).

7. Analise os detalhes do modelo de alteração que você estiver criando.

   Se você quiser alterar o modelo de alteração antes de enviá-lo para análise, escolha Actions, Edit (Ações, Editar).

   Se você estiver satisfeito com o conteúdo do modelo de alteração, escolhaSubmit for review (Enviar para análise). Os usuários em sua organização ou conta que foram especificados como revisores de modelo na guia Settings (Configurações) do Change Manager serão notificados de que um novo modelo de alteração está com a revisão pendente.

   Se um tópico do Amazon Simple Notification Service (Amazon SNS) tiver sido especificado para modelos de alteração, as notificações serão enviadas quando o modelo de alteração for rejeitado ou aprovado. Se você não receber notificações relacionadas a este modelo de alteração, retorne ao Change Manager mais tarde para verificar seu status.

Criar modelos de alteração usando ferramentas de linha de comando

Os procedimentos a seguir descrevem como usar o AWS Command Line Interface (AWS CLI) (no Linux,macOS ou Windows) ou o AWS Tools for Windows PowerShell para criar uma solicitação de alteração no Change Manager, uma ferramenta do AWS Systems Manager.

Para criar um novo modelo de alteração

1. Instale e configure a AWS CLI ou o Ferramentas da AWS para PowerShell, caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI e Instalar o Ferramentas da AWS para PowerShell.

2. Crie um arquivo JSON em sua máquina local com um nome, como `MyChangeTemplate.json` e, em seguida, cole o conteúdo do modelo de alteração nesse arquivo.

   > **Note**
   >
   > Os modelos de alteração usam uma versão do esquema 0.3 que não inclui o mesmo suporte dos runbooks do Automation.

   Veja um exemplo a seguir.

> **Note**
>
> O parâmetro `minRequiredApprovals` é usado para especificar quantos revisores em um nível especificado devem aprovar uma solicitação de alteração criada usando esse modelo.
>
> Este exemplo demonstra dois níveis de aprovações. Você pode especificar até cinco níveis de aprovações, mas somente um nível é necessário.
>
> No primeiro nível, o usuário específico "John-Doe" deve aprovar cada solicitação de alteração. Depois disso, quaisquer três membros da função do IAM `Admin` devem aprovar a solicitação de alteração.
>
> Para obter mais informações sobre aprovações para modelos de alteração, consulte Sobre as aprovações em seus modelos de alteração.

```
{
   "description": "This change template demonstrates the feature set available for
 creating
  change templates for Change Manager. This template starts a Runbook workflow
  for the Automation runbook called AWS-HelloWorld",
   "templateInformation": "### Document Name: HelloWorldChangeTemplate\n\n
    ## What does this document do?\n
    This change template demonstrates the feature set available for creating change
 templates for Change Manager.
    This template starts a Runbook workflow for the Automation runbook called AWS-
HelloWorld.\n\n
    ## Input Parameters\n* ApproverSnsTopicArn: (Required) Amazon Simple
 Notification Service ARN for approvers.\n
    * Approver: (Required) The name of the approver to send this request to.\n
    * ApproverType: (Required) The type of reviewer.  * Allowed Values: IamUser,
 IamGroup, IamRole, SSOGroup, SSOUser\n\n
    ## Output Parameters\nThis document has no outputs\n",
  "schemaVersion": "0.3",
  "parameters": {
     "ApproverSnsTopicArn": {
        "type": "String",
        "description": "Amazon Simple Notification Service ARN for approvers."
     },
     "Approver": {
        "type": "String",
        "description": "IAM approver"
```

```
      },
      "ApproverType": {
         "type": "String",
         "description": "Approver types for the request. Allowed values include
IamUser, IamGroup, IamRole, SSOGroup, and SSOUser."
      }
   },
   "executableRunBooks": [
      {
         "name": "AWS-HelloWorld",
         "version": "1"
      }
   ],
   "emergencyChange": false,
   "autoApprovable": false,
   "mainSteps": [
      {
         "name": "ApproveAction1",
         "action": "aws:approve",
         "timeoutSeconds": 3600,
         "inputs": {
            "Message": "A sample change request has been submitted for your review
in Change Manager. You can approve or reject this request.",
            "EnhancedApprovals": {
               "NotificationArn": "{{ ApproverSnsTopicArn }}",
               "Approvers": [
                  {
                     "approver": "John-Doe",
                     "type": "IamUser",
                     "minRequiredApprovals": 1
                  }
               ]
            }
         }
      },
        {
         "name": "ApproveAction2",
         "action": "aws:approve",
         "timeoutSeconds": 3600,
         "inputs": {
            "Message": "A sample change request has been submitted for your review
in Change Manager. You can approve or reject this request.",
            "EnhancedApprovals": {
               "NotificationArn": "{{ ApproverSnsTopicArn }}",
```

```
                "Approvers": [
                   {
                      "approver": "Admin",
                      "type": "IamRole",
                      "minRequiredApprovals": 3
                   }
                ]
             }
          }
       }
   ]
}
```

3. Execute o comando a seguir para criar o modelo de alteração.

   Linux & macOS

   ```
   aws ssm create-document \
       --name MyChangeTemplate \
       --document-format JSON \
       --document-type Automation.ChangeTemplate \
       --content file://MyChangeTemplate.json \
       --tags Key=tag-key,Value=tag-value
   ```

   Windows

   ```
   aws ssm create-document ^
       --name MyChangeTemplate ^
       --document-format JSON ^
       --document-type Automation.ChangeTemplate ^
       --content file://MyChangeTemplate.json ^
       --tags Key=tag-key,Value=tag-value
   ```

   PowerShell

   ```
   $json = Get-Content -Path "C:\path\to\file\MyChangeTemplate.json" | Out-String
   New-SSMDocument `
       -Content $json `
       -Name "MyChangeTemplate" `
       -DocumentType "Automation.ChangeTemplate" `
       -Tags "Key=tag-key,Value=tag-value"
   ```

Para obter informações sobre outras opções que você pode especificar, consulte create-document.

O sistema retorna informações como estas.

```
{
   "DocumentDescription":{
      "CreatedDate":1.585061751738E9,
      "DefaultVersion":"1",
      "Description":"Use this template to update an EC2 Linux AMI. Requires one
      approver specified in the template and an approver specified in the
 request.",
      "DocumentFormat":"JSON",
      "DocumentType":"Automation",
      "DocumentVersion":"1",
      "Hash":"0d3d879b3ca072e03c12638d0255ebd004d2c65bd318f8354fcde820dEXAMPLE",
      "HashType":"Sha256",
      "LatestVersion":"1",
      "Name":"MyChangeTemplate",
      "Owner":"123456789012",
      "Parameters":[
         {
            "DefaultValue":"",
            "Description":"Level one approvers",
            "Name":"LevelOneApprovers",
            "Type":"String"
         },
         {
            "DefaultValue":"",
            "Description":"Level one approver type",
            "Name":"LevelOneApproverType",
            "Type":"String"
         },
  "cloudWatchMonitors": {
      "monitors": [
         "my-cloudwatch-alarm"
      ]
  }
      ],
      "PlatformTypes":[
         "Windows",
         "Linux"
```

```
        ],
        "SchemaVersion":"0.3",
        "Status":"Creating",
        "Tags":[

        ]
    }
}
```

Os usuários em sua organização ou conta que foram especificados como revisores de modelo na guia Settings (Configurações) do Change Manager serão notificados de que um novo modelo de alteração está com a revisão pendente.

Se um tópico do Amazon Simple Notification Service (Amazon SNS) tiver sido especificado para modelos de alteração, as notificações serão enviadas quando o modelo de alteração for rejeitado ou aprovado. Se você não receber notificações relacionadas a este modelo de alteração, retorne ao Change Manager mais tarde para verificar seu status.

Revisar e aprovar ou rejeitar solicitações de alteração (modelos)

Se você foi designado como revisor para modelos de alteração no Change Manager, uma ferramenta do AWS Systems Manager, você será notificado quando um novo modelo de alteração, ou uma nova versão de um modelo de alteração, estiver aguardando sua revisão. Um tópico de um Amazon Simple Notification Service (Amazon SNS) envia as notificações.

> **Note**
>
> Essa funcionalidade depende da conta ser configurada para usar um tópico do Amazon SNS para enviar notificações sobre a revisão do modelo de alteração. Para obter informações sobre como especificar um tópico de notificação do avaliador de modelos, consulte Tarefa 1: Configurar o gerenciamento de identidade de usuário e revisores de modelo do Change Manager.

Para revisar o modelo de alteração, você pode seguir o link em sua notificação ou entrar no AWS Management Console diretamente e executar as etapas deste procedimento.

Para revisar e aprovar ou rejeitar um modelo de alteração

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.

2. No painel de navegação, escolha Change Manager.
3. Na seção Change templates (Modelos de alterações), na parte inferior da guia Overview (Visão geral), selecione o número em Pending review (Análise pendente).
4. Na lista Change templates (Modelos de alterações), localize e escolha o nome do modelo de alteração a ser analisado.
5. Na página de resumo, revise o conteúdo proposto do modelo de alteração e siga um destes procedimentos:

   - Para aprovar o modelo de alteração, que permite que ele seja usado em solicitações de alteração, escolha Approve (Aprovar).
   - Para rejeitar o modelo de alteração, o que impedirá que ele seja usado em solicitações de alteração, escolha Reject (Rejeitar).

Excluir modelos de alteração

Este tópico descreve como excluir modelos que você criou no Change Manager, uma ferramenta do Systems Manager. Se você estiver usando o Change Manager para uma organização, esse procedimento é executado em sua conta de administrador delegado.

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Change Manager.
3. Escolha a guia Modelos.
4. Escolha o nome do modelo a ser excluído.
5. Escolha Actions (Ações), Delete template (Excluir modelo).
6. Na caixa de diálogo de confirmação, insira **DELETE** e selecione Delete (Excluir).

Trabalhar com solicitações de alteração

Uma solicitação de alteração é uma solicitação no Change Manager para executar um runbook do Automation que atualiza um ou mais recursos na AWS ou em ambientes on-premises. Uma solicitação de alteração é criada usando um modelo de alteração.

Quando você cria uma solicitação de alteração no Change Manager, uma ferramenta do AWS Systems Manager, um ou mais aprovadores em sua organização ou conta devem revisar e aprovar a solicitação. Sem as aprovações necessárias, o fluxo de trabalho do runbook, que faz as alterações solicitadas, não tem permissão para ser executado.

Tópicos

- Criar solicitações de alteração
- Revisar e aprovar ou rejeitar solicitações de alteração

Criar solicitações de alteração

Quando você cria uma solicitação de alteração no Change Manager, uma ferramenta do AWS Systems Manager, o modelo de alteração selecionado normalmente faz o seguinte:

- Designa aprovadores para a solicitação de alteração ou especifica quantas aprovações são necessárias
- Especifica o tópico do Amazon Simple Notification Service (Amazon SNS) a ser usado para notificar os aprovadores sobre a solicitação de alteração
- Especifica um alarme do Amazon CloudWatch para monitorar o fluxo de trabalho do runbook para a solicitação de alteração
- Identifica quais runbooks do Automation você pode escolher para fazer a alteração solicitada

Em alguns casos, um modelo de alteração pode ser configurado para que você especifique seu próprio runbook do Automation a ser usado e para especificar quem deve revisar e aprovar a solicitação.

> **Important**
>
> Se você usar o Change Manager em uma organização, recomendamos sempre fazer as alterações na conta de administrador delegado. Embora seja possível fazer alterações de outras contas na organização, essas alterações não serão relatadas ou visíveis na conta do administrador delegado.

Tópicos

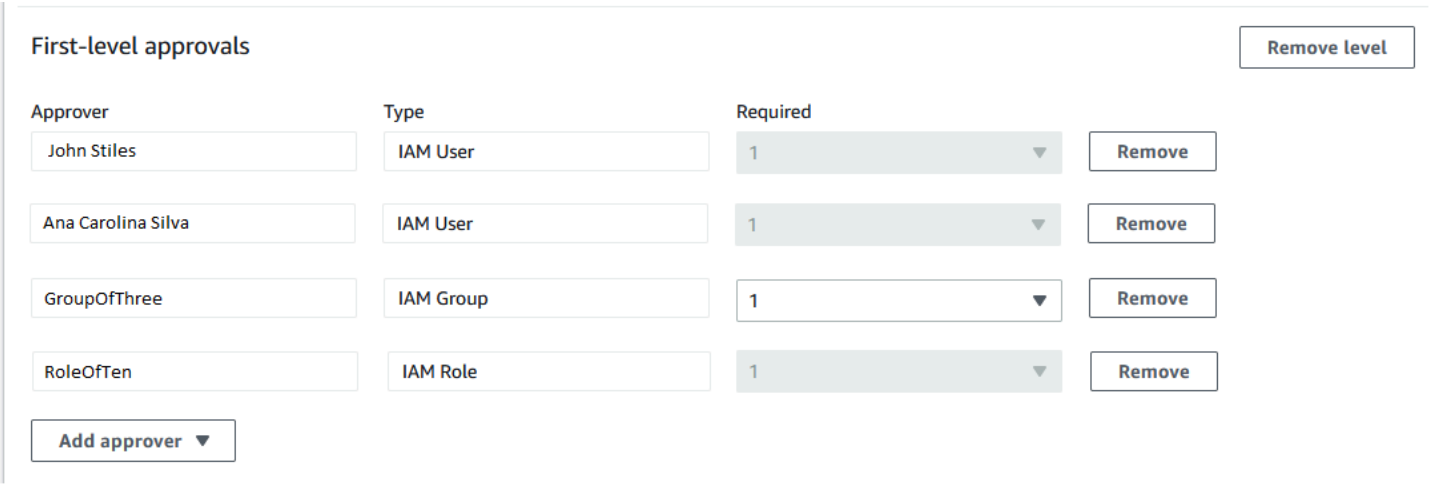
- Sobre as aprovações de solicitação de alteração
- Criar solicitações de alteração (console)
- Criar solicitações de alteração (AWS CLI)

Sobre as aprovações de solicitação de alteração

Dependendo dos requisitos especificados em um modelo de alteração, as solicitações de alteração criadas nele podem requerer aprovações de até cinco níveis antes que o fluxo de trabalho do runbook para a solicitação possa ocorrer. Para cada um desses níveis, o criador do modelo poderá especificar até cinco possíveis aprovadores. Um aprovador não está limitado a um único usuário. Nesse sentido, um aprovador também pode ser um grupo do IAM ou um perfil do IAM. Para grupos do IAM e perfis do IAM, um ou mais usuários pertencentes ao grupo ou perfil podem fornecer aprovações para receber o número total de aprovações requeridas para uma solicitação de alteração. Os criadores do modelo também poderão especificar mais aprovadores do que o requerido pelo modelo de alteração.

Fluxos de trabalho de aprovação originais e atualizados e/ou aprovações

Ao usar modelos de alteração criados antes de 23 de janeiro de 2023, uma aprovação deve ser recebida de cada aprovador especificado para que a solicitação de alteração seja aprovada nesse nível. Por exemplo, na configuração do nível de aprovação mostrada na imagem a seguir, quatro aprovadores estão especificados. Os aprovadores especificados incluem dois usuários (John Stiles e Ana Carolina Silva), um grupo de usuários que contém três membros (GroupOfThree) e um perfil de usuário que representa dez usuários (RoleOfTen).

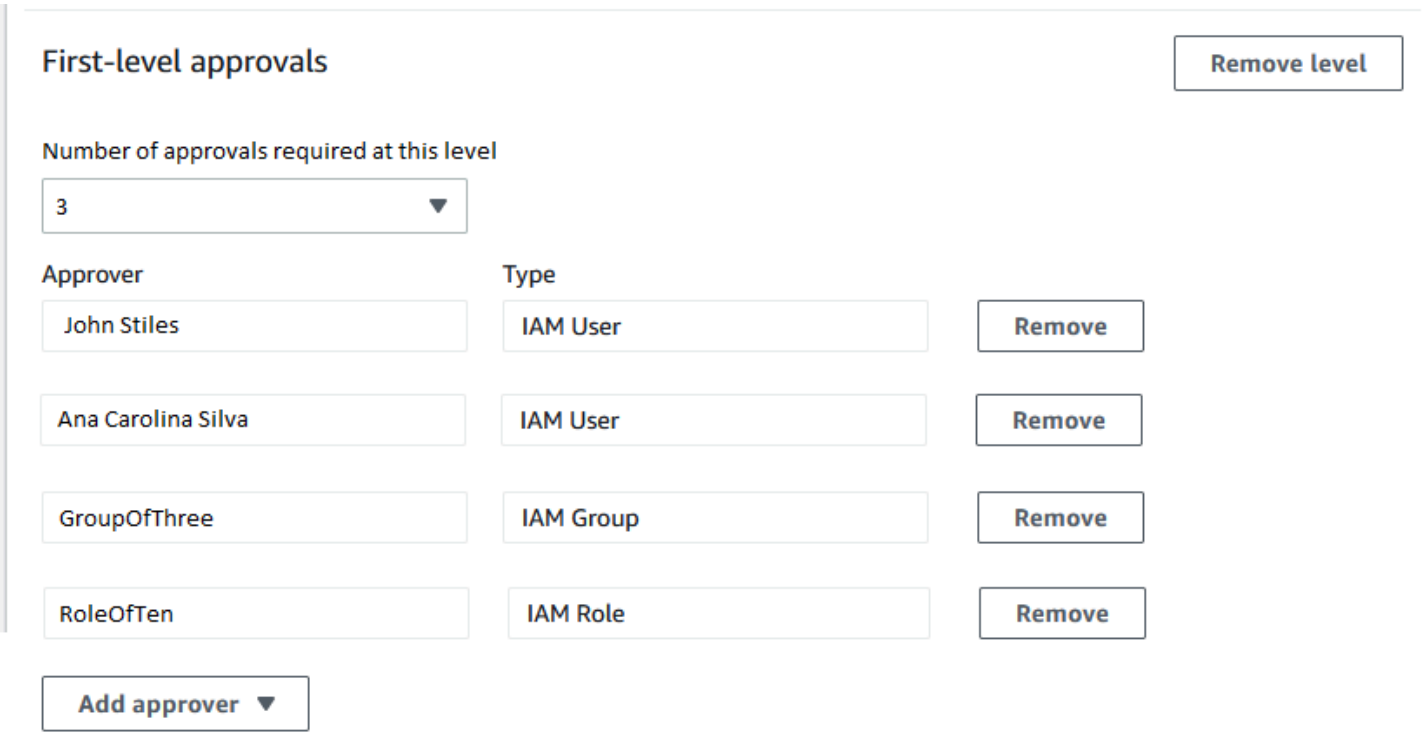


Para que a solicitação de alteração seja aprovada nesse nível, ela deve ser aprovada por John Stiles, Ana Carolina Silva, um membro do grupo `GroupOfThree` e um membro do perfil `RoleOfTen`.

Ao usar modelos de alteração criados em ou após 23 de janeiro de 2023, para cada nível de aprovação, os criadores de modelos poderão especificar um número total geral de aprovações requeridas. Essas aprovações poderão ser originadas de qualquer combinação de usuários, grupos

e perfis especificados como aprovadores. Um modelo de alteração pode requerer somente uma aprovação para um nível, mas especificar, por exemplo, dois usuários individuais, dois grupos e um perfil como possíveis aprovadores.

Por exemplo, na área do nível de aprovação mostrada na imagem a seguir, são requeridas três aprovações. Os aprovadores especificados pelo modelo incluem dois usuários (John Stiles e Ana Carolina Silva), um grupo de usuários que contém três membros (`GroupOfThree`) e um perfil de usuário que representa dez usuários (`RoleOfTen`).



Se todos os três usuários do grupo `GroupOfThree` aprovarem a sua solicitação de alteração, ela será aprovada para esse nível. Não é necessário receber uma aprovação de cada usuário, grupo ou perfil. O número mínimo de aprovações pode ser originado de qualquer combinação de possíveis aprovadores.

Quando a solicitação de alteração é criada, as notificações são enviadas aos assinantes do tópico do Amazon SNS que foi especificado para notificações de aprovação nesse nível. É possível que o criador do modelo de alteração tenha especificado o tópico de notificação que deve ser usado ou permitido que você o especificasse.

Após o número mínimo de aprovações requeridas ser recebido em um nível, as notificações serão enviadas aos aprovadores inscritos no tópico do Amazon SNS para o próximo nível, e assim sucessivamente.

Não importa quantos níveis de aprovação e aprovadores estejam especificados, somente uma rejeição a uma solicitação de alteração é requerida para evitar que o fluxo de trabalho do runbook para essa solicitação ocorra.

Criar solicitações de alteração (console)

O procedimento a seguir descreve como criar uma solicitação de alteração usando o console do Systems Manager.

Para criar uma solicitação de alteração (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Change Manager.
3. Selecione Create request (Criar solicitação).
4. Pesquise e selecione um modelo de alteração que você deseja usar para essa solicitação de alteração.
5. Escolha Próximo.
6. Para Name (Nome), insira um nome para a solicitação de alteração que facilite a identificação de sua finalidade, como **`UpdateEC2LinuxAMI-us-east-2`**.
7. Para Runbook, selecione o runbook que quiser usar para fazer a alteração solicitada.

   > **Note**
   >
   > Se a opção para selecionar um runbook não estiver disponível, o autor do modelo de alteração especificou qual runbook deve ser usado.

8. Para Change request information (Alterar as informações da solicitação), use Markdown para fornecer informações adicionais sobre a solicitação de alteração para ajudar os revisores a decidir se devem aprovar ou rejeitar a solicitação de alteração. O autor do modelo que você está usando pode ter fornecido instruções ou perguntas para você responder.

   > **Note**
   >
   > Markdown é uma linguagem de marcação, que permite adicionar descrições de documentos no estilo wiki e etapas individuais dentro do documento. Para obter mais informações sobre como usar Markdown, consulte Usar Markdown na AWS.

9. Na seção Workflow start time (Horário de início do fluxo de trabalho), escolha uma das seguintes opções:

   - Execute a operação em um horário agendado: para Requested start time (Horário de início solicitado), insira a data e a hora proposta para executar o fluxo de trabalho do runbook para esta solicitação. Para Estimated end time (Hora estimada para o término), insira a data e a hora em que você espera que o fluxo de trabalho do runbook seja concluído. (Esse tempo é apenas uma estimativa que você fornece aos revisores).

     > **Tip**
     >
     > SelecioneView Change Calendar (Exibir o calendário de alterações) para verificar se há eventos de bloqueio para o tempo especificado.

   - Execute a operação o mais rápido possível após a aprovação: se a solicitação de alteração for aprovada, o fluxo de trabalho do runbook será executado assim que houver um período sem restrições no qual as alterações podem ser feitas.

10. Na seção Change request approvals (Aprovações de solicitação de alteração), faça o seguinte:

    1. Se as opções de Approval type (Tipo de aprovação) forem apresentadas, escolha uma das seguintes opções:

       - Aprovação automática: o modelo de alteração selecionado é configurado para permitir que as solicitações de alteração sejam executadas automaticamente sem revisão por nenhum aprovador. Continue na etapa 11.

         > **Note**
         >
         > As permissões especificadas nas políticas do IAM que regem seu uso do Systems Manager não devem restringir o envio de solicitações de alteração de aprovação automática para que elas sejam executadas automaticamente.

       - Especifique os aprovadores: você deve adicionar um ou mais usuários, grupos ou funções do IAM para revisar e aprovar essa solicitação de alteração.

> **Note**
>
> Você pode optar por especificar revisores mesmo que as permissões especificadas nas políticas do IAM que regem seu uso do Systems Manager permitam que você execute solicitações de alteração de aprovação automática.

2. Escolha Add approver (Adicionar aprovador) e selecione um ou mais usuários, grupos ou funções do AWS Identity and Access Management (IAM) nas listas de revisores disponíveis.

> **Note**
>
> Um ou mais aprovadores já podem estar especificados. Isso significa que os aprovadores obrigatórios já estão especificados no modelo de alteração selecionado. Não é possível remover esses aprovadores da solicitação. Se o botão Adicionar aprovador não estiver disponível, o modelo escolhido não permitirá que revisores adicionais sejam adicionados às solicitações.

Para obter mais informações sobre aprovações para solicitações de alteração, consulte Sobre as aprovações de solicitação de alteração.

3. Em SNS topic to notify approvers (Tópico do SNS para notificar aprovadores), escolha uma das opções a seguir para especificar o tópico do Amazon SNS em sua conta a ser usado para enviar notificações aos aprovadores que você estiver adicionando a essa solicitação de alteração.

> **Note**
>
> Se a opção para especificar um tópico do Amazon SNS não estiver disponível, o modelo de alteração selecionado já especificará o tópico do Amazon SNS a ser usado.

- Insira um nome do recurso da Amazon (ARN) – Para ARN do tópico, insira o ARN de um tópico existente do Amazon SNS. Esse tópico pode estar em qualquer uma das contas da sua organização.

- Selecione um tópico do SNS existente: para Target notification topic (Tópico de notificação de destino), selecione o ARN de um tópico existente do Amazon SNS em sua conta atual. (Essa opção não estará disponível se você ainda não tiver criado nenhum tópico do Amazon SNS na sua Conta da AWS e Região da AWS.)

> **Note**
>
> O tópico do Amazon SNS selecionado deve ser configurado para especificar as notificações que ele envia e os assinantes para os quais eles são enviados. Sua política de acesso também deve conceder permissões ao Systems Manager, para que o Change Manager possa enviar notificações. Para obter mais informações, consulte Configurar os tópicos do Amazon SNS para as notificações do Change Manager.

4. Escolha Adicionar notificação.

11. Escolha Próximo.
12. Para IAM role (Função do IAM), selecione uma função do IAM na sua conta atual que tenha as permissões necessárias para executar os runbooks especificados para esta solicitação de alteração.

    Essa função também é referida como a função de serviço, ou função assumida, para o Automation. Para obter mais informações sobre essa função, consulte Configurar a automação.
13. Na seção Deployment location (Local de implantação), escolha uma das seguintes opções:

> **Note**
>
> Se você estiver usando o Change Manager com uma única Conta da AWS e não com uma organização configurada no AWS Organizations, você não precisará especificar um local de implantação.

- Aplique alterações a esta conta. O fluxo de trabalho do runbook é executado somente na conta atual. Para uma organização, isso significa a conta de administrador delegado.
- Aplique alterações a várias unidades organizacionais (UOs). Faça o seguinte:

1. Para Accounts and organizational units (OUs) (Contas e unidades organizacionais, UOs), insira o ID de uma conta de membro em sua organização, no formato **123456789012** ou o ID de uma unidade organizacional, no formato **o-o96EXAMPLE**.
2. (Opcional) Em Execution role name (Nome da função de execução), insira o nome da função do IAM na conta de destino ou a UO que tiver as permissões necessárias para executar os runbooks especificados para esta solicitação de alteração. Todas as contas em qualquer UO que você especificar devem usar o mesmo nome para essa função.
3. (Opcional) Escolha Add another target location (Adicionar outro local de destino) para cada conta adicional ou UO que você deseja especificar e repita as etapas A e B.
4. Em Target (Região da AWS de destino), selecione a região na qual fazer a alteração, como `Ohio (us-east-2)` para a região Leste dos EUA (Ohio).
5. Expanda Rate control (Controle de taxa).

   Para Concurrency (Simultaneidade), insira um número e, na lista, selecione se isso representa o número ou porcentagem de contas em que o fluxo de trabalho do runbook pode ser executado ao mesmo tempo.

   Em Error threshold (Limite de erro), insira um número e, na lista, selecione se isso representa o número ou porcentagem de contas em que o fluxo de trabalho do runbook pode falhar antes que a operação seja interrompida.

14. Na seção Deployment targets (Destinos das implantações), faça o seguinte:

    1. Escolha uma das seguintes opções:

       - Recurso único: a alteração deve ser feita para apenas um recurso. Por exemplo, um único nó ou uma única Amazon Machine Image (AMI), dependendo da operação definida nos runbooks para essa solicitação de alteração.
       - Vários recursos: para Parameter (Parâmetro), selecione um dos parâmetros disponíveis nos runbooks para essa solicitação de alteração. Essa seleção reflete o tipo de recurso que está sendo atualizado.

         Por exemplo, se o runbook para esta solicitação de alteração for `AWS-RetartEC2Instance`, você poderá escolher `InstanceId` e, em seguida, definir quais instâncias serão atualizadas selecionando uma das seguintes opções:

         - Especifique as tags: insira um par chave-valor com o qual todos os recursos a serem atualizados serão marcados.

- Escolha um grupo de recursos: escolha o nome do grupo de recursos ao qual todos os recursos a serem atualizados pertencem.
- Especifique valores do parâmetro: identifique os recursos a serem atualizados na seção Parâmetros do runbook.
- Selecione todas as instâncias como destino: faça a alteração em todos os nós gerenciados nos locais de destino.

2. Se você escolheu Multiple resources (Vários recursos), expanda Rate control (Controle de taxa).

   Para Concurrency (Simultaneidade), insira um número e, na lista, selecione se isso representa o número ou porcentagem de destinos nos quais o fluxo de trabalho do runbook pode ser atualizado ao mesmo tempo.

   Em Error threshold (Limite de erros), insira um número e, na lista, selecione se isso representa o número ou a porcentagem de destinos onde a atualização pode falhar antes que a operação seja interrompida.

15. Se você escolheu Specify parameter values (Especificar valores de parâmetro) para atualizar vários recursos na etapa anterior: na seção Runbook parameters (Parâmetros do runbook), especifique valores para os parâmetros de entrada necessários. Os valores de parâmetro que você deve fornecer são baseados no conteúdo dos runbooks do Automation associados ao modelo de alteração escolhido.

    Por exemplo, se o modelo de alteração usar o runbook do `AWS-RetartEC2Instance`, você deverá inserir um ou mais IDs de instância para o parâmetro InstanceId. Como alternativa, selecione Show interactive instance picker (Mostrar o seletor de instâncias interativas) e selecione as instâncias disponíveis uma a uma.

16. Escolha Próximo.

17. Na página Review and submit (Analise e envie), verifique novamente os recursos e as opções que você especificou para esta solicitação de alteração.

    Selecione o botão Edit (Editar) para qualquer seção na qual você deseja fazer alterações.

    Quando estiver satisfeito com os detalhes da solicitação de alteração, escolhaSubmit for approval (Enviar para aprovação).

Se um tópico do Amazon SNS foi especificado no modelo de alteração escolhido para a solicitação, as notificações são enviadas quando a solicitação for rejeitada ou aprovada. Se você não receber notificações para a solicitação, retorne ao Change Manager para verificar o status da sua solicitação.

Criar solicitações de alteração (AWS CLI)

Você pode criar uma solicitação de alteração usando o AWS Command Line Interface (AWS CLI) especificando opções e parâmetros para a solicitação de alteração em um arquivo JSON e usando o comando `--cli-input-json` para incluí-la em seu comando.

Para criar uma solicitação de alteração (AWS CLI)

1. Instale e configure a AWS CLI ou o Ferramentas da AWS para PowerShell, caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI e Instalar o Ferramentas da AWS para PowerShell.
2. Crie um arquivo JSON em sua máquina local com um nome, como `MyChangeRequest.json` e cole o seguinte conteúdo nesse arquivo:

   Substitua os *`espaços reservados`* pelos valores da solicitação de alteração.

   > **Note**
   >
   > Esse JSON de exemplo cria uma solicitação de alteração usando o modelo de alteração `AWS-HelloWorldChangeTemplate` e o runbook `AWS-HelloWorld`. Para ajudar você a adaptar esse exemplo para as suas próprias solicitações de alteração, consulte StartChangeRequestExecution na AWS Systems Manager API Reference para obter informações sobre todos os parâmetros disponíveis
   > Para obter mais informações sobre aprovações para solicitações de alteração, consulte Sobre as aprovações de solicitação de alteração.

```
{
    "ChangeRequestName": "MyChangeRequest",
    "DocumentName": "AWS-HelloWorldChangeTemplate",
    "DocumentVersion": "$DEFAULT",
    "ScheduledTime": "2021-12-30T03:00:00",
    "ScheduledEndTime": "2021-12-30T03:05:00",
    "Tags": [
```

```
        {
            "Key": "Purpose",
            "Value": "Testing"
        }
    ],
    "Parameters": {
        "Approver": [
            "JohnDoe"
        ],
        "ApproverType": [
            "IamUser"
        ],
        "ApproverSnsTopicArn": [
            "arn:aws:sns:us-east-2:123456789012:MyNotificationTopic"
        ]
    },
    "Runbooks": [
        {
            "DocumentName": "AWS-HelloWorld",
            "DocumentVersion": "1",
            "MaxConcurrency": "1",
            "MaxErrors": "1",
            "Parameters": {
                "AutomationAssumeRole": [
                    "arn:aws:iam::123456789012:role/MyChangeManagerAssumeRole"
                ]
            }
        }
    ],
    "ChangeDetails": "### Document Name: HelloWorldChangeTemplate\n\n## What does
 this document do?\nThis change template demonstrates the feature set available
 for creating change templates for Change Manager. This template starts a Runbook
 workflow for the Automation document called AWS-HelloWorld.\n\n## Input Parameters
\n* ApproverSnsTopicArn: (Required) Amazon Simple Notification Service ARN for
 approvers.\n* Approver: (Required) The name of the approver to send this request
 to.\n* ApproverType: (Required) The type of reviewer.\n  * Allowed Values:
 IamUser, IamGroup, IamRole, SSOGroup, SSOUser\n\n## Output Parameters\nThis
 document has no outputs \n"
}
```

3. No diretório em que você criou o arquivo JSON, execute o seguinte comando:

```
aws ssm start-change-request-execution --cli-input-json file://MyChangeRequest.json
```

O sistema retorna informações como estas.

```
{
    "AutomationExecutionId": "b3c1357a-5756-4839-8617-2d2a4EXAMPLE"
}
```

Revisar e aprovar ou rejeitar solicitações de alteração

Se estiver designado como revisor para uma solicitação de alteração no Change Manager, uma ferramenta do AWS Systems Manager, você será notificado por meio de um tópico do Amazon Simple Notification Service (Amazon SNS) quando uma nova solicitação de alteração estiver aguardando sua análise.

> **Note**
>
> Essa funcionalidade depende de um Amazon SNS estar especificado no modelo de alteração para enviar notificações de revisão. Para mais informações, consulte Configurar os tópicos do Amazon SNS para as notificações do Change Manager.

Para revisar a solicitação de alteração, você pode seguir o link em sua notificação ou entrar no AWS Management Console diretamente e executar as etapas deste procedimento.

> **Note**
>
> Se um tópico do Amazon SNS for atribuído aos revisores em um modelo de alteração, as notificações serão enviadas aos assinantes do tópico quando a solicitação de alteração alterar o status.
> Para obter mais informações sobre aprovações para solicitações de alteração, consulte Sobre as aprovações de solicitação de alteração.

Revisar e aprovar ou rejeitar solicitações de alteração (console)

Os procedimentos a seguir descrevem como usar o console do Systems Manager (Gerenciador de sistemas) para revisar e aprovar ou rejeitar solicitações de mudança.

Para revisar e aprovar ou rejeitar uma única solicitação de alteração

1. Abra o link na notificação por e-mail que você recebeu e entre no AWS Management Console, que o direcionará para a avaliação da solicitação de alteração.
2. Na página de resumo, revise o conteúdo proposto da solicitação de alteração.

   Para aprovar a solicitação de alteração, escolha Approve (Aprovar). Na caixa de diálogo, forneça os comentários que você deseja adicionar para essa aprovação e escolha Approve (Aprovar). O fluxo de trabalho do runbook representado por essa solicitação começa a ser executado quando agendado ou assim que as alterações não estiverem bloqueadas por nenhuma restrição.

   - ou -

   Para rejeitar a solicitação de alteração, escolha Reject (Rejeitar). Na caixa de diálogo, forneça os comentários que você deseja adicionar para essa rejeição e escolha Reject (Rejeitar).

Para revisar e aprovar ou rejeitar solicitações de alteração em massa

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Change Manager.
3. Escolha a guia Approvals (Aprovações).
4. (Opcional) Revise os detalhes das solicitações pendentes de sua aprovação escolhendo o nome de cada solicitação e, em seguida, retorne à guia Approvals (Aprovações).
5. Marque a caixa de seleção de cada solicitação de alteração que você deseja aprovar.

   - ou -

   Marque a caixa de seleção de cada solicitação de alteração que você deseja rejeitar.
6. Na caixa de diálogo, forneça os comentários que deseja adicionar para esta aprovação ou rejeição.
7. Dependendo se você está aprovando ou rejeitando as solicitações de alteração selecionadas, escolha Aprovar ou Rejeitar.

Revisar e aprovar ou rejeitar uma solicitação de alteração (linha de comando)

O procedimento a seguir descreve como usar o AWS Command Line Interface (AWS CLI) (no Linux, macOS ou Windows) para analisar e aprovar ou rejeitar uma solicitação de alteração.

Para revisar e aprovar ou rejeitar as solicitações de alteração

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Crie um arquivo JSON em sua máquina local que especifique os parâmetros para a chamada da AWS CLI.

```
{
  "OpsItemFilters":
  [
    {
      "Key": "OpsItemType",
      "Values": ["/aws/changerequest"],
      "Operator": "Equal"
    }
  ],
  "MaxResults": number
}
```

   Você pode filtrar os resultados de um aprovador específico, determinando o Amazon Resource Name (ARN) do aprovador no arquivo JSON. Aqui está um exemplo.

```
{
  "OpsItemFilters":
  [
    {
      "Key": "OpsItemType",
      "Values": ["/aws/changerequest"],
      "Operator": "Equal"
    },
    {
      "Key": "ChangeRequestByApproverArn",
      "Values": ["arn:aws:iam::account-id:user/user-name"],
      "Operator": "Equal"
    }
  ],
  "MaxResults": number
}
```

3. Execute o comando a seguir para visualizar o número máximo de solicitações de alteração especificadas no arquivo JSON.

Linux & macOS

```
aws ssm describe-ops-items \
--cli-input-json file://filename.json
```

Windows

```
aws ssm describe-ops-items ^
--cli-input-json file://filename.json
```

4. Execute o comando a seguir para aprovar ou rejeitar uma solicitação de alteração.

   Linux & macOS

   ```
   aws ssm send-automation-signal \
       --automation-execution-id ID \
       --signal-type Approve_or_Reject \
       --payload Comment="message"
   ```

   Windows

   ```
   aws ssm send-automation-signal ^
   --automation-execution-id ID ^
       --signal-type Approve_or_Reject ^
       --payload Comment="message"
   ```

   Se um tópico do Amazon SNS tiver sido especificado no modelo de alteração escolhido para a solicitação, as notificações serão enviadas quando a solicitação for rejeitada ou aprovada. Se você não receber notificações para a solicitação, retorne ao Change Manager para verificar o status da sua solicitação. Para obter informações sobre outras opções ao usar esse comando, consulte send-automation-signal na seção AWS Systems Manager da AWS CLI Command Reference.

Revisar detalhes, tarefas e cronogramas das solicitações de alteração (console)

Você pode visualizar informações sobre uma solicitação de alteração, incluindo solicitações para as quais as alterações foram processadas, no painel do Change Manager, uma ferramenta do AWS Systems Manager. Esses detalhes incluem um link para a operação do Automation, que executa os

runbooks que fazem a alteração. Um ID de execução do Automation é gerado quando a solicitação é criada, mas o processo não é executado até que todas as aprovações tenham sido fornecidas e não haja restrições para bloquear a alteração.

Para revisar detalhes, tarefas e cronogramas das solicitações de alteração

1. No painel de navegação, escolha Change Manager.
2. Selecione a guia Requests (Solicitações).
3. Na seção Change requests (Solicitações de alteração), procure a solicitação de alteração que você quiser analisar.

   Você pode usar as opções Create date range (Criar intervalo de datas) para limitar os resultados a um período de tempo específico.

   Você pode filtrar solicitações pelas seguintes propriedades:

   - `Status`
   - `Request ID`
   - `Approver`
   - `Requester`

   Por exemplo, para visualizar detalhes de todas as solicitações de alteração concluídas com êxito nas últimas 24 horas, faça o seguinte:

   1. Para Create date range (Criar intervalo de datas), escolha1d.
   2. Na caixa de pesquisa, selecione Status, CompletedWithSuccess (Status, concluído com êxito).
   3. Nos resultados, escolha o nome da solicitação de alteração concluída com êxito para revisar os resultados.
4. Visualizar informações sobre a solicitação de alteração nas seguintes guias:

   - Detalhes do pedido: visualize os detalhes básicos sobre a solicitação de alteração, incluindo o solicitante, o modelo de alteração e os runbooks do Automation selecionados para a alteração. Você também pode seguir um link para os detalhes da operação do Automation e exibir informações sobre quaisquer parâmetros do runbook especificados na solicitação, alarmes do Amazon CloudWatch atribuídos à solicitação de alteração e aprovações e comentários fornecidos para a solicitação.

- Tarefa: exiba informações sobre a tarefa na alteração, incluindo o status da tarefa para solicitações de alteração concluídas, os recursos de destino, as etapas nos runbooks do Automation associados e detalhes do limite de erros e simultaneidade.
- Cronograma: exiba um resumo de todos os eventos associados à solicitação de alteração, listados por data e hora. O resumo indica quando a solicitação de alteração foi criada, ações dos aprovadores atribuídos, uma observação de quando as solicitações de alteração aprovadas são programadas para execução, detalhes do fluxo de trabalho do runbook e alterações de status para o processo de alteração geral e de cada etapa do runbook.
- Associated events: exibe detalhes auditáveis sobre solicitações de alteração registradas no AWS CloudTrail Lake. Os detalhes incluem quais ações da API foram executadas, os parâmetros de solicitação incluídos para essas ações, a conta do usuário que executou a ação, os recursos atualizados durante o processo e outros.

  Quando você habilita o rastreamento de eventos do CloudTrail Lake, o CloudTrail Lake cria um armazenamento de dados de eventos para eventos relacionados às suas solicitações de alteração. Os detalhes de eventos estão disponíveis para a conta ou organização em que a solicitação de alteração foi executada. É possível ativar o rastreamento de eventos do CloudTrail Lake a partir de qualquer solicitação de alteração em sua conta ou organização. Para obter informações sobre como habilitar a integração com o CloudTrail Lake e criar um armazenamento de dados de eventos, consulte Monitoramento dos seus eventos de solicitação de alteração.

  > **Note**
  >
  > Há uma cobrança para o uso do CloudTrail Lake. Para obter detalhes, consulte Definição de preço do AWS CloudTrail.

Visualizar contagens agregadas de solicitações de alteração (linha de comando)

É possível visualizar contagens agregadas de solicitações de alteração no Change Manager, uma ferramenta do AWS Systems Manager, usando a operação da API GetOpsSummary. Esta operação de API pode retornar contagens para uma única Conta da AWS em uma única Região da AWS ou para várias contas e regiões.

> **Note**
>
> Se você quiser exibir contagens agregadas de solicitações de alteração para várias Contas da AWS e várias Regiões da AWS, defina e configure uma sincronização de dados dos recursos. Para obter mais informações, consulte Creating a resource data sync for Inventory.

O procedimento a seguir descreve como usar o AWS Command Line Interface (AWS CLI) (no Linux, macOS ou Windows) para visualizar contagens agregadas de solicitações de alteração.

Para visualizar as contagens agregadas das solicitações de alteração

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.
2. Execute um dos seguintes comandos:

   Uma única conta e região

   Este comando retorna uma contagem de todas as solicitações de alteração para a Conta da AWS e Região da AWS, para as quais a AWS CLI está configurada.

   Linux & macOS

   ```
   aws ssm get-ops-summary \
   --filters Key=AWS:OpsItem.OpsItemType,Values="/aws/changerequests",Type=Equal \
   --aggregators AggregatorType=count,AttributeName=Status,TypeName=AWS:OpsItem
   ```

   Windows

   ```
   aws ssm get-ops-summary ^
   --filters Key=AWS:OpsItem.OpsItemType,Values="/aws/changerequests",Type=Equal ^
   --aggregators AggregatorType=count,AttributeName=Status,TypeName=AWS:OpsItem
   ```

   O sistema retorna informações como as seguintes:

   ```
   {
       "Entities": [
           {
   ```

```
            "Data": {
                "AWS:OpsItem": {
                    "Content": [
                        {
                            "Count": "38",
                            "Status": "Open"
                        }
                    ]
                }
            }
        }
    ]
}
```

Várias contas e/ou regiões

Este comando retorna uma contagem de todas as solicitações de alteração para a Contas da AWS e Regiões da AWS especificada na sincronização de dados do recurso.

Linux & macOS

```
aws ssm get-ops-summary \
    --sync-name resource_data_sync_name \
    --filters Key=AWS:OpsItem.OpsItemType,Values="/aws/
changerequests",Type=Equal \
    --aggregators AggregatorType=count,AttributeName=Status,TypeName=AWS:OpsItem
```

Windows

```
aws ssm get-ops-summary ^
    --sync-name resource_data_sync_name ^
    --filters Key=AWS:OpsItem.OpsItemType,Values="/aws/
changerequests",Type=Equal ^
    --aggregators AggregatorType=count,AttributeName=Status,TypeName=AWS:OpsItem
```

O sistema retorna informações como as seguintes:

```
{
    "Entities": [
        {
            "Data": {
```

```
                "AWS:OpsItem": {
                    "Content": [
                        {
                            "Count": "43",
                            "Status": "Open"
                        },
                        {
                            "Count": "2",
                            "Status": "Resolved"
                        }
                    ]
                }
            }
        }
    ]
}
```

Várias contas e uma região específica

Este comando retorna uma contagem de todas as solicitações de alteração para as Contas da AWS especificadas na sincronização de dados do recurso. No entanto, ele retorna apenas dados da região especificada no comando.

Linux & macOS

```
aws ssm get-ops-summary \
    --sync-name resource_data_sync_name \
    --filters Key=AWS:OpsItem.SourceRegion,Values='Region',Type=Equal
 Key=AWS:OpsItem.OpsItemType,Values="/aws/changerequests",Type=Equal \
    --aggregators AggregatorType=count,AttributeName=Status,TypeName=AWS:OpsItem
```

Windows

```
aws ssm get-ops-summary ^
    --sync-name resource_data_sync_name ^
    --filters Key=AWS:OpsItem.SourceRegion,Values='Region',Type=Equal
 Key=AWS:OpsItem.OpsItemType,Values="/aws/changerequests",Type=Equal ^
    --aggregators AggregatorType=count,AttributeName=Status,TypeName=AWS:OpsItem
```

Várias contas e regiões com saída agrupada por região

Este comando retorna uma contagem de todas as solicitações de alteração para a Contas da AWS e Regiões da AWS especificada na sincronização de dados do recurso. A saída exibe informações de contagem por região.

Linux & macOS

```
aws ssm get-ops-summary \
    --sync-name resource_data_sync_name \
    --filters Key=AWS:OpsItem.OpsItemType,Values="/aws/
changerequests",Type=Equal \
    --aggregators
 '[{"AggregatorType":"count","TypeName":"AWS:OpsItem","AttributeName":"Status","Aggregat
[{"AggregatorType":"count","TypeName":"AWS:OpsItem","AttributeName":"SourceRegion"}]}]'
```

Windows

```
aws ssm get-ops-summary ^
    --sync-name resource_data_sync_name ^
    --filters Key=AWS:OpsItem.OpsItemType,Values="/aws/
changerequests",Type=Equal ^
    --aggregators
 '[{"AggregatorType":"count","TypeName":"AWS:OpsItem","AttributeName":"Status","Aggregat
[{"AggregatorType":"count","TypeName":"AWS:OpsItem","AttributeName":"SourceRegion"}]}]'
```

O sistema retorna informações como as seguintes:

```
{
        "Entities": [
            {
                "Data": {
                    "AWS:OpsItem": {
                        "Content": [
                            {
                                "Count": "38",
                                "SourceRegion": "us-east-1",
                                "Status": "Open"
                            },
                            {
                                "Count": "4",
                                "SourceRegion": "us-east-2",
```

```
                            "Status": "Open"
                        },
                        {
                            "Count": "1",
                            "SourceRegion": "us-west-1",
                            "Status": "Open"
                        },
                        {
                            "Count": "2",
                            "SourceRegion": "us-east-2",
                            "Status": "Resolved"
                        }
                    ]
                }
            }
        }
    ]
}
```

Várias contas e regiões com saída agrupada por contas e regiões

Este comando retorna uma contagem de todas as solicitações de alteração para a Contas da AWS e Regiões da AWS especificada na sincronização de dados do recurso. A saída agrupa as informações de contagem por contas e regiões.

Linux & macOS

```
aws ssm get-ops-summary \
    --sync-name resource_data_sync_name \
    --filters Key=AWS:OpsItem.OpsItemType,Values="/aws/
changerequests",Type=Equal \
    --aggregators
 '[{"AggregatorType":"count","TypeName":"AWS:OpsItem","AttributeName":"Status","Aggregat
[{"AggregatorType":"count","TypeName":"AWS:OpsItem","AttributeName":"SourceAccountId","A
[{"AggregatorType":"count","TypeName":"AWS:OpsItem","AttributeName":"SourceRegion"}]}]}]
```

Windows

```
aws ssm get-ops-summary ^
    --sync-name resource_data_sync_name ^
    --filters Key=AWS:OpsItem.OpsItemType,Values="/aws/
changerequests",Type=Equal ^
```

```
    --aggregators
 '[{"AggregatorType":"count","TypeName":"AWS:OpsItem","AttributeName":"Status","Aggregat
[{"AggregatorType":"count","TypeName":"AWS:OpsItem","AttributeName":"SourceAccountId","A
[{"AggregatorType":"count","TypeName":"AWS:OpsItem","AttributeName":"SourceRegion"}]]}]}]
```

O sistema retorna informações como as seguintes:

```
{
    "Entities": [
        {
            "Data": {
                "AWS:OpsItem": {
                    "Content": [
                        {
                            "Count": "38",
                            "SourceAccountId": "123456789012",
                            "SourceRegion": "us-east-1",
                            "Status": "Open"
                        },
                        {
                            "Count": "4",
                            "SourceAccountId": "111122223333",
                            "SourceRegion": "us-east-2",
                            "Status": "Open"
                        },
                        {
                            "Count": "1",
                            "SourceAccountId": "111122223333",
                            "SourceRegion": "us-west-1",
                            "Status": "Open"
                        },
                        {
                            "Count": "2",
                            "SourceAccountId": "444455556666",
                            "SourceRegion": "us-east-2",
                            "Status": "Resolved"
                        },
                        {
                            "Count": "1",
                            "SourceAccountId": "222222222222",
                            "SourceRegion": "us-east-1",
                            "Status": "Open"
```

```
                        }
                    ]
                }
            }
        }
    ]
}
```

Auditar e registrar atividades do Change Manager em log

Você pode auditar atividades no Change Manager, uma ferramenta do AWS Systems Manager, usando o Amazon CloudWatch e alarmes do AWS CloudTrail.

Para obter mais informações sobre opções de auditoria e registro para o Systems Manager, consulte Registrar em log e monitorar no AWS Systems Manager.

Auditore a atividade do Change Manager usando alarmes do CloudWatch

Você pode configurar e atribuir um alarme do CloudWatch a um modelo de alteração. Se quaisquer condições definidas no alarme forem atendidas, as ações especificadas para o alarme serão realizadas. Na configuração do alarme, você pode especificar um tópico do Amazon Simple Notification Service (Amazon SNS) para notificar quando uma condição de alarme for atendida.

Para obter mais informações sobre como criar um modelo do Change Manager, consulte Trabalhar com modelos de alteração.

Para obter mais informações sobre como criar alarmes do CloudWatch, consulte Usar alarmes do CloudWatch no Guia do usuário do Amazon CloudWatch.

Auditore a atividade do Change Manager usando o CloudTrail

O CloudTrail captura chamadas à API feitas no console do Systems Manager, na AWS Command Line Interface (AWS CLI) e no SDK do Systems Manager. Você pode visualizar as informações no console do CloudTrail ou em um bucket do Amazon Simple Storage Service (Amazon S3), onde elas estão armazenadas. Um bucket é usado em todos os logs do CloudTrail para sua conta.

Os logs de ações do Change Manager mostram a criação do documento de modelo de alteração, as aprovações e rejeições da solicitação de alteração e o modelo de alteração, atividade gerada por runbooks do Automation e muito mais. Para obter mais informações sobre como visualizar e utilizar os logs do CloudTrail de atividades do Systems Manager, consulte Registrar em log chamadas de API do AWS Systems Manager com o AWS CloudTrail.

Solução de problemas do Change Manager

Use as informações a seguir para obter ajuda para solucionar problemas com o Change Manager, uma ferramenta do AWS Systems Manager.

Tópicos

- Erro “Group {GUID} not found” (Grupo {GUID} não encontrado) durante aprovações de solicitação ao usar o Active Directory.

Erro “Group ***{GUID}*** not found” (Grupo {GUID} não encontrado) durante aprovações de solicitação ao usar o Active Directory.

Problema: quando o AWS IAM Identity Center (IAM Identity Center) é usado para o gerenciamento de identidade de usuários, um membro de um grupo do Active Directory que recebe permissões de aprovação no Change Manager recebe um erro “não autorizado” ou “grupo não encontrado”.

- Solução: quando você seleciona grupos do Active Directory no IAM Identity Center para acesso ao AWS Management Console, o sistema agenda uma sincronização periódica que copia informações desses grupos do Active Directory no IAM Identity Center. Esse processo deve ser concluído antes que os usuários autorizados por meio da associação de grupo do Active Directory possam aprovar com êxito uma solicitação. Para obter mais informações, consulte Conectar ao diretório do Microsoft AD no Guia do usuário do AWS IAM Identity Center.

Documentos do AWS Systems Manager

Um documento do AWS Systems Manager (documento SSM) define as ações que o Systems Manager realiza nas suas instâncias gerenciadas. O Systems Manager inclui mais de 100 documentos pré-configurados, que você pode usar especificando parâmetros no runtime. Documentos pré-configurados podem ser encontrados no console de documentos do Systems Manager escolhendo a guia Owned by Amazon (Propriedade da Amazon) ou especificando a Amazon para o filtro `Owner` ao chamar a operação da API `ListDocuments`. Os documentos usam JSON (JavaScript Object Notation) ou YAML e incluem etapas e parâmetros especificados por você.

Para maior segurança, desde 14 de julho de 2025, os documentos do SSM são compatíveis com a interpolação de variáveis de ambiente ao processar parâmetros. Esse recurso, disponível na versão 2.2 do esquema e com a versão 3.3.2746.0 ou superior do SSM Agent, ajuda a evitar ataques de injeção de comando.

Para começar a usar o , abra o Systems Manager console (Console do gerenciador de sistemas). No painel de navegação, escolha Documents.

> **Important**
>
> No Systems Manager, um documento SSM pertencente à Amazon é um documento criado e gerenciado pela própria Amazon Web Services. Os documentos pertencentes à Amazon incluem um prefixo como `AWS-*` no nome do documento. O proprietário do documento é considerado a Amazon, e não uma conta de usuário específica em AWS. Esses documentos estão disponíveis publicamente para uso de todos.

Como a ferramenta Documentos pode beneficiar minha organização?

O Documentos, uma ferramenta do AWS Systems Manager, oferece os seguintes benefícios:

- Categorias de documentos

  Para ajudar a encontrar os documentos necessários, escolha uma categoria dependendo do tipo de documento que você está procurando. Para ampliar sua pesquisa, você pode escolher várias categorias do mesmo tipo de documento. Não há suporte para a escolha de categorias de diferentes tipos de documentos. As categorias só são compatíveis com documentos de propriedade da Amazon.
- Versões de documentos

  Você pode criar e salvar diferentes versões de documentos. Em seguida, pode especificar uma versão padrão para cada documento. A versão padrão de um documento pode ser atualizada para uma versão mais recente ou revertida para uma mais antiga. Quando você altera o conteúdo de um documento, o Systems Manager incrementa automaticamente a versão do documento. Você pode recuperar ou usar qualquer versão de um documento especificando a versão do documento no console, comandos da AWS Command Line Interface (AWS CLI) ou chamadas de API.
- Personalize documentos para suas necessidades

  Se quiser personalizar as etapas e ações em um documento, você pode criar seu próprio. O sistema armazena o documento com sua Conta da AWS na Região da AWS em que você o cria. Para obter mais informações sobre como criar um documento do SSM, consulte Criar conteúdo de documento do SSM.
- Tags de documentos

Você pode atribuir uma tag aos seus documentos para ajudar a identificar rapidamente um ou mais documentos de acordo com as tags que tiver atribuído a eles. Por exemplo, você pode marcar documentos para ambientes, departamentos, usuários, grupos ou períodos específicos. Você pode também restringir o acesso aos documentos criando uma política do AWS Identity and Access Management (IAM) que especifique as tags que um usuário ou grupo pode acessar.

- Compartilhar documentos

  Você pode tornar seus documentos públicos ou compartilhá-los com contas da Contas da AWS específicas na mesma Região da AWS. Compartilhar documentos entre contas pode ser útil se, por exemplo, você desejar que todas as instâncias do Amazon Elastic Compute Cloud (Amazon EC2) fornecidas aos seus clientes ou funcionários tenham a mesma configuração. Além de manter atualizadas as aplicações ou patches nessas instâncias, talvez você queira restringir algumas atividades nas instâncias dos clientes. Ou talvez você queira garantir que as instâncias usadas por contas de funcionário na organização recebam acesso a recursos internos específicos. Para obter mais informações, consulte Compartilhar documentos do Systems Manager.

Quem deve usar documentos?

- Qualquer cliente da AWS que deseje usar as ferramentas do Systems Manager para melhorar sua eficiência operacional em escala, reduzir erros associados à intervenção manual e reduzir o tempo de resolução de problemas comuns.
- Especialistas em infraestrutura que desejem automatizar tarefas de implantação e configuração.
- Administradores que desejem resolver problemas comuns de forma confiável, melhorar a eficiência da solução de problemas e reduzir operações repetitivas.
- Usuários que desejarem automatizar uma tarefa que normalmente executam manualmente.

Quais são os tipos de documentos de SSM?

A tabela a seguir descreve os diferentes tipos de documentos do SSM e seus usos.

Tipo	Use com	Detalhes
ApplicationConfiguration	AWS AppConfig	O AWS AppConfig, uma ferramenta do AWS Systems Manager, permite criar,

Tipo	Use com	Detalhes
ApplicationConfigurationSchema		gerenciar e implantar rapidamente configurações de aplicações. Você pode armazenar dados de configuração em um documento SSM criando um documento que usa o tipo de documento `ApplicationConfiguration` . Para obter mais informações, consulte Configurações de forma livre no Guia do usuário do AWS AppConfig. Se você criar uma configuração em um documento do SSM, deverá especificar um esquema JSON correspondente. O esquema usa o tipo de documento `ApplicationConfigurationSchema` e, como um conjunto de regras, define as propriedades permitidas para cada definição de configuração de aplicação . Para obter mais informações, consulte About validators (Sobre validadores) no AWS AppConfig User Guide. (Guia do usuário do).

Tipo	Use com	Detalhes
Runbook de automação	Automação State Manager Maintenance Windows	Use runbooks do Automation ao realizar tarefas comuns de manutenção e implantação, como criar ou atualizar uma Amazon Machine Image (AMI). O State Manager usa runbooks do Automation para aplicar uma configuração. Essas ações podem ser executadas em um ou mais destinos em qualquer ponto durante o ciclo de vida de uma instância. O Maintenance Windows usa ao realizar tarefas comuns de manutenção e implantação com base na programação especificada. Todos os runbooks de automação compatíveis com sistemas operacionais baseados em Linux também são compatíveis com as instâncias do EC2 para macOS.

Tipo	Use com	Detalhes
Alterar documento do calendário	Change Calendar	O Change Calendar, uma ferramenta do AWS Systems Manager, usa o tipo de documento `ChangeCalendar` . Um documento do Change Calendar armazena uma entrada de calendário e eventos associados que podem permitir ou impedir que as ações de automação alterem seu ambiente. No Change Calendar, um documento armazena dados do iCalendar 2.0 em formato de texto simples. O Change Calendar não tem suporte em instâncias EC2 para macOS.

Tipo	Use com	Detalhes
Modelo AWS CloudFormation	AWS CloudFormation	Os modelos do AWS CloudFormation descrevem os recursos que você deseja provisionar nas suas pilhas do CloudFormation. Com o armazenamento de modelos do CloudFormation como documentos do Systems Manager, você pode se beneficiar dos recursos de documento do Systems Manager Isso inclui criar e comparar várias versões do seu modelo e compartilhar seu modelo com outras contas na mesma Região da AWS. Você pode criar e editar modelos e pilhas do CloudFormation usando o Application Manager, uma ferramenta do Systems Manager. Para obter mais informações, consulte Trabalhar com modelos e pilhas do AWS CloudFormation no Application Manager..

Tipo	Use com	Detalhes
Documento de comando	Run Command State Manager Maintenance Windows	O Run Command, uma ferramenta do AWS Systems Manager, usa documentos do Command para executar comandos. O State Manager, uma ferramenta do AWS Systems Manager usa documentos de comando para aplicar uma configuração. Essas ações podem ser executadas em um ou mais destinos em qualquer ponto durante o ciclo de vida de uma instância. O Maintenance Windows, uma ferramenta do AWS Systems Manager, usa documentos de comando para aplicar uma configuração de acordo com a programação especificada. A maioria dos documentos de comando é aceita em todos os sistemas operacionais Linux e Windows Server aos quais o Systems Manager oferece suporte. Os seguintes documentos de comandos são aceitos em instâncias do EC2 para macOS: • `AWS-ConfigureAWSPackage` • `AWS-RunPatchBaseline`

Tipo	Use com	Detalhes
		• `AWS-RunPatchBaselineAssociation` • `AWS-RunShellScript`
Modelo de pacote de conformidade do AWS Config	AWS Config	Os modelos de pacote de conformidade do AWS Config são documentos formatados em YAML usados para criar pacotes de conformidade que contêm a lista de regras gerenciadas ou personalizadas e ações de remediação do AWS Config. Para obter mais informações, consulte Pacotes de conformidade.
Documento de pacote	Distributor	No Distributor, uma ferramenta do AWS Systems Manager, um pacote é representado por um documento do SSM. Um documento de pacote inclui arquivos ZIP anexados que contêm software ou ativos para instalar nas instâncias gerenciadas. Criar um pacote no Distributor cria o documento de pacote. O Distributor não tem suporte em instâncias gerenciadas no Oracle Linux e macOS

Tipo	Use com	Detalhes
Documento de política	State Manager	O Inventory, uma ferramenta do AWS Systems Manager, usa o documento de política `AWS-GatherSoftwareInventory` com uma associação ao State Manager para coletar dados de inventário de instâncias gerenciadas. Ao criar seus próprios documentos do SSM, os runbooks do Automation e os documentos de comando são o método preferido para aplicar uma política em uma instância gerenciada. O Systems Manager Inventory e o documento de políticas `AWS-GatherSoftwareInventory` são compatíveis com todos os sistemas operacionais suportados pelo Systems Manager.

Tipo	Use com	Detalhes
Modelo de análise pós-incidente	Análise pós-incidente do Incident Manager	O Incident Manager usa o modelo de análise pós-incidente para criar uma análise baseada nas práticas recomendadas para o gerenciamento de operações da AWS. Use o modelo para criar uma análise que sua equipe possa usar para identificar melhorias na sua resposta a incidentes.

Tipo	Use com	Detalhes
Documento de sessão	Session Manager	O Session Manager, uma ferramenta do AWS Systems Manager, usa documentos de sessão para determinar qual tipo de sessão iniciar, como uma sessão de encaminha mento de portas, uma sessão para executar um comando interativo ou uma sessão para criar um túnel SSH. Documentos de sessão são aceitos em todos os sistemas operacionais Linux e Windows Server aos quais o Systems Manager oferece suporte. Os seguintes documentos Command são suportados em instâncias do EC2 paramacOS : • `AWS-PasswordReset` • `AWS-StartInteracti veCommand` • `AWS-StartPortForwa rdingSession` • `AWS-StartPortForwa rdingSessionToSock et` • `AWS-StartSSHSession`

Cotas de documentos do SSM

Para obter mais informações sobre cotas de documentos do SSM, consulte Systems Manager service quotas no Referência geral da Amazon Web Services.

Tópicos

- Componentes do documento
- Criar conteúdo de documento do SSM
- Trabalhar com documentos
- Solução de problemas de tratamento de parâmetros

Componentes do documento

Esta seção contém informações sobre os componentes presentes nos documentos do SSM.

Conteúdo

- Esquemas, atributos e exemplos
- Elementos e parâmetros de dados
- Referência de plug-ins de documentos de comando

Esquemas, atributos e exemplos

Os documentos do AWS Systems Manager (SSM) usam as versões de esquema a seguir.

- Os documentos do tipo `Command` podem usar o esquema versão 1.2, 2.0 e 2.2. Se você usa documentos do esquema 1.2, recomendamos criar documentos que utilizem o esquema versão 2.2.
- Os documentos do tipo `Policy` devem usar o esquema versão 2.0 ou posterior.
- Os documentos do tipo `Automation` devem usar o esquema versão 0.3.
- Os documentos do tipo `Session` devem usar o esquema versão 1.0.
- Você pode criar documentos em JSON ou YAML.

Para obter mais informações sobre o esquema de documentos do `Session`, consulte Esquema do documento de sessão.

Ao usar a versão de esquema mais recentes para documentos `Command` e `Policy`, você poderá aproveitar os seguintes recursos.

Recursos de documentos da versão 2.2 do esquema

Recurso	Detalhes
Edição de documentos	Agora, documentos podem ser atualizados. Com a versão 1.2, qualquer atualização de um documento exigia que você o salvasse com um nome diferente.
Versionamento automático	Qualquer atualização de um documento cria uma nova versão. Esta não é uma versão de esquema, mas uma versão do documento.
Versão padrão	Se você possui várias versões de um documento, pode especificar qual delas é o documento padrão.
Sequenciamento	Os plugins ou as etapas em um documento são executados na ordem que você especificou.
Suporte entre plataformas	O suporte entre plataformas permite que você especifique diferentes sistemas operacion ais para diferentes plugins dentro do mesmo documento do SSM. O suporte entre plataform as usa o parâmetro `precondition` dentro de uma etapa.
Interpolação de parâmetros	Permite que parâmetros de string sejam interpolados em variáveis de ambiente antes da execução do comando, fornecendo melhor segurança contra injeções de comando. Quando definido como `ENV_VAR`, o agente cria uma variável de ambiente denominada `SSM_`*`parameter-name`* que contém o valor do parâmetro.

> **Note**
>
> Você deve manter o AWS Systems Manager SSM Agent atualizado em suas instâncias com a versão mais recente para usar os novos recursos do Systems Manager e os recursos de documento do SSM. Para obter mais informações, consulte Atualização do SSM Agent por meio de Run Command.

A tabela a seguir lista as diferenças entre as principais versões de esquema.

Versão 1.2	Versão 2.2 (versão mais recente)	Detalhes
runtimeConfig	mainSteps	Na versão 2.2, a seção `mainSteps` substitui `runtimeConfig`. O `mainSteps` A seção permite que o Systems Manager execute etapas em sequência.
propriedades	inputs	Na versão 2.2, a seção `inputs` substitui a seção `properties`. A seção `inputs` aceita parâmetros para as etapas.
Comandos da do	runCommand	Na versão 2.2, a seção `inputs` usa o parâmetro `runCommand` em vez do parâmetro `commands`.
id	action	Na versão 2.2, `Action` substitui `ID`. Esta é apenas uma mudança de nome.

Versão 1.2	Versão 2.2 (versão mais recente)	Detalhes
não aplicável	nome	Na versão 2.2, `name` é um nome definido pelo usuário para uma etapa.

Usar o parâmetro precondition

Com o esquema versão 2.2 ou superior, você pode usar o parâmetro `precondition` para especificar o sistema operacional de destino de cada plugin ou para validar parâmetros de entrada que você definiu em seu documento do SSM. O`precondition`suporta referenciar os parâmetros de entrada do documento SSM, e`platformType`usando valores de`Linux`,`MacOS`, e`Windows`. Somente a`StringEquals`é suportado.

Para documentos que utilizam o esquema versão 2.2 ou posterior, se a `precondition` não for especificada, cada plugin será executado ou ignorado com base na compatibilidade do plugin com o sistema operacional. Compatibilidade de plugins com o sistema operacional é avaliada antes do`precondition`. Para documentos que utilizam o esquema 2.0 ou anterior, os plugins incompatíveis geram um erro.

Por exemplo, em um documento com a versão 2.2 do esquema, se `precondition` não for especificado e o plugin `aws:runShellScript` estiver relacionado, a etapa será executada em instâncias do Linux, mas será ignorada pelo sistema em instâncias do Windows Server porque o `aws:runShellScript` não é compatível com instâncias do Windows Server. No entanto, para um documento de esquema versão 2.0, se você especificar o plugin `aws:runShellScript` e depois executar o documento em instâncias do Windows Server, a execução falhará. Você poderá ver um exemplo do parâmetro de pré-condição em um documento do SSM ainda nesta seção.

Versão 2.2 do esquema

Elementos de nível superior

O exemplo a seguir mostra os elementos de nível superior de um documento do SSM usando o esquema versão 2.2.

YAML

```
---
```

```
schemaVersion: "2.2"
description: A description of the document.
parameters:
  parameter 1:
    property 1: "value"
    property 2: "value"
  parameter 2:
    property 1: "value"
    property 2: "value"
mainSteps:
  - action: Plugin name
    name: A name for the step.
    inputs:
      input 1: "value"
      input 2: "value"
      input 3: "{{ parameter 1 }}"
```

JSON

```
{
   "schemaVersion": "2.2",
   "description": "A description of the document.",
   "parameters": {
       "parameter 1": {
          "property 1": "value",
          "property 2": "value"
       },
       "parameter 2":{
          "property 1": "value",
          "property 2": "value"
       }
    },
   "mainSteps": [
      {
         "action": "Plugin name",
         "name": "A name for the step.",
         "inputs": {
            "input 1": "value",
            "input 2": "value",
            "input 3": "{{ parameter 1 }}"
         }
      }
   ]
```

```
}
```

Exemplo de esquema versão 2.2 do

Veja como o exemplo a seguir usa o plugin `aws:runPowerShellScript` para executar um comando do PowerShell nas instâncias de destino.

YAML

```
---
schemaVersion: "2.2"
description: "Example document"
parameters:
  Message:
    type: "String"
    description: "Example parameter"
    default: "Hello World"
    allowedValues:
    - "Hello World"
mainSteps:
  - action: "aws:runPowerShellScript"
    name: "example"
    inputs:
      timeoutSeconds: '60'
      runCommand:
      - "Write-Output {{Message}}"
```

JSON

```
{
   "schemaVersion": "2.2",
   "description": "Example document",
   "parameters": {
      "Message": {
         "type": "String",
         "description": "Example parameter",
         "default": "Hello World",
         "allowedValues": ["Hello World"]
      }
   },
   "mainSteps": [
      {
```

```
         "action": "aws:runPowerShellScript",
         "name": "example",
         "inputs": {
            "timeoutSeconds": "60",
            "runCommand": [
               "Write-Output {{Message}}"
            ]
         }
      }
   ]
}
```

Exemplos do parâmetro precondition da versão 2.2 do esquema

O esquema versão 2.2 fornece suporte para várias plataformas. Isso significa que, dentro de um único documento do SSM, você pode especificar diferentes sistemas operacionais para diferentes plugins. O suporte entre plataformas usa o parâmetro `precondition` dentro de uma etapa, como mostra o exemplo a seguir. Você também pode usar o`precondition`para validar os parâmetros de entrada definidos no documento do SSM. Veja isso no segundo exemplo a seguir.

YAML

```
---
schemaVersion: '2.2'
description: cross-platform sample
mainSteps:
- action: aws:runPowerShellScript
  name: PatchWindows
  precondition:
    StringEquals:
    - platformType
    - Windows
  inputs:
    runCommand:
    - cmds
- action: aws:runShellScript
  name: PatchLinux
  precondition:
    StringEquals:
    - platformType
    - Linux
  inputs:
```

```
    runCommand:
    - cmds
```

JSON

```
{
   "schemaVersion": "2.2",
   "description": "cross-platform sample",
   "mainSteps": [
      {
         "action": "aws:runPowerShellScript",
         "name": "PatchWindows",
         "precondition": {
            "StringEquals": [
               "platformType",
               "Windows"
            ]
         },
         "inputs": {
            "runCommand": [
               "cmds"
            ]
         }
      },
      {
         "action": "aws:runShellScript",
         "name": "PatchLinux",
         "precondition": {
            "StringEquals": [
               "platformType",
               "Linux"
            ]
         },
         "inputs": {
            "runCommand": [
               "cmds"
            ]
         }
      }
   ]
}
```

YAML

```
---
schemaVersion: '2.2'
parameters:
  action:
    type: String
    allowedValues:
    - Install
    - Uninstall
  confirmed:
    type: String
    allowedValues:
    - True
    - False
mainSteps:
- action: aws:runShellScript
  name: InstallAwsCLI
  precondition:
    StringEquals:
    - "{{ action }}"
    - "Install"
  inputs:
    runCommand:
    - sudo apt install aws-cli
- action: aws:runShellScript
  name: UninstallAwsCLI
  precondition:
    StringEquals:
    - "{{ action }} {{ confirmed }}"
    - "Uninstall True"
  inputs:
    runCommand:
    - sudo apt remove aws-cli
```

JSON

```
{
   "schemaVersion": "2.2",
   "parameters": {
      "action": {
         "type": "String",
         "allowedValues": [
```

```
            "Install",
            "Uninstall"
         ]
      },
      "confirmed": {
         "type": "String",
         "allowedValues": [
            true,
            false
         ]
      }
   },
   "mainSteps": [
      {
         "action": "aws:runShellScript",
         "name": "InstallAwsCLI",
         "precondition": {
            "StringEquals": [
               "{{ action }}",
               "Install"
            ]
         },
         "inputs": {
            "runCommand": [
               "sudo apt install aws-cli"
            ]
         }
      },
      {
         "action": "aws:runShellScript",
         "name": "UninstallAwsCLI",
         "precondition": {
            "StringEquals": [
               "{{ action }} {{ confirmed }}",
               "Uninstall True"
            ]
         },
         "inputs": {
            "runCommand": [
               "sudo apt remove aws-cli"
            ]
         }
      }
   ]
```

```
}
```

Exemplo de interpolação do esquema versão 2.2 com versões do SSM Agent anteriores à 3.3.2746.0

Nas versões do SSM Agent anteriores à 3.3.2746.0, o agente ignora o parâmetro `interpolationType` e, em vez disso, executa uma substituição de string bruta. Se estiver referenciando `SSM_`*`parameter-name`* de forma explícita, você deve definir explicitamente. No exemplo a seguir para Linux, a variável de ambiente `SSM_Message` é referenciada explicitamente.

```
{
    "schemaVersion": "2.2",
    "description": "An example document",
    "parameters": {
        "Message": {
            "type": "String",
            "description": "Message to be printed",
            "default": "Hello",
            "interpolationType" : "ENV_VAR",
      "allowedPattern: "^[^"]*$"

        }
    },
    "mainSteps": [{
        "action": "aws:runShellScript",
        "name": "printMessage",
        "inputs": {
            "runCommand": [
              "if [ -z "${SSM_Message+x}" ]; then",
              "    export SSM_Message=\"{{Message}}\"",
              "fi",
              "",
              "echo $SSM_Message"
            ]
        }
    }
}
```

> **Note**
>
> `allowedPattern` não será tecnicamente necessário se um documento do SSM não usar chaves duplas: `{{ }}`

Exemplo de esquema versão 2.2 do State Manager

Você pode usar o seguinte documento do SSM com o State Manager, uma ferramenta do Systems Manager, para baixar e instalar o software antivírus ClamAV. O State Manager impõe uma configuração específica, o que significa que cada vez que a associação do State Manager for executada, o sistema verificará se o software ClamAV está instalado. Se não, o State Manager executa novamente esse documento.

YAML

```
---
schemaVersion: '2.2'
description: State Manager Bootstrap Example
parameters: {}
mainSteps:
- action: aws:runShellScript
  name: configureServer
  inputs:
    runCommand:
    - sudo yum install -y httpd24
    - sudo yum --enablerepo=epel install -y clamav
```

JSON

```
{
   "schemaVersion": "2.2",
   "description": "State Manager Bootstrap Example",
   "parameters": {},
   "mainSteps": [
      {
         "action": "aws:runShellScript",
         "name": "configureServer",
         "inputs": {
            "runCommand": [
               "sudo yum install -y httpd24",
```

```
                "sudo yum --enablerepo=epel install -y clamav"
            ]
          }
       }
    ]
}
```

Exemplo de inventário do esquema versão 2.2

Você pode usar o documento SSM a seguir com o State Manager para coletar metadados de inventário sobre suas instâncias.

YAML

```
---
schemaVersion: '2.2'
description: Software Inventory Policy Document.
parameters:
  applications:
    type: String
    default: Enabled
    description: "(Optional) Collect data for installed applications."
    allowedValues:
    - Enabled
    - Disabled
  awsComponents:
    type: String
    default: Enabled
    description: "(Optional) Collect data for AWS Components like amazon-ssm-agent."
    allowedValues:
    - Enabled
    - Disabled
  networkConfig:
    type: String
    default: Enabled
    description: "(Optional) Collect data for Network configurations."
    allowedValues:
    - Enabled
    - Disabled
  windowsUpdates:
    type: String
    default: Enabled
```

```
    description: "(Optional) Collect data for all Windows Updates."
    allowedValues:
    - Enabled
    - Disabled
  instanceDetailedInformation:
    type: String
    default: Enabled
    description: "(Optional) Collect additional information about the instance,
 including
      the CPU model, speed, and the number of cores, to name a few."
    allowedValues:
    - Enabled
    - Disabled
  customInventory:
    type: String
    default: Enabled
    description: "(Optional) Collect data for custom inventory."
    allowedValues:
    - Enabled
    - Disabled
mainSteps:
- action: aws:softwareInventory
  name: collectSoftwareInventoryItems
  inputs:
    applications: "{{ applications }}"
    awsComponents: "{{ awsComponents }}"
    networkConfig: "{{ networkConfig }}"
    windowsUpdates: "{{ windowsUpdates }}"
    instanceDetailedInformation: "{{ instanceDetailedInformation }}"
    customInventory: "{{ customInventory }}"
```

JSON

```
{
   "schemaVersion": "2.2",
   "description": "Software Inventory Policy Document.",
   "parameters": {
      "applications": {
         "type": "String",
         "default": "Enabled",
         "description": "(Optional) Collect data for installed applications.",
         "allowedValues": [
            "Enabled",
```

```
            "Disabled"
         ]
      },
      "awsComponents": {
         "type": "String",
         "default": "Enabled",
         "description": "(Optional) Collect data for AWS Components like amazon-ssm-
agent.",
         "allowedValues": [
            "Enabled",
            "Disabled"
         ]
      },
      "networkConfig": {
         "type": "String",
         "default": "Enabled",
         "description": "(Optional) Collect data for Network configurations.",
         "allowedValues": [
            "Enabled",
            "Disabled"
         ]
      },
      "windowsUpdates": {
         "type": "String",
         "default": "Enabled",
         "description": "(Optional) Collect data for all Windows Updates.",
         "allowedValues": [
            "Enabled",
            "Disabled"
         ]
      },
      "instanceDetailedInformation": {
         "type": "String",
         "default": "Enabled",
         "description": "(Optional) Collect additional information about the
 instance, including\nthe CPU model, speed, and the number of cores, to name a
 few.",
         "allowedValues": [
            "Enabled",
            "Disabled"
         ]
      },
      "customInventory": {
         "type": "String",
```

```
            "default": "Enabled",
            "description": "(Optional) Collect data for custom inventory.",
            "allowedValues": [
               "Enabled",
               "Disabled"
            ]
         }
      },
      "mainSteps": [
         {
            "action": "aws:softwareInventory",
            "name": "collectSoftwareInventoryItems",
            "inputs": {
               "applications": "{{ applications }}",
               "awsComponents": "{{ awsComponents }}",
               "networkConfig": "{{ networkConfig }}",
               "windowsUpdates": "{{ windowsUpdates }}",
               "instanceDetailedInformation": "{{ instanceDetailedInformation }}",
               "customInventory": "{{ customInventory }}"
            }
         }
      ]
}
```

Exemplo de esquema versão 2.2 do **`AWS-ConfigureAWSPackage`**

O exemplo a seguir mostra o documento `AWS-ConfigureAWSPackage`. O `mainSteps` Inclui a seção `aws:configurePackage` plugin no `action` Etapa.

> **Note**
>
> Nos sistemas operacionais Linux, somente os pacotes `AmazonCloudWatchAgent` e `AWSSupport-EC2Rescue` são suportados.

YAML

```
---
schemaVersion: '2.2'
description: 'Install or uninstall the latest version or specified version of an AWS
```

```
  package. Available packages include the following: AWSPVDriver,
 AwsEnaNetworkDriver,
  AwsVssComponents, and AmazonCloudWatchAgent, and AWSSupport-EC2Rescue.'
parameters:
  action:
    description: "(Required) Specify whether or not to install or uninstall the
 package."
    type: String
    allowedValues:
    - Install
    - Uninstall
  name:
    description: "(Required) The package to install/uninstall."
    type: String
    allowedPattern: "^arn:[a-z0-9][-.a-z0-9]{0,62}:[a-z0-9][-.a-z0-9]{0,62}:([a-
z0-9][-.a-z0-9]{0,62})?:([a-z0-9][-.a-z0-9]{0,62})?:package\\/[a-zA-Z][a-zA-Z0-9\\-
_]{0,39}$|^[a-zA-Z][a-zA-Z0-9\\-_]{0,39}$"
  version:
    type: String
    description: "(Optional) A specific version of the package to install or
 uninstall."
mainSteps:
- action: aws:configurePackage
  name: configurePackage
  inputs:
    name: "{{ name }}"
    action: "{{ action }}"
    version: "{{ version }}"
```

JSON

```
{
   "schemaVersion": "2.2",
   "description": "Install or uninstall the latest version or specified version
 of an AWS package. Available packages include the following: AWSPVDriver,
 AwsEnaNetworkDriver, AwsVssComponents, and AmazonCloudWatchAgent, and AWSSupport-
EC2Rescue.",
   "parameters": {
      "action": {
         "description":"(Required) Specify whether or not to install or uninstall
 the package.",
         "type":"String",
         "allowedValues":[
```

```
            "Install",
            "Uninstall"
         ]
      },
      "name": {
         "description": "(Required) The package to install/uninstall.",
         "type": "String",
         "allowedPattern": "^arn:[a-z0-9][-.a-z0-9]{0,62}:[a-z0-9][-.a-z0-9]{0,62}:
([a-z0-9][-.a-z0-9]{0,62})?:([a-z0-9][-.a-z0-9]{0,62})?:package\\/[a-zA-Z][a-zA-
Z0-9\\-_]{0,39}$|^[a-zA-Z][a-zA-Z0-9\\-_]{0,39}$"
      },
      "version": {
         "type": "String",
         "description": "(Optional) A specific version of the package to install or
 uninstall."
      }
   },
   "mainSteps":[
      {
         "action": "aws:configurePackage",
         "name": "configurePackage",
         "inputs": {
            "name": "{{ name }}",
            "action": "{{ action }}",
            "version": "{{ version }}"
         }
      }
   ]
}
```

Versão 1.2 do esquema

O exemplo a seguir mostra os elementos de nível superior de um documento do esquema versão 1.2.

```
{
   "schemaVersion":"1.2",
   "description":"A description of the SSM document.",
   "parameters":{
      "parameter 1":{
         "one or more parameter properties"
      },
```

```
      "parameter 2":{
         "one or more parameter properties"
      },
      "parameter 3":{
         "one or more parameter properties"
      }
   },
   "runtimeConfig":{
      "plugin 1":{
         "properties":[
            {
               "one or more plugin properties"
            }
         ]
      }
   }
}
```

Exemplo de esquema versão 1.2 do **`aws:runShellScript`**

O exemplo a seguir mostra o`AWS-RunShellScript`Documento do MUS do. A seção runtimeConfig inclui o plugin `aws:runShellScript`.

```
{
    "schemaVersion":"1.2",
    "description":"Run a shell script or specify the commands to run.",
    "parameters":{
        "commands":{
            "type":"StringList",
            "description":"(Required) Specify a shell script or a command to run.",
            "minItems":1,
            "displayType":"textarea"
        },
        "workingDirectory":{
            "type":"String",
            "default":"",
            "description":"(Optional) The path to the working directory on your
 instance.",
            "maxChars":4096
        },
        "executionTimeout":{
            "type":"String",
            "default":"3600",
```

```
            "description":"(Optional) The time in seconds for a command to complete
 before it is considered to have failed. Default is 3600 (1 hour). Maximum is 172800
 (48 hours).",
            "allowedPattern":"([1-9][0-9]{0,3})|(1[0-9]{1,4})|(2[0-7][0-9]{1,3})|
(28[0-7][0-9]{1,2})|(28800)"
        }
    },
    "runtimeConfig":{
        "aws:runShellScript":{
            "properties":[
                {
                    "id":"0.aws:runShellScript",
                    "runCommand":"{{ commands }}",
                    "workingDirectory":"{{ workingDirectory }}",
                    "timeoutSeconds":"{{ executionTimeout }}"
                }
            ]
        }
    }
}
```

Versão 0.3 do esquema

Elementos de nível superior

O exemplo a seguir mostra os elementos de nível superior de um runbook do Automation versão 0.3 do esquema no formato JSON.

```
{
    "description": "document-description",
    "schemaVersion": "0.3",
    "assumeRole": "{{assumeRole}}",
    "parameters": {
        "parameter1": {
            "type": "String",
            "description": "parameter-1-description",
            "default": ""
        },
        "parameter2": {
            "type": "String",
            "description": "parameter-2-description",
            "default": ""
        }
```

```
    },
    "variables": {
        "variable1": {
            "type": "StringMap",
            "description": "variable-1-description",
            "default": {}
        },
        "variable2": {
            "type": "String",
            "description": "variable-2-description",
            "default": "default-value"
        }
    },
    "mainSteps": [
        {
            "name": "myStepName",
            "action": "action-name",
            "maxAttempts": 1,
            "inputs": {
                "Handler": "python-only-handler-name",
                "Runtime": "runtime-name",
                "Attachment": "script-or-zip-name"
            },
            "outputs": {
                "Name": "output-name",
                "Selector": "selector.value",
                "Type": "data-type"
            }
        }
    ],
    "files": {
        "script-or-zip-name": {
            "checksums": {
                "sha256": "checksum"
            },
            "size": 1234
        }
    }
}
```

Exemplo de runbook de automação YAML

O exemplo a seguir mostra o conteúdo de um runbook do Automation, no formato YAML. Este exemplo funcional da versão 0.3 do esquema do documento também demonstra o uso do Markdown para formatar descrições de documentos.

```
description: >-
  ##Title: LaunchInstanceAndCheckState

  -----

  **Purpose**: This Automation runbook first launches an EC2 instance
  using the AMI ID provided in the parameter ```imageId```. The second step of
  this document continuously checks the instance status check value for the
  launched instance until the status ```ok``` is returned.


  ##Parameters:

  -----

  Name | Type | Description | Default Value

  ------------- | ------------- | ------------- | -------------

  assumeRole | String | (Optional) The ARN of the role that allows Automation to
  perform the actions on your behalf. | -

  imageId  | String | (Optional) The AMI ID to use for launching the instance.
  The default value uses the latest Amazon Linux AMI ID available. | {{
  ssm:/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-gp2 }}
schemaVersion: '0.3'
assumeRole: 'arn:aws:iam::111122223333::role/AutomationServiceRole'
parameters:
  imageId:
    type: String
    default: '{{ ssm:/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-gp2 }}'
    description: >-
      (Optional) The AMI ID to use for launching the instance. The default value
      uses the latest released Amazon Linux AMI ID.
  tagValue:
    type: String
    default: ' LaunchedBySsmAutomation'
    description: >-
      (Optional) The tag value to add to the instance. The default value is
```

```
      LaunchedBySsmAutomation.
  instanceType:
    type: String
    default: t2.micro
    description: >-
      (Optional) The instance type to use for the instance. The default value is
      t2.micro.
mainSteps:
  - name: LaunchEc2Instance
    action: 'aws:executeScript'
    outputs:
      - Name: payload
        Selector: $.Payload
        Type: StringMap
    inputs:
      Runtime: python3.8
      Handler: launch_instance
      Script: ''
      InputPayload:
        image_id: '{{ imageId }}'
        tag_value: '{{ tagValue }}'
        instance_type: '{{ instanceType }}'
      Attachment: launch.py
    description: >-
      **About This Step**


      This step first launches an EC2 instance using the ```aws:executeScript```
      action and the provided python script.
  - name: WaitForInstanceStatusOk
    action: 'aws:executeScript'
    inputs:
      Runtime: python3.8
      Handler: poll_instance
      Script: |-
        def poll_instance(events, context):
          import boto3
          import time

          ec2 = boto3.client('ec2')

          instance_id = events['InstanceId']

          print('[INFO] Waiting for instance status check to report ok', instance_id)
```

```
          instance_status = "null"

          while True:
            res = ec2.describe_instance_status(InstanceIds=[instance_id])

            if len(res['InstanceStatuses']) == 0:
              print("Instance status information is not available yet")
              time.sleep(5)
              continue

            instance_status = res['InstanceStatuses'][0]['InstanceStatus']['Status']

            print('[INFO] Polling to get status of the instance', instance_status)

            if instance_status == 'ok':
              break

            time.sleep(10)

          return {'Status': instance_status, 'InstanceId': instance_id}
      InputPayload: '{{ LaunchEc2Instance.payload }}'
    description: >-
      **About This Step**


      The python script continuously polls the instance status check value for
      the instance launched in Step 1 until the ```ok``` status is returned.
files:
  launch.py:
    checksums:
      sha256: 18871b1311b295c43d0f...[truncated]...772da97b67e99d84d342ef4aEXAMPLE
```

Exemplos de tratamento seguro de parâmetros

Os exemplos a seguir demonstram o tratamento seguro de parâmetros usando a variável de ambiente `interpolationType`.

Execução segura básica de comandos

Este exemplo mostra como tratar com segurança um parâmetro de comando:

> **Note**
>
> `allowedPattern` não é tecnicamente exigido em documentos do SSM que não usam chaves duplas: `{{ }}`

YAML

```
---

schemaVersion: '2.2'
description: An example document.
parameters:
  Message:
    type: String
    description: "Message to be printed"
    default: Hello
    interpolationType: ENV_VAR
    allowedPattern: "^[^"]*$"
mainSteps:
  - action: aws:runShellScript
    name: printMessage
    precondition:
      StringEquals:
        - platformType
        - Linux
    inputs:
      runCommand:
        - echo {{Message}}
```

JSON

```
{
    "schemaVersion": "2.2",
    "description": "An example document.",
    "parameters": {
        "Message": {
            "type": "String",
            "description": "Message to be printed",
            "default": "Hello",
            "interpolationType": "ENV_VAR",
            "allowedPattern": "^[^"]*$"
```

```
        }
    },
    "mainSteps": [{
        "action": "aws:runShellScript",
        "name": "printMessage",
        "precondition": {
           "StringEquals": ["platformType", "Linux"]
        },
        "inputs": {
            "runCommand": [
              "echo {{Message}}"
            ]
        }
    }]
}
```

Uso de parâmetros em linguagens interpretadas

Este exemplo demonstra o tratamento seguro de parâmetros em Python:

YAML

```
---
schemaVersion: '2.2'
description: 'Secure Python script execution'
parameters:
  inputData:
    type: String
    description: 'Input data for processing'
    interpolationType: 'ENV_VAR'
mainSteps:
  - action: aws:runPowerShellScript
    name: runPython
    inputs:
      runCommand:
        - |
          python3 -c '
          import os
          import json

          # Safely access parameter through environment variable
          input_data = os.environ.get("SSM_inputData", "")
```

```
          # Process the data
          try:
              processed_data = json.loads(input_data)
              print(f"Successfully processed: {processed_data}")
          except json.JSONDecodeError:
              print("Invalid JSON input")
          '
```

Exemplo de compatibilidade com versões anteriores

Este exemplo mostra como tratar parâmetros de forma segura enquanto mantém a compatibilidade com versões anteriores:

YAML

```
---
schemaVersion: '2.2'
description: 'Backwards compatible secure parameter handling'
parameters:
  userInput:
    type: String
    description: 'User input to process'
    interpolationType: 'ENV_VAR'
    allowedPattern: '^[^"]*$'

mainSteps:
  - action: aws:runShellScript
    name: processInput
    inputs:
      runCommand:
        - |
          # Handle both modern and legacy agent versions
          if [ -z "${SSM_userInput+x}" ]; then
              # Legacy agent - fall back to direct parameter reference
              export SSM_userInput="{{userInput}}"
          fi

          # Process the input securely
          echo "Processing input: $SSM_userInput"
```

> **Note**
>
> `allowedPattern` não é tecnicamente exigido em documentos do SSM que não usam chaves duplas: `{{ }}`

Práticas recomendadas de segurança de parâmetros

Siga estas práticas recomendadas ao tratar parâmetros em documentos do SSM:

- Use a interpolação de variáveis de ambiente: sempre use `interpolationType: "ENV_VAR"` para parâmetros de string que serão usados na execução do comando.
- Implemente uma validação de entrada: use `allowedPattern` para restringir os valores dos parâmetros a padrões seguros.
- Lide com sistemas legados: inclua lógica de fallback para as versões anteriores do SSM Agent que não são compatíveis com a interpolação de variáveis de ambiente.
- Escape de caracteres especiais: ao usar valores de parâmetros em comandos, escape adequadamente dos caracteres especiais para evitar a interpretação pelo shell.
- Limite o escopo do parâmetro: use os padrões de parâmetros mais restritivos possíveis para seu caso de uso.

Elementos e parâmetros de dados

Este tópico descreve os elementos de dados usados nos documentos do SSM. A versão do esquema usada para criar um documento define a sintaxe e os elementos de dados que o documento aceita. Recomendamos usar a versão 2.2 ou posterior do esquema para documentos do Command. Runbooks de automação usam o esquema versão 0.3. Além disso, runbooksde automação são compatíveis com o uso de Markdown, uma linguagem de marcação, que permite adicionar descrições de estilo wiki a documentos e etapas individuais dentro do documento. Para obter mais informações sobre o uso de Markdown, consulte Usar o Markdown no console no Guia de conceitos básicos do AWS Management Console.

A seção a seguir descreve os elementos de dados que você pode incluir em um documento SSM.

Elementos de dados de nível superior

schemaVersion

A versão do esquema a ser usada.

Tipo: versão

Obrigatório: Sim

description

Informações que você fornece para descrever a finalidade do documento. Você também pode usar esse campo para especificar se um parâmetro requer um valor para que um documento seja executado ou se fornecer um valor para o parâmetro é opcional. Parâmetros obrigatórios e opcionais podem ser vistos nos exemplos neste tópico.

Tipo: String

Obrigatório: Não

parameters

Uma estrutura que define os parâmetros que o documento aceita.

Para maior segurança ao tratar parâmetros de string, você pode usar a interpolação de variáveis de ambiente especificando a propriedade `interpolationType`. Quando definido como `ENV_VAR`, o sistema cria uma variável de ambiente denominada `SSM_`*`parameter-name`* contendo o valor do parâmetro.

Veja a seguir um exemplo de um parâmetro usando a variável de ambiente `interpolationType`:

```
{
    "schemaVersion": "2.2",
    "description": "An example document.",
    "parameters": {
        "Message": {
            "type": "String",
            "description": "Message to be printed",
            "default": "Hello",
            "interpolationType" : "ENV_VAR",
```

```
            "allowedPattern": "^[^"]*$"

        }
    },
    "mainSteps": [{
        "action": "aws:runShellScript",
        "name": "printMessage",
        "precondition" : {
           "StringEquals" : ["platformType", "Linux"]
        },
        "inputs": {
            "runCommand": [
              "echo {{Message}}"
            ]
        }
    }
}
```

> **Note**
>
> `allowedPattern` não é tecnicamente exigido em documentos do SSM que não usam chaves duplas: `{{ }}`

Para os parâmetros usados com frequência, recomendamos armazená-los no Parameter Store, uma ferramenta do AWS Systems Manager. Em seguida, você pode definir parâmetros em seu documento que façam referência a parâmetros do Parameter Store como o valor padrão deles. Para fazer referência a um parâmetro do Parameter Store, use a sintaxe a seguir.

```
{{ssm:parameter-name}}
```

Você pode usar um parâmetro que faça referência a um parâmetro do Parameter Store da mesma forma que faria com qualquer outro parâmetro de documento. No exemplo a seguir, o valor padrão para o parâmetro `commands` é o parâmetro `myShellCommands` do Parameter Store. Ao especificar o parâmetro `commands` como uma string `runCommand`, o documento executa os comandos armazenados no parâmetro `myShellCommands`.

YAML

```
---
```

```
schemaVersion: '2.2'
description: runShellScript with command strings stored as Parameter Store
 parameter
parameters:
  commands:
    type: StringList
    description: "(Required) The commands to run on the instance."
    default: ["{{ ssm:myShellCommands }}"],
            interpolationType : 'ENV_VAR'
            allowedPattern: '^[^"]*$'

mainSteps:
- action: aws:runShellScript
  name: runShellScriptDefaultParams
  inputs:
    runCommand:"{{ commands }}"
```

JSON

```
{
    "schemaVersion": "2.2",
    "description": "runShellScript with command strings stored as Parameter Store
 parameter",
    "parameters": {
      "commands": {
        "type": "StringList",
        "description": "(Required) The commands to run on the instance.",
        "default": ["{{ ssm:myShellCommands }}"],
        "interpolationType" : "ENV_VAR"
      }
    },
    "mainSteps": [
      {
        "action": "aws:runShellScript",
        "name": "runShellScriptDefaultParams",
        "inputs": {
            "runCommand": [
              "{{ commands }}"
          ]
        }
      }
    ]
  }
```

> **Note**
>
> Você pode fazer referência aos parâmetros `String` e `StringList` do Parameter Store na seção `parameters` do seu documento. Você não pode fazer referência a parâmetros `SecureString` do Parameter Store.

Para obter mais informações sobre o Parameter Store, consulte AWS Systems Manager Parameter Store.

Tipo: estrutura

A estrutura `parameters` aceita os seguintes campos e valores:

- `type`: (obrigatório) os valores permitidos incluem os seguintes: `String`, `StringList`, `Integer`, `Boolean`, `MapList` e `StringMap`. Para ver exemplos de cada tipo, consulte Exemplos de `type` de parâmetro de documentos SSM na próxima seção.

  > **Note**
  >
  > Os documentos do tipo de comando oferecem suporte somente aos tipos de parâmetros `String` e `StringList`.

- `description`: (opcional) uma descrição do parâmetro.
- `default`: (opcional) o valor padrão do parâmetro ou uma referência a um parâmetro no Parameter Store.
- `allowedValues`: (opcional) uma matriz de valores permitidos para o parâmetro. A definição de valores permitidos para o parâmetro valida a entrada do usuário. Se um usuário inserir um valor que não é permitido, a execução falhará ao iniciar.

  YAML

```
DirectoryType:
  type: String
  description: "(Required) The directory type to launch."
  default: AwsMad
  allowedValues:
  - AdConnector
  - AwsMad
  - SimpleAd
```

JSON

```
"DirectoryType": {
  "type": "String",
  "description": "(Required) The directory type to launch.",
  "default": "AwsMad",
  "allowedValues": [
    "AdConnector",
    "AwsMad",
    "SimpleAd"
  ]
}
```

- `allowedPattern`: (opcional) uma expressão regular que valida se a entrada do usuário corresponde ao padrão definido para o parâmetro. Se a entrada do usuário não corresponder ao padrão permitido, a execução não será iniciada.

> **Note**
>
> O Systems Manager realiza duas validações para `allowedPattern`. A primeira validação é realizada usando a biblioteca de regex Java no nível da API quando você usa um documento. A segunda validação é realizada no SSM Agent usando a biblioteca de regex GO antes de processar o documento.

YAML

```
InstanceId:
  type: String
  description: "(Required) The instance ID to target."
  allowedPattern: "^i-(?:[a-f0-9]{8}|[a-f0-9]{17})$"
  default: ''
```

JSON

```
"InstanceId": {
  "type": "String",
  "description": "(Required) The instance ID to target.",
  "allowedPattern": "^i-(?:[a-f0-9]{8}|[a-f0-9]{17})$",
  "default": ""
```

```
}
```

- `displayType`: (opcional) usado para exibir um `textfield` ou uma `textarea` no AWS Management Console. `textfield` é uma caixa de texto de uma única linha. `textarea` é uma área de texto de várias linhas.
- `minItems`: (opcional) o número mínimo de itens permitidos.
- `maxItems`: (opcional) o número máximo de itens permitidos.
- `minChars`: (opcional) o número mínimo de caracteres de parâmetro permitidos.
- `maxChars`: (opcional) o número máximo de caracteres de parâmetro permitidos.
- `interpolationType`: (opcional) define como os valores dos parâmetros são processados antes da execução do comando. Quando definido como `ENV_VAR`, o valor do parâmetro é disponibilizado como uma variável de ambiente denominada `SSM_`*`parameter-name`*. Esse recurso ajuda a evitar a injeção de comando tratando os valores dos parâmetros como strings literais.

  Tipo: String

  Valores válidos: `ENV_VAR`

Obrigatório: Não

variables

(Somente na versão 0.3 do esquema) Valores que podem ser referenciados ou atualizados em todas as etapas em um runbook de automação. As variáveis são semelhantes aos parâmetros, mas diferem de uma forma muito importante. Os valores dos parâmetros são estáticos no contexto de um runbook, mas os valores das variáveis podem ser alterados no contexto do runbook. Ao atualizar o valor de uma variável, o tipo de dados deve corresponder ao tipo de dados definido. Para informações sobre como atualizar valores de variáveis em uma automação, consulte [`aws:updateVariable`: atualiza um valor para uma variável do runbook]

Tipo: Boolean | Integer | MapList | String | StringList | StringMap

Obrigatório: Não

YAML

```
variables:
    payload:
```

```
        type: StringMap
        default: "{}"
```

JSON

```
{
    "variables": [
        "payload": {
            "type": "StringMap",
            "default": "{}"
        }
    ]
}
```

runtimeConfig

(Esquema somente para a versão 1.2) A configuração para a instância conforme aplicada por um ou mais plugins do Systems Manager. Não há garantia de execução dos plugins em sequência.

Tipo: Dictionary<string,PluginConfiguration>

Obrigatório: Não

mainSteps

(Somente para versões 0.3, 2.0 e 2.2 do esquema) um objeto que pode incluir várias etapas (plugins). Plugins são definidos dentro de etapas. As etapas são executadas na ordem sequencial listada no documento.

Tipo: Dictionary<string,PluginConfiguration>

Obrigatório: Sim

outputs

(Somente versão 0.3 do esquema) dados gerados pela execução deste documento que podem ser usados em outros processos. Por exemplo, se o documento criar uma nova AMI, você poderá especificar "CreateImage.ImageId" como o valor de saída e usar essa saída para criar novas instâncias em uma execução de automação subsequente. Para obter mais informações sobre saídas, consulte Uso de saídas de ações como entradas.

Tipo: Dictionary<string,OutputConfiguration>

Obrigatório: Não

files

(Somente a versão 0.3 do esquema) os arquivos de script (e as respectivas somas de verificação) anexados ao documento e executados durante uma execução de automação. Aplica-se somente a documentos que incluem a ação `aws:executeScript` e para os quais anexos foram especificados em uma ou mais etapas.

Para saber mais sobre os runtimes compatíveis com os runbooks do Automation, consulte `aws:executeScript` – Executa um script. Para obter mais informações sobre como incluir scripts em runbooks do Automation, consulte Uso de scripts em runbooks e Experiência de design visual para runbooks de automação.

Ao criar um runbook de automação com anexos, é necessário especificar arquivos de anexo usando a opção `--attachments` (na AWS CLI) ou `Attachments` (na API e no SDK). É possível especificar o local do arquivo para documentos do SSM e arquivos armazenados nos buckets do Amazon Simple Storage Service (Amazon S3). Para obter mais informações, consulte Anexos na referência da API da AWS Systems Manager.

YAML

```
---
files:
  launch.py:
    checksums:
      sha256: 18871b1311b295c43d0f...
[truncated]...772da97b67e99d84d342ef4aEXAMPLE
```

JSON

```
"files": {
    "launch.py": {
        "checksums": {
            "sha256": "18871b1311b295c43d0f...
[truncated]...772da97b67e99d84d342ef4aEXAMPLE"
        }
    }
}
```

Tipo: Dictionary<string,FilesConfiguration>

Obrigatório: Não

Exemplos de **`type`** de parâmetro de documentos SSM

Os tipos de parâmetro em documentos SSM são estáticos. Isso significa que o tipo de parâmetro não pode ser alterado depois de definido. Ao usar parâmetros com plugins de documento do , o tipo de um parâmetro não pode ser alterado dinamicamente dentro da entrada de um plugin. Por exemplo, você não pode fazer referência a um parâmetro `Integer` dentro da entrada `runCommand` do plugin `aws:runShellScript` porque essa entrada aceita uma string ou lista de strings. Para usar um parâmetro para uma entrada de plugin, o tipo de parâmetro deve corresponder ao tipo aceito. Por exemplo, você deve especificar um parâmetro de tipo `Boolean` para a entrada `allowDowngrade` do plugin `aws:updateSsmAgent`. Se o tipo de parâmetro não corresponder ao tipo de entrada de um plugin, o documento do não será validado e o sistema não criará o documento. Isso também é verdadeiro ao usar parâmetros downstream dentro de entradas para outros plugins ou ações do AWS Systems Manager Automation. Por exemplo, não é possível fazer referência a um parâmetro `StringList` dentro da entrada `documentParameters` do plugin `aws:runDocument`. A entrada `documentParameters` aceita um mapa de strings mesmo que o tipo de parâmetro de documento do SSM downtream seja um parâmetro `StringList` e corresponda ao parâmetro que você está referenciando.

Ao usar parâmetros com ações do Automation do , os tipos de parâmetro não são validados quando você cria o documento SSM na maioria dos casos. Somente quando você usa a ação `aws:runCommand`, os tipos de parâmetro são validados ao criar o documento SSM. Em todos os outros casos, a validação do parâmetro ocorre durante a execução da automação quando a entrada de uma ação é verificada antes de executar a ação. Por exemplo, se o parâmetro de entrada for uma `String` e você fizer referência a ele como o valor da entrada `MaxInstanceCount` da ação `aws:runInstances`, o documento SSM será criado. No entanto, ao executar o documento, a automação falha ao validar a ação `aws:runInstances` porque a entrada `MaxInstanceCount` requer um `Integer`.

Veja a seguir exemplos de cada parâmetro `type`.

String

Uma sequência de zero ou mais caracteres Unicode colocados entre aspas. Por exemplo, "i-1234567890abcdef0". Use barras invertidas como caractere de escape.

Os parâmetros de string podem incluir um campo opcional `interpolationType` com o valor `ENV_VAR` para habilitar a interpolação de variáveis de ambiente para melhorar a segurança.

YAML

```
---
InstanceId:
  type: String
  description: "(Optional) The target EC2 instance ID."
  interpolationType: ENV_VAR
```

JSON

```
"InstanceId":{
  "type":"String",
  "description":"(Optional) The target EC2 instance ID.",
  "interpolationType": "ENV_VAR"
}
```

StringList

Uma lista de itens de strings separadas por vírgulas. Por exemplo, ["cd ~", "pwd"].

YAML

```
---
commands:
  type: StringList
  description: "(Required) Specify a shell script or a command to run."
  default: ""
  minItems: 1
  displayType: textarea
```

JSON

```
"commands":{
  "type":"StringList",
  "description":"(Required) Specify a shell script or a command to run.",
  "minItems":1,
  "displayType":"textarea"
}
```

Booleano

Aceita somente `true` ou `false`. Não aceita "true" ou 0.

YAML

```
---
canRun:
  type: Boolean
  description: ''
  default: true
```

JSON

```
"canRun": {
  "type": "Boolean",
  "description": "",
  "default": true
}
```

Inteiro

Números inteiros. Não aceita números decimais, como 3,14159, ou números colocados entre aspas, como "3".

YAML

```
---
timeout:
  type: Integer
  description: The type of action to perform.
  default: 100
```

JSON

```
"timeout": {
  "type": "Integer",
  "description": "The type of action to perform.",
  "default": 100
}
```

StringMap

Um mapeamento de chaves para valores. Chaves e valores devem ser strings. Por exemplo, {"Env”: “Prod"}.

YAML

```
---
notificationConfig:
  type: StringMap
  description: The configuration for events to be notified about
  default:
    NotificationType: 'Command'
    NotificationEvents:
    - 'Failed'
    NotificationArn: "$dependency.topicArn"
  maxChars: 150
```

JSON

```
"notificationConfig" : {
  "type" : "StringMap",
  "description" : "The configuration for events to be notified about",
  "default" : {
    "NotificationType" : "Command",
    "NotificationEvents" : ["Failed"],
    "NotificationArn" : "$dependency.topicArn"
  },
  "maxChars" : 150
}
```

MapList

Uma lista de objetos StringMap.

YAML

```
blockDeviceMappings:
  type: MapList
  description: The mappings for the create image inputs
  default:
  - DeviceName: "/dev/sda1"
    Ebs:
      VolumeSize: "50"
  - DeviceName: "/dev/sdm"
    Ebs:
      VolumeSize: "100"
  maxItems: 2
```

JSON

```
"blockDeviceMappings":{
  "type":"MapList",
  "description":"The mappings for the create image inputs",
  "default":[
    {
      "DeviceName":"/dev/sda1",
      "Ebs":{
        "VolumeSize":"50"
      }
    },
    {
      "DeviceName":"/dev/sdm",
      "Ebs":{
        "VolumeSize":"100"
      }
    }
  ],
  "maxItems":2
}
```

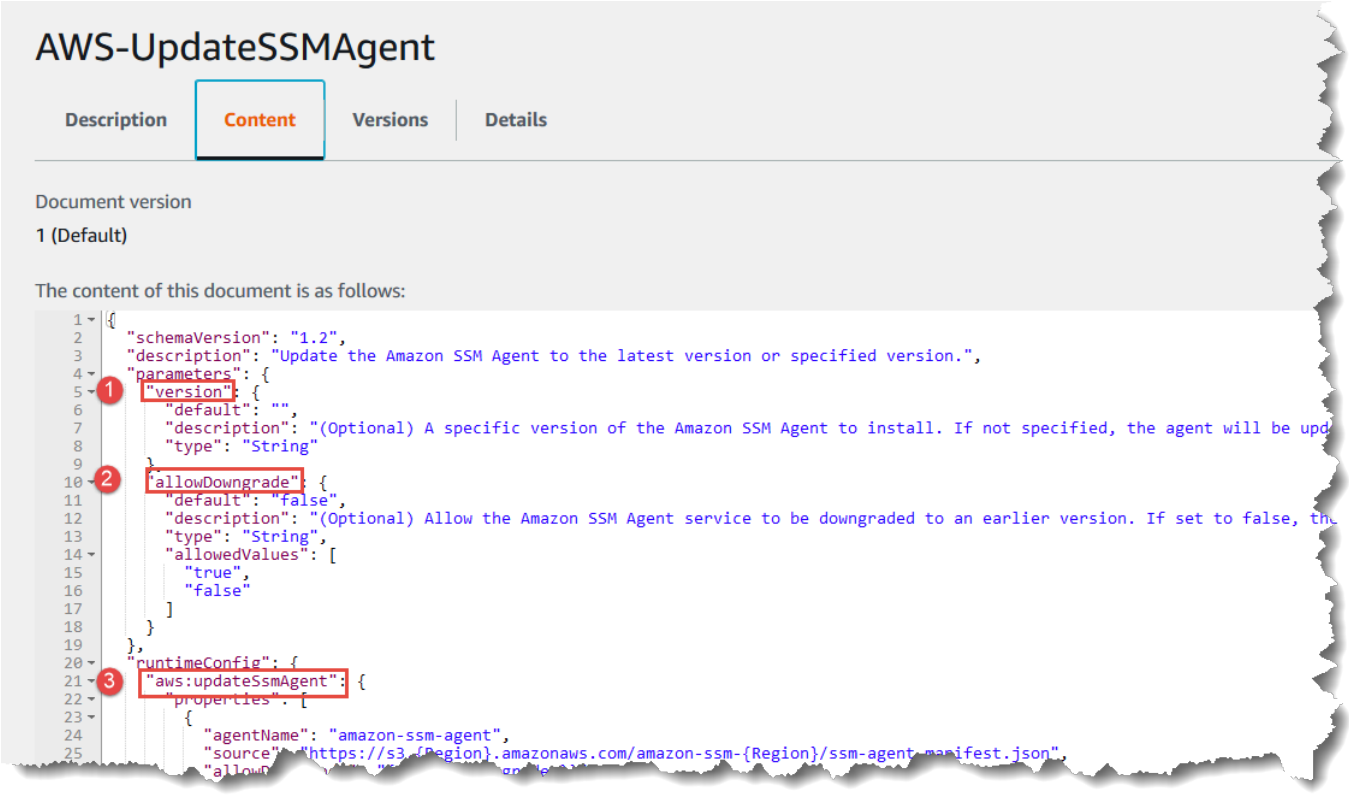
Visualizar o conteúdo do documento do SSM Command

Para visualizar os parâmetros obrigatórios e opcionais para um documento de comando do AWS Systems Manager (SSM), além das ações executadas pelo documento, você pode visualizar o conteúdo do documento no console do Systems Manager.

Para visualizar o conteúdo do documento do Comando SSM

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Documents.
3. Na caixa de pesquisa, selecione Tipo de documento e, em seguida, Comando.
4. Selecione o nome de um documento existente e, em seguida, escolha a guia Content (Conteúdo).
5. No campo de conteúdo, revise os parâmetros disponíveis e as etapas de ação para o documento.

Por exemplo, a seguinte imagem mostra que (1) `version` e (2) `allowDowngrade` são parâmetros opcionais para o documento `AWS-UpdateSSMAgent` e que a primeira ação executada pelo documento é (3) `aws:updateSsmAgent`.



Referência de plug-ins de documentos de comando

Essa referência descreve os plugins que você pode especificar em um documento de comando do AWS Systems Manager (SSM). Esses plugins não podem ser usados em documentos de automação do SSM que usam ações de automação. Para obter informações sobre ações de automação do AWS Systems Manager, consulte Referência de ações do Systems Manager Automation.

O Systems Manager determina as ações a serem executadas em uma instância gerenciada lendo o conteúdo de um documento do SSM. Todo documento contém uma seção de execução de código. Dependendo da versão do esquema do documento, essa seção de execução de código pode conter um ou mais plugins ou etapas. Para a finalidade deste tópico da Ajuda, plugins e etapas são chamados de plugins. Esta seção inclui informações sobre cada um dos plugins do Systems Manager. Para obter mais informações sobre documentos, incluindo informações sobre a criação de documentos e as diferenças entre as versões de esquema, consulte Documentos do AWS Systems Manager.

Para plug-ins que aceitam parâmetros String, como `aws:runShellScript` e `aws:runPowerShellScript`, o parâmetro `interpolationType` pode ser usado para aumentar a segurança, tratando as entradas de parâmetros como literais de string em vez de comandos potencialmente executáveis. Por exemplo:

```
{
    "schemaVersion": "2.2",
    "description": "runShellScript with command strings stored as Parameter Store
 parameter",
    "parameters": {
      "commands": {
        "type": "StringList",
        "description": "(Required) The commands to run on the instance.",
        "default": ["{{ ssm:myShellCommands }}"],
        "interpolationType" : "ENV_VAR"
      }
    },
    //truncated
 }
```

> **Note**
>
> Alguns dos plugins descritos aqui são executados apenas em instâncias do Windows Server ou em instâncias do Linux. São indicadas dependências de plataforma para cada plugin. Os plugins de documento a seguir são compatíveis com instâncias do Amazon Elastic Compute Cloud (Amazon EC2) para o macOS:
>
> - `aws:refreshAssociation`
> - `aws:runShellScript`
> - `aws:runPowerShellScript`
> - `aws:softwareInventory`
> - `aws:updateSsmAgent`

Conteúdo

- Entradas compartilhadas
- aws:applications
- aws:cloudWatch

- aws:configureDocker
- aws:configurePackage
- aws:domainJoin
- aws:downloadContent
- aws:psModule
- aws:refreshAssociation
- aws:runDockerAction
- aws:runDocument
- aws:runPowerShellScript
- aws:runShellScript
- aws:softwareInventory
- aws:updateAgent
- aws:updateSsmAgent

Entradas compartilhadas

comSSM Agentversão 3.0.502 e posterior somente, todos os plugins podem usar as seguintes entradas:

finallyStep

A última etapa que você deseja que o documento seja executado. Se essa entrada for definida para uma etapa, ela terá precedência sobre um`exit`especificado no parâmetroon`Failure`ouon`Success`entradas. Para que uma etapa com essa entrada seja executada conforme esperado, a etapa deve ser a última definida no`mainSteps`do documento.

Tipo: booliano

Valores válidos: `true` | `false`

Obrigatório: Não

onFailure

Se você especificar esta entrada para um plugin com a opção`exite` a etapa falhar, o status da etapa reflete a falha e o documento não executa as etapas restantes, a menos que um`finallyStep`foi definido. Se você especificar esta entrada para um plugin com a

opçãosuccessAndExite a etapa falhar, o status da etapa mostra bem-sucedida e o documento não executa as etapas restantes, a menos que umfinallyStepfoi definido.

Tipo: String

Valores válidos: exit | successAndExit

Obrigatório: Não

onSuccess

Se você especificar essa entrada para um plugin e a etapa for executada com êxito, o documento não executará nenhuma etapa restante, a menos que umfinallyStepfoi definido.

Tipo: String

Valores válidos: exit

Obrigatório: Não

YAML

```
---
schemaVersion: '2.2'
description: Shared inputs example
parameters:
  customDocumentParameter:
    type: String
    description: Example parameter for a custom Command-type document.
mainSteps:
- action: aws:runDocument
  name: runCustomConfiguration
  inputs:
    documentType: SSMDocument
    documentPath: "yourCustomDocument"
    documentParameters: '"documentParameter":{{customDocumentParameter}}'
    onSuccess: exit
- action: aws:runDocument
  name: ifConfigurationFailure
  inputs:
    documentType: SSMDocument
    documentPath: "yourCustomRepairDocument"
    onFailure: exit
```

```
- action: aws:runDocument
  name: finalConfiguration
  inputs:
    documentType: SSMDocument
    documentPath: "yourCustomFinalDocument"
    finallyStep: true
```

JSON

```
{
   "schemaVersion": "2.2",
   "description": "Shared inputs example",
   "parameters": {
      "customDocumentParameter": {
         "type": "String",
         "description": "Example parameter for a custom Command-type document."
      }
   },
   "mainSteps":[
      {
         "action": "aws:runDocument",
         "name": "runCustomConfiguration",
         "inputs": {
            "documentType": "SSMDocument",
            "documentPath": "yourCustomDocument",
            "documentParameters": "\"documentParameter\":
{{customDocumentParameter}}",
            "onSuccess": "exit"
         }
      },
      {
         "action": "aws:runDocument",
         "name": "ifConfigurationFailure",
         "inputs": {
            "documentType": "SSMDocument",
            "documentPath": "yourCustomRepairDocument",
            "onFailure": "exit"
         }
      },
      {
         "action": "aws:runDocument",
         "name":"finalConfiguration",
         "inputs": {
```

```
            "documentType": "SSMDocument",
            "documentPath": "yourCustomFinalDocument",
            "finallyStep": true
         }
      }
   ]
}
```

aws:applications

Instala, repara ou desinstala aplicativos em uma instância do EC2. Este plugin é executado somente em sistemas operacionais Windows Server.

Sintaxe

Esquema 2.2

YAML

```
---
schemaVersion: '2.2'
description: aws:applications plugin
parameters:
  source:
    description: "(Required) Source of msi."
    type: String
mainSteps:
- action: aws:applications
  name: example
  inputs:
    action: Install
    source: "{{ source }}"
```

JSON

```
{
  "schemaVersion":"2.2",
  "description":"aws:applications",
  "parameters":{
    "source":{
    "description":"(Required) Source of msi.",
    "type":"String"
```

```
    }
  },
  "mainSteps":[
    {
      "action":"aws:applications",
      "name":"example",
      "inputs":{
        "action":"Install",
        "source":"{{ source }}"
      }
    }
  ]
}
```

Esquema 1.2

YAML

```
---
runtimeConfig:
  aws:applications:
    properties:
    - id: 0.aws:applications
      action: "{{ action }}"
      parameters: "{{ parameters }}"
      source: "{{ source }}"
      sourceHash: "{{ sourceHash }}"
```

JSON

```
{
   "runtimeConfig":{
      "aws:applications":{
         "properties":[
            {
               "id":"0.aws:applications",
               "action":"{{ action }}",
               "parameters":"{{ parameters }}",
               "source":"{{ source }}",
               "sourceHash":"{{ sourceHash }}"
            }
         ]
```

```
        }
    }
}
```

Propriedades

action

A medida a ser tomada.

Tipo: Enum

Valores válidos: `Install` | `Repair` | `Uninstall`

Obrigatório: Sim

parameters

Os parâmetros para o instalador.

Tipo: String

Obrigatório: Não

origem

O URL do arquivo `.msi` para o aplicativo.

Tipo: string

Obrigatório: Sim

sourceHash

O hash SHA256 do arquivo `.msi`.

Tipo: String

Obrigatório: Não

`aws:cloudWatch`

Exporte dados do Windows Server para o Amazon CloudWatch ou Amazon CloudWatch Logs e monitore os dados usando as métricas do CloudWatch. Este plugin é executado somente em

sistemas operacionais Windows Server. Para obter mais informações sobre como configurar a integração do CloudWatch, ao Amazon Elastic Compute Cloud (Amazon EC2) consulte Coletar métricas, logs e rastreamentos com o agente do CloudWatch no Guia de usuário do Amazon CloudWatch.

> **Important**
>
> O agente unificado do CloudWatch substituiu SSM Agent como a ferramenta para enviar dados de log para o Amazon CloudWatch Logs. Não há compatibilidade com o plugin aws:cloudWatch do SSM Agent. Recomendamos usar somente o agente do CloudWatch unificado nos processos de coleta de logs. Para obter mais informações, consulte os tópicos a seguir.
>
> - Enviar logs de nós para o CloudWatch Logs unificado (agente do CloudWatch)
> - Migrar a coleta de logs de nós do Windows Server para o agente do CloudWatch
> - Coletar métricas, logs e rastreamentos com o agente do CloudWatch no Guia do usuário do Amazon CloudWatch.

Você pode exportar e monitorar os seguintes tipos de dados:

ApplicationEventLog

Envia dados de log de eventos da aplicação para o CloudWatch Logs.

CustomLogs

Envia qualquer arquivo de log baseado em texto para o Amazon CloudWatch Logs. O plugin do CloudWatch cria uma impressão digital para os arquivos de log. O sistema associa então um deslocamento de dados com cada impressão digital. O plugin faz upload dos arquivos quando há alterações, registra o deslocamento e associa o deslocamento com uma impressão digital. Esse método é usado para evitar uma situação em que um usuário ativa o plugin, associa o serviço com um diretório que contém um grande número de arquivos, e o sistema faz upload de todos os arquivos.

> **Warning**
>
> Lembre-se de que, se seu aplicativo truncar ou tentar limpar os logs durante a sondagem, qualquer log especificado para `LogDirectoryPath` pode perder entradas. Se, por

exemplo, você quiser limitar o tamanho do arquivo de log, crie um novo arquivo de log quando o limite for atingido e continue a gravar dados no novo arquivo.

ETW

Envia os dados de rastreamento de eventos para Windows (ETW) ao CloudWatch Logs.

IIS

Envia dados de log do IIS ao CloudWatch Logs.

PerformanceCounter

Envia contadores de performance do Windows para o CloudWatch. Você poderá selecionar categorias diferentes para fazer upload no CloudWatch como métricas. Para cada contador de performance que você deseja carregar, crie uma seção PerformanceCounter com um ID exclusivo (por exemplo, "PerformanceCounter2", "PerformanceCounter3" e assim por diante) e configure as respectivas propriedades.

> **Note**
>
> Se o AWS Systems Manager SSM Agent ou o plugin do CloudWatch for interrompido, os dados do contador de performance não serão registrados em log no CloudWatch. Esse comportamento é diferente daquele dos logs personalizados ou dos logs de eventos do Windows. Os logs personalizados e logs de eventos do Windows preservarão dados do contador de performance e farão upload dos dados no CloudWatch depois que o SSM Agent ou o plugin do CloudWatch estiver disponível.

SecurityEventLog

Envia dados de log de eventos de segurança para o CloudWatch Logs.

SystemEventLog

Envia dados de log de eventos de sistema ao CloudWatch Logs

Você pode definir os seguintes destinos para os dados:

CloudWatch

O destino para o qual os dados de métrica do contador de performance são enviados. Você pode incluir mais seções com IDs exclusivos (por exemplo, "CloudWatch2", "CloudWatch3" etc.) e especificar uma região diferente para cada novo ID a fim de enviar os mesmos dados para diferentes locais.

CloudWatchLogs

O destino para o qual os dados de log são enviados. Você pode incluir mais seções com IDs exclusivos (por exemplo, "CloudWatchLogs2", "CloudWatchLogs3" etc.) e especificar uma região diferente para cada novo ID a fim de enviar os mesmos dados para diferentes locais.

Sintaxe

```
"runtimeConfig":{
        "aws:cloudWatch":{
            "settings":{
                "startType":"{{ status }}"
            },
            "properties":"{{ properties }}"
        }
    }
```

Configurações e propriedades

AccessKey

Seu ID de chave de acesso da . Essa propriedade é obrigatória, a menos que você tenha lançado sua instância usando uma função do IAM. Essa propriedade não pode ser usada com o SSM.

Tipo: String

Obrigatório: Não

CategoryName

A categoria de contador de performance no Performance Monitor.

Tipo: string

Obrigatório: Sim

CounterName

O nome do contador de performance no Performance Monitor.

Tipo: string

Obrigatório: Sim

CultureName

O local em que o time stamp é registrado. Se CultureName estiver em branco, o mesmo local usado atualmente por sua instância do Windows Server será usado como o padrão.

Tipo: String

Valores válidos: para obter uma lista de valores comportados, consulte National Language Support (NLS) no site da Microsoft. Observe que os valores div, div-MV, hu e hu-HU não são compatíveis.

Obrigatório: Não

DimensionName

Uma dimensão para sua métrica do Amazon CloudWatch. Se você especificar `DimensionName`, também deverá especificar `DimensionValue`. Esses parâmetros produzem outro modo de exibição para a listagem de métricas. Você pode usar a mesma dimensão para várias métricas de modo a visualizar todas as métricas que pertencem a uma dimensão específica.

Tipo: String

Obrigatório: Não

DimensionValue

Um valor de dimensão para a métrica do Amazon CloudWatch.

Tipo: String

Obrigatório: Não

Codificação

A codificação de arquivo a ser usada (por exemplo, UTF-8). Use o nome da codificação, não o nome da exibição.

Tipo: String

Valores válidos: para obter uma lista de valores comportados, consulte Encoding Class na biblioteca Microsoft Learn.

Obrigatório: Sim

Filtro

O prefixo dos nomes de log. Deixe esse parâmetro em branco para monitorar todos os arquivos.

Tipo: String

Valores válidos: para obter uma lista de valores compatíveis, consulte FileSystemWatcherFilter property na biblioteca do MSDN.

Obrigatório: Não

Fluxos

Cada tipo de dados para fazer upload, bem como o destino dos dados (CloudWatch ou CloudWatch Logs). Por exemplo, para enviar um contador de performance definido de acordo com `"Id": "PerformanceCounter"` para o destino do CloudWatch definido em `"Id": "CloudWatch"`, insira "PerformanceCounter,CloudWatch". Da mesma forma, para enviar o log personalizado, o log do ETW e o log do sistema para o destino do CloudWatch Logs definido em `"Id": "ETW"`, insira “(ETW),CloudWatchLogs”. Além disso, é possível enviar o mesmo contador de performance ou arquivo de log para mais de um destino. Por exemplo, para enviar o log do aplicativo para dois destinos diferentes que você definiu em `"Id": "CloudWatchLogs"` e `"Id": "CloudWatchLogs2"`, insira "ApplicationEventLog, (CloudWatchLogs, CloudWatchLogs2)".

Tipo: String

Valores válidos (origem): `ApplicationEventLog` | `CustomLogs` | `ETW` | `PerformanceCounter` | `SystemEventLog` | `SecurityEventLog`

Valores válidos (destino): `CloudWatch` | `CloudWatchLogs` | `CloudWatch`*n* | `CloudWatchLogs`*n*

Obrigatório: Sim

FullName

O nome completo do componente.

Tipo: string

Obrigatório: Sim

Id

Identifica a origem de dados ou o destino. Esse identificador deve ser exclusivo no arquivo de configuração.

Tipo: string

Obrigatório: Sim

InstanceName

O nome da instância do contador de performance. Não use um asterisco (*) para indicar todas as instâncias, pois cada componente do contador de performance só é compatível com uma única métrica. Você pode, porém, usar _Total.

Tipo: string

Obrigatório: Sim

Níveis

Os tipos de mensagem a enviar para o Amazon CloudWatch.

Tipo: String

Valores válidos:

- 1 - Somente o upload de mensagens de erro.
- 2 - Somente o upload de mensagens de aviso.
- 4 - Somente o upload de mensagens de informação.

Observe que você pode adicionar valores para incluir mais de um tipo de mensagem. Por exemplo, 3 significa que mensagens de erro (1) e mensagens de aviso (2) estão incluídas. Um valor de 7 significa que mensagens de erro (1), mensagens de aviso (2) e mensagens informativas (4) estão incluídas.

Obrigatório: Sim

> **Note**
>
> Os logs de segurança do Windows devem definir os níveis como 7.

LineCount

O número de linha no cabeçalho para identificar o arquivo de log. Por exemplo, os arquivos de log do IIS têm cabeçalhos praticamente idênticos. Você pode especificar 3, que leria as três primeiras linhas do cabeçalho do arquivo de log para identificá-lo. Em arquivos de log do IIS, a terceira linha é o time stamp que é diferente entre arquivos de log.

Tipo: inteiro

Obrigatório: Não

LogDirectoryPath

Em CustomLogs, o caminho em que os logs são armazenados na instância do EC2. Para logs do IIS, a pasta em que os logs do IIS são armazenados para um site específico (por exemplo, C:\\inetpub\\logs\\LogFiles\\W3SVC***n***). Para logs do IIS, somente o formato de log W3C é comportado. Não há suporte para os formatos IIS, NCSA e Personalizado.

Tipo: string

Obrigatório: Sim

LogGroup

O nome para o seu grupo de logs. Esse nome é exibido na tela Log Groups (Grupos de logs) no console do CloudWatch.

Tipo: String

Obrigatório: Sim

LogName

O nome do arquivo de log.

1. Para encontrar o nome do log, no Visualizador de Eventos, no painel de navegação, clique em Logs de aplicações e serviços.
2. Na lista de logs, clique com o botão direito do mouse no log do qual você deseja fazer upload (por exemplo, Microsoft>Windows>Backup>Operacional) e clique em Criar Modo de Exibição Personalizado.
3. Na caixa de diálogo Create Custom View (Criar modo de exibição personalizado), selecione a guia XML. O LogName está na tag <Select Path=> (por exemplo, `Microsoft-Windows-Backup`). Copie o texto para o parâmetro LogName.

Tipo: String

Valores válidos: `Application` | `Security` | `System` | `Microsoft-Windows-WinINet/Analytic`

Obrigatório: Sim

LogStream

O fluxo de logs de destino. Se você usar {instance_id}, o padrão, o ID de instância dessa instância será usado como nome do fluxo de logs.

Tipo: String

Valores válidos: `{instance_id}` | `{hostname}` | `{ip_address}` *`<log_stream_name>`*

Se você especificar o nome de uma transmissão de log que ainda não existe, o CloudWatch Logs a criará automaticamente. Você pode usar uma cadeia de caracteres literal ou as variáveis predefinidas ({instance_id}, {hostname}, {ip_address}, ou uma combinação das três, para definir um nome de fluxo de log.

O nome da transmissão de logs especificado nesse parâmetro aparece na tela Log Groups > Streams for ***`<YourLogStream>`*** (Grupos de logs > Transmissões para <SuaTransmissãoDeLogs>) no console do CloudWatch.

Obrigatório: Sim

MetricName

A métrica do CloudWatch na qual você deseja que os dados de performance sejam incluídos.

> **Note**
>
> Não use caracteres especiais no nome. Caso contrário, a métrica e os alarmes associados podem não funcionar.

Tipo: string

Obrigatório: Sim

NameSpace

O namespace da métrica em que você deseja que os dados do contador de performance sejam gravados.

Tipo: string

Obrigatório: Sim

PollInterval

Quantos segundos devem transcorrer para que seja feito upload do novo contador de performance e dos dados de log.

Tipo: inteiro

Valores válidos: defina como 5 ou mais segundos. É recomendável 15 segundos (00:00:15).

Obrigatório: Sim

Região

A Região da AWS para a qual você deseja enviar dados de log. Embora seja possível enviar contadores de performance para uma região diferente daquela para onde você envia os dados de log, recomendamos que você defina esse parâmetro como a mesma região onde sua instância está sendo executada.

Tipo: String

Valores válidos: IDs de regiões do Regiões da AWS suportado pelo Systems Manager e pelo CloudWatch Logs, como `us-east-2`, `eu-west-1` e `ap-southeast-1`. Para listas de Regiões da AWS com suporte por cada serviço, consulte Amazon CloudWatch Logs Service Endpoints e Systems Manager service endpoints no Referência geral da Amazon Web Services.

Obrigatório: Sim

SecretKey

Sua chave de acesso secreta da . Essa propriedade é obrigatória, a menos que você tenha lançado sua instância usando uma função do IAM.

Tipo: String

Obrigatório: Não

startType

Ative ou desative o CloudWatch na instância.

Tipo: String

Valores válidos: `Enabled` | `Disabled`

Obrigatório: Sim

TimestampFormat

O formato time stamp que você deseja usar. Para obter uma lista de valores compatíveis, consulte Cadeias de caracteres de formato de data e hora personalizado na biblioteca do MSDN.

Tipo: string

Obrigatório: Sim

TimeZoneKind

Fornece informações de fuso horário quando essas informações não estiverem incluídas no time stamp do log. Se esse parâmetro for deixado em branco e se o carimbo de data/hora não incluir informações de fuso horário, o CloudWatch Logs adotará como padrão o fuso horário local. Esse parâmetro será ignorado se o carimbo já incluir informações de fuso horário.

Tipo: String

Valores válidos: `Local` | `UTC`

Obrigatório: Não

Unidade

A unidade de medida adequada para a métrica.

Tipo: String

Valores válidos: segundos | microssegundos | milissegundos | bytes | kilobytes | megabytes | gigabytes | terabytes | bits | kilobits | megabits | gigabits | terabits | porcentagem | contagem | bytes/segundo | kilobytes/segundo | megabytes/segundo | gigabytes/segundo | terabytes/segundo | bits/segundo | kilobits/segundo | megabits/segundo | gigabits/segundo | terabits/segundo | contagem/segundos | nenhum.

Obrigatório: Sim

`aws:configureDocker`

(Esquema versão 2.0 ou posterior) configura uma instância para trabalhar com contêineres e o docker. Este plug-in é compatível com a maioria das variantes do Linux e sistemas operacionais Windows Server.

Sintaxe

Esquema 2.2

YAML

```
---
schemaVersion: '2.2'
description: aws:configureDocker
parameters:
  action:
    description: "(Required) The type of action to perform."
    type: String
    default: Install
    allowedValues:
    - Install
    - Uninstall
mainSteps:
- action: aws:configureDocker
  name: configureDocker
  inputs:
    action: "{{ action }}"
```

JSON

```
{
  "schemaVersion": "2.2",
  "description": "aws:configureDocker plugin",
  "parameters": {
    "action": {
      "description": "(Required) The type of action to perform.",
      "type": "String",
      "default": "Install",
      "allowedValues": [
        "Install",
        "Uninstall"
      ]
    }
  },
  "mainSteps": [
    {
      "action": "aws:configureDocker",
      "name": "configureDocker",
```

```
      "inputs": {
        "action": "{{ action }}"
      }
    }
  ]
}
```

Entradas

action

O tipo de ação a ser executada.

Tipo: Enum

Valores válidos: `Install` | `Uninstall`

Obrigatório: Sim

`aws:configurePackage`

(Esquema versão 2.0 ou posterior) Instale ou desinstale um pacote do AWS Systems Manager Distributor. Você pode instalar a versão mais recente, a versão padrão ou uma versão especificada do pacote. Os pacotes fornecidos pela AWS também são compatíveis. Este plugin é executado nos sistemas operacionais Windows Server e Linux, mas nem todos os pacotes disponíveis são compatíveis com sistemas operacionais Linux.

Pacotes da AWS disponíveis para Windows Server incluem o seguinte: `AWSPVDriver`, `AWSNVMe`, `AwsEnaNetworkDriver`, `AwsVssComponents`, `AmazonCloudWatchAgent`, `CodeDeployAgent`, e `AWSSupport-EC2Rescue.`

Os pacotes disponíveis da AWS para sistemas operacionais Linux incluem o seguinte: `AmazonCloudWatchAgent`, `CodeDeployAgent` `AWSSupport-EC2Rescue`.

Sintaxe

Esquema 2.2

YAML

```
---
```

```
schemaVersion: '2.2'
description: aws:configurePackage
parameters:
  name:
    description: "(Required) The name of the AWS package to install or uninstall."
    type: String
  action:
    description: "(Required) The type of action to perform."
    type: String
    default: Install
    allowedValues:
    - Install
    - Uninstall
  ssmParameter:
    description: "(Required) Argument stored in Parameter Store."
    type: String
    default: "{{ ssm:parameter_store_arg }}"
mainSteps:
- action: aws:configurePackage
  name: configurePackage
  inputs:
    name: "{{ name }}"
    action: "{{ action }}"
    additionalArguments:
      "{\"SSM_parameter_store_arg\": \"{{ ssmParameter }}\", \"SSM_custom_arg\":
 \"myValue\"}"
```

JSON

```
{
   "schemaVersion": "2.2",
   "description": "aws:configurePackage",
   "parameters": {
      "name": {
         "description": "(Required) The name of the AWS package to install or
 uninstall.",
         "type": "String"
      },
      "action": {
         "description": "(Required) The type of action to perform.",
         "type": "String",
         "default": "Install",
         "allowedValues": [
```

```
            "Install",
            "Uninstall"
         ]
      },
      "ssmParameter": {
         "description": "(Required) Argument stored in Parameter Store.",
         "type": "String",
         "default": "{{ ssm:parameter_store_arg }}"
      }
   },
   "mainSteps": [
      {
         "action": "aws:configurePackage",
         "name": "configurePackage",
         "inputs": {
            "name": "{{ name }}",
            "action": "{{ action }}",
            "additionalArguments": "{\"SSM_parameter_store_arg\":
 \"{{ ssmParameter }}\", \"SSM_custom_arg\": \"myValue\"}"
         }
      }
   ]
}
```

Entradas

nome

O nome do pacote da AWS a ser instalado ou desinstalado. Os pacotes disponíveis incluem o seguinte: `AWSPVDriver`, `AwsEnaNetworkDriver`, `AwsVssComponents` e `AmazonCloudWatchAgent`.

Tipo: string

Obrigatório: Sim

action

Instala ou desinstala um pacote.

Tipo: Enum

Valores válidos: `Install` | `Uninstall`

Obrigatório: Sim

installationType

O tipo de instalação a ser executada. Se você especificar `Uninstall and reinstall`, o pacote será completamente desinstalado e, depois, reinstalado. A aplicação estará indisponível até que a reinstalação seja concluída. Se você especificar `In-place update`, somente arquivos novos ou alterados serão adicionados à instalação existente de acordo com as instruções fornecidas em um script de atualização. O aplicativo permanece disponível durante todo o processo de atualização. O`In-place update`A opção não tem suporte paraAWS-pacotes publicados.`Uninstall and reinstall`é o valor padrão.

Tipo: Enum

Valores válidos: `Uninstall and reinstall` | `In-place update`

Obrigatório: Não

additionalArguments

Uma string JSON dos parâmetros adicionais a serem fornecidos aos scripts de instalação, desinstalação ou atualização. Cada parâmetro deve ser prefixado com `SSM_`. Você pode fazer referência a um parâmetro Parameter Store em seus argumentos adicionais usando a convenção `{{ssm:`*`parameter-name`*`}}`. Para usar o parâmetro adicional em seu script de instalação, desinstalação ou atualização, você deve fazer referência ao parâmetro como uma variável de ambiente usando a sintaxe apropriada para o sistema operacional. Por exemplo, no PowerShell, você faz referência ao`SSM_arg`Argumento como`$Env:SSM_arg`. Não há limite para o número de argumentos que você define, mas a entrada de argumento adicional tem um limite de 4096 caracteres. Esse limite inclui todas as chaves e valores que você define.

Tipo: StringMap

Obrigatório: Não

version

Uma versão específica do pacote a ser instalada ou desinstalada. No caso de instalação, o sistema instala a última versão publicada, por padrão. No caso de desinstalação, o sistema desinstala a versão atualmente instalada, por padrão. Se não for encontrada nenhuma versão instalada, é feito download da última versão publicada e a ação de desinstalação é executada.

Tipo: String

Obrigatório: Não

`aws:domainJoin`

Ingressar uma instância do EC2 em um domínio. Esse plugin é executado em sistemas operacionais Linux e Windows Server. Este plugin altera o nome do host das instâncias Linux para o formato EC2AMAZ-*XXXXXXX*. Para obter mais informações sobre como ingressar instâncias do EC2, consulte Associe uma instância do EC2 ao Diretório do Microsoft AD gerenciado pela AWS no Manual de administração do AWS Directory Service.

Sintaxe

Esquema 2.2

YAML

```
---
schemaVersion: '2.2'
description: aws:domainJoin
parameters:
  directoryId:
    description: "(Required) The ID of the directory."
    type: String
  directoryName:
    description: "(Required) The name of the domain."
    type: String
  directoryOU:
    description: "(Optional) The organizational unit to assign the computer object
 to."
    type: String
  dnsIpAddresses:
    description: "(Required) The IP addresses of the DNS servers for your
 directory."
    type: StringList
  hostname:
    description: "(Optional) The hostname you want to assign to the node."
    type: String
mainSteps:
- action: aws:domainJoin
  name: domainJoin
  inputs:
    directoryId: "{{ directoryId }}"
```

```
    directoryName: "{{ directoryName }}"
    directoryOU: "{{ directoryOU }}"
    dnsIpAddresses: "{{ dnsIpAddresses }}"
    hostname: "{{ hostname }}"
```

JSON

```
{
  "schemaVersion": "2.2",
  "description": "aws:domainJoin",
  "parameters": {
    "directoryId": {
      "description": "(Required) The ID of the directory.",
      "type": "String"
    },
    "directoryName": {
      "description": "(Required) The name of the domain.",
      "type": "String"
    },
    "directoryOU": {
        "description": "(Optional) The organizational unit to assign the computer
 object to.",
        "type": "String"
      },
    "dnsIpAddresses": {
      "description": "(Required) The IP addresses of the DNS servers for your
 directory.",
      "type": "StringList"
    },
    "hostname": {
        "description": "(Optional) The hostname you want to assign to the node.",
        "type": "String"
      }
  },
  "mainSteps": [
    {
      "action": "aws:domainJoin",
      "name": "domainJoin",
      "inputs": {
        "directoryId": "{{ directoryId }}",
        "directoryName": "{{ directoryName }}",
        "directoryOU":"{{ directoryOU }}",
        "dnsIpAddresses":"{{ dnsIpAddresses }}",
```

```
        "hostname":"{{ hostname }}"
      }
    }
  ]
}
```

Esquema 1.2

YAML

```
---
runtimeConfig:
  aws:domainJoin:
    properties:
      directoryId: "{{ directoryId }}"
      directoryName: "{{ directoryName }}"
      directoryOU: "{{ directoryOU }}"
      dnsIpAddresses: "{{ dnsIpAddresses }}"
```

JSON

```
{
   "runtimeConfig":{
      "aws:domainJoin":{
         "properties":{
            "directoryId":"{{ directoryId }}",
            "directoryName":"{{ directoryName }}",
            "directoryOU":"{{ directoryOU }}",
            "dnsIpAddresses":"{{ dnsIpAddresses }}"
         }
      }
   }
}
```

Propriedades

directoryId

O ID do diretório.

Tipo: string

Obrigatório: Sim

Exemplo: "directoryId": "d-1234567890"

directoryName

O nome do domínio.

Tipo: string

Obrigatório: Sim

Exemplo: "directoryName": "example.com"

directoryOU

A unidade organizacional (UO).

Tipo: String

Obrigatório: Não

Exemplo: "directoryOU": "OU=test,DC=example,DC=com"

dnsIpAddresses

Os endereços IP dos servidores DNS.

Tipo: StringList

Obrigatório: Sim

Exemplo: "dnsIpAddresses": ["198.51.100.1","198.51.100.2"]

hostname

O nome de host que você deseja atribuir ao nó. Caso não seja fornecido, não haverá modificação de nome para instâncias Windows, enquanto instâncias Linux utilizarão o padrão de nomenclatura predefinido. Quando fornecido, as instâncias Windows utilizarão o valor exato indicado. Para instâncias Linux, servirá como um prefixo (a menos que `keepHostName` seja definido como “true”).

Tipo: String

Obrigatório: Não

keepHostName

Determina se o nome do host é alterado para instâncias Linux quando ingressadas no domínio. É um parâmetro somente Linux. Por padrão (sem entradas para `hostname`, `hostnameNumAppendDigits` e com `keepHostName` definido como "false"), hosts Linux serão renomeados para o padrão EC2AMAZ-XXXXXX. Ao ser definido como “true”, mantém o nome de host original e ignora entradas para `hostname` e `hostnameNumAppendDigits`.

Tipo: booliano

Obrigatório: Não

hostnameNumAppendDigits

Define o número de dígitos numéricos aleatórios que devem ser acrescentados após o valor do nome de host. É um parâmetro somente Linux usado em conjunto com o parâmetro `hostname`. Será ignorado se `hostname` não for fornecido.

Tipo: String

Valores permitidos: de 1 a 5

Obrigatório: Não

Exemplos

Para ver exemplos, consulte Associar uma instância do Amazon EC2 ao seuAWS Managed Microsoft AD no Guia de administração do AWS Directory Service.

`aws:downloadContent`

(Esquema versão 2.0 ou posterior) Baixe os documentos e scripts do SSM de locais remotos. Não há suporte a repositórios do GitHub Enterprise. Este plugin é compatível com os sistemas operacionais Linux e Windows Server.

Sintaxe

Esquema 2.2

YAML

```
---
schemaVersion: '2.2'
```

```
description: aws:downloadContent
parameters:
  sourceType:
    description: "(Required) The download source."
    type: String
  sourceInfo:
    description: "(Required) The information required to retrieve the content from
      the required source."
    type: StringMap
mainSteps:
- action: aws:downloadContent
  name: downloadContent
  inputs:
    sourceType: "{{ sourceType }}"
    sourceInfo: "{{ sourceInfo }}"
```

JSON

```
{
  "schemaVersion": "2.2",
  "description": "aws:downloadContent",
  "parameters": {
    "sourceType": {
    "description": "(Required) The download source.",
    "type": "String"
  },
  "sourceInfo": {
    "description": "(Required) The information required to retrieve the content from
 the required source.",
    "type": "StringMap"
    }
  },
  "mainSteps": [
    {
      "action": "aws:downloadContent",
      "name": "downloadContent",
      "inputs": {
        "sourceType":"{{ sourceType }}",
        "sourceInfo":"{{ sourceInfo }}"
      }
    }
  ]
}
```

Entradas

sourceType

A origem do download. O Systems Manager é compatível com os tipos de origem a seguir para baixar scripts e documentos SSM: `GitHub`, `Git`, `HTTP`, `S3` e `SSMDocument`.

Tipo: string

Obrigatório: Sim

sourceInfo

As informações necessárias para recuperar o conteúdo da fonte necessária.

Tipo: StringMap

Obrigatório: Sim

Para sourceType **`GitHub,`**, especifique o seguinte:

- owner: o proprietário do repositório.
- repository: o nome do repositório.
- path: o caminho para o arquivo ou diretório do qual fazer download.
- getOptions: opções extras para recuperar conteúdo de uma ramificação diferente da principal ou de uma confirmação específica no repositório. O getOptions poderá ser omitido se você estiver usando a confirmação mais recente na ramificação principal. Se seu repositório foi criado após 1º de outubro de 2020, a ramificação padrão pode ser nomeada principal em vez de mestre. Nesse caso, você precisará especificar valores para o parâmetro GetOptions.

  Esse parâmetro usa o seguinte formato:

  - branch:refs/heads/*`branch_name`*

    O padrão é `master`.

    Para especificar uma ramificação não padrão, use o seguinte formato:

    branch:refs/heads/*`branch_name`*
  - commitID:*`commitID`*

    O padrão é `head`.

Para usar a versão do documento do SSM em uma confirmação diferente da mais recente, especifique o ID completo da confirmação. Por exemplo:

```
"getOptions": "commitID:bbc1ddb94...b76d3bEXAMPLE",
```

- tokenInfo: o parâmetro do Systems Manager (um parâmetro SecureString) em que você armazena as informações de token de acesso do GitHub no formato `{{ssm-secure:secure-string-token-name}}`.

> **Note**
>
> O campo `tokenInfo` é o único campo do plugin de documento SSM que é compatível com um parâmetro SecureString. Os parâmetros SecureString não são compatíveis com nenhum outro campo, nem com outros plugins de documentos SSM.

```
{
    "owner":"TestUser",
    "repository":"GitHubTest",
    "path":"scripts/python/test-script",
    "getOptions":"branch:master",
    "tokenInfo":"{{ssm-secure:secure-string-token}}"
}
```

Para sourceType **`Git`**, especifique o seguinte:

- repositório

  O URL do repositório Git do arquivo ou diretório que você deseja baixar.

  Tipo: String

Além disso, é possível especificar qualquer um dos seguintes parâmetros opcionais:

- getOptions

  Opções extras para recuperar conteúdo de uma ramificação diferente da principal ou de uma confirmação específica no repositório. O getOptions poderá ser omitido se você estiver usando a confirmação mais recente na ramificação principal.

  Tipo: String

Esse parâmetro usa o seguinte formato:

- branch:refs/heads/*branch_name*

  O padrão é `master`.

  `"branch"` é necessário somente se o documento do SSM estiver armazenado em uma ramificação diferente de `master`. Por exemplo:

  ```
  "getOptions": "branch:refs/heads/main"
  ```

- commitID:*commitID*

  O padrão é `head`.

  Para usar a versão do documento do SSM em uma confirmação diferente da mais recente, especifique o ID completo da confirmação. Por exemplo:

  ```
  "getOptions": "commitID:bbc1ddb94...b76d3bEXAMPLE",
  ```

- privateSSHKey

  A chave SSH a ser usada ao se conectar a`repository`que você especificar. Você pode usar este formato para referencair um parâmetro `SecureString` para o valor da sua chave SSH: `{{ssm-secure:`*your-secure-string-parameter*`}}`.

  Tipo: String

- skipHostKeyChecking

  Determina o valor da opção StrictHostKeyChecking quando se conecta ao `repository` que você especificar. O valor padrão é `false`.

  Tipo: booliano

- username

  O nome de usuário a ser usado ao se conectar a`repository`você especifica usando HTTP. Você pode usar este formato para referenciar um parâmetro `SecureString` para o valor do seu nome de usuário: `{{ssm-secure:`*your-secure-string-parameter*`}}`.

  Tipo: String

- password

  A senha a ser usada ao se conectar aorepositoryvocê especifica usando HTTP. Você pode usar este formato para referencair um parâmetro `SecureString` para o valor da sua senha: `{{ssm-secure:your-secure-string-parameter}}`.

  Tipo: String

Para sourceType **`HTTP`**, especifique o seguinte:

- url

  path: o URL do arquivo ou diretório do qual você deseja fazer download do .

  Tipo: String

Além disso, é possível especificar qualquer um dos seguintes parâmetros opcionais:

- allowInsecureDownload

  Determina se um download pode ser realizado em uma conexão que não esteja criptografada com o Security Socket Layer (SSL) ou o Transport Layer Security (TLS). O valor padrão é `false`. Não recomendamos realizar downloads sem criptografia. Se você optar por fazê-lo, você assume todos os riscos associados. A segurança é uma responsabilidade compartilhada entre a AWS e você. Isso é descrito como o modelo de responsabilidade compartilhada da. Para saber mais, consulteModelo de responsabilidade compartilhada.

  Tipo: booliano

- authMethod

  Determina se um nome de usuário e senha são usados para autenticação quando se conectar à `url` que você especificar. Se você especificar`Basic`ou`Digest`, você deve fornecer valores para o`username`e`password`parâmetros. Para usar a`Digest`Método do,SSM AgentA versão 3.0.1181.0 ou posterior deve ser instalada na sua instância. O`Digest`suporta criptografia MD5 e SHA256.

  Tipo: String

  Valores válidos: `None` | `Basic` | `Digest`

- username

O nome de usuário a ser usado ao se conectar a`ourl`você especificar usando`Basic`autenticação. Você pode usar este formato para referenciar um parâmetro `SecureString` para o valor do seu nome de usuário: {{ssm-secure:*your-secure-string-parameter*}}.

Tipo: String

- password

A senha a ser usada ao se conectar a`ourl`você especificar usando`Basic`autenticação. Você pode usar este formato para referencair um parâmetro `SecureString` para o valor da sua senha: {{ssm-secure:*your-secure-string-parameter*}}.

Tipo: String

Para sourceType **S3**, especifique o seguinte:

- path: o URL para o arquivo ou diretório do qual você deve baixar no Amazon S3.

```
{
    "path": "https://s3.amazonaws.com/amzn-s3-demo-bucket/powershell/
helloPowershell.ps1"
}
```

Para sourceType **SSMDocument**, especifique um dos seguintes:

- name: o nome e a versão do documento no seguinte formato: `name:version`. A versão é opcional.

```
{
    "name": "Example-RunPowerShellScript:3"
}
```

- name: o ARN do documento no seguinte formato: `arn:aws:ssm:`*region*`:`*account_id*`:document/`*document_name*

```
{
   "name":"arn:aws:ssm:us-east-2:3344556677:document/MySharedDoc"
}
```

destinationPath

Um caminho local opcional na instância em que você deseja fazer download do arquivo. Se você não especificar um caminho, o conteúdo será obtido por download em um caminho relativo ao ID do comando.

Tipo: String

Obrigatório: Não

`aws:psModule`

Instalar módulos do PowerShell em uma instância do Amazon EC2. Este plugin é executado somente em sistemas operacionais Windows Server.

Sintaxe

Esquema 2.2

YAML

```
---
schemaVersion: '2.2'
description: aws:psModule
parameters:
  source:
    description: "(Required) The URL or local path on the instance to the
 application
      .zip file."
    type: String
mainSteps:
- action: aws:psModule
  name: psModule
  inputs:
    source: "{{ source }}"
```

JSON

```
{
  "schemaVersion": "2.2",
  "description": "aws:psModule",
  "parameters": {
    "source": {
```

```
      "description": "(Required) The URL or local path on the instance to the
 application .zip file.",
      "type": "String"
    }
  },
  "mainSteps": [
    {
      "action": "aws:psModule",
      "name": "psModule",
      "inputs": {
        "source": "{{ source }}"
      }
    }
  ]
}
```

Esquema 1.2

YAML

```
---
runtimeConfig:
  aws:psModule:
    properties:
    - runCommand: "{{ commands }}"
      source: "{{ source }}"
      sourceHash: "{{ sourceHash }}"
      workingDirectory: "{{ workingDirectory }}"
      timeoutSeconds: "{{ executionTimeout }}"
```

JSON

```
{
   "runtimeConfig":{
      "aws:psModule":{
         "properties":[
            {
               "runCommand":"{{ commands }}",
               "source":"{{ source }}",
               "sourceHash":"{{ sourceHash }}",
               "workingDirectory":"{{ workingDirectory }}",
               "timeoutSeconds":"{{ executionTimeout }}"
```

```
            }
         ]
      }
   }
}
```

Propriedades

runCommand

O comando do PowerShell a ser executado após a instalação do módulo.

Tipo: StringList

Obrigatório: Não

origem

O URL ou o caminho local na instância para o arquivo `.zip` do aplicativo.

Tipo: string

Obrigatório: Sim

sourceHash

O hash SHA256 do arquivo `.zip`.

Tipo: String

Obrigatório: Não

timeoutSeconds

O tempo em segundos para um comando ser concluído antes de ser considerado como tendo falhado.

Tipo: String

Obrigatório: Não

workingDirectory

O caminho para o diretório de trabalho em sua instância.

Tipo: String

Obrigatório: Não

`aws:refreshAssociation`

(Esquema versão 2.0 ou posterior) Atualizar (forçar aplicação) uma associação sob demanda. Essa ação alterará o estado do sistema com base no que é definido na associação selecionada ou em todas as associações vinculadas aos destinos. Esse plugin é executado em sistemas operacionais Linux e Microsoft Windows Server.

Sintaxe

Esquema 2.2

YAML

```
---
schemaVersion: '2.2'
description: aws:refreshAssociation
parameters:
  associationIds:
    description: "(Optional) List of association IDs. If empty, all associations
 bound
      to the specified target are applied."
    type: StringList
mainSteps:
- action: aws:refreshAssociation
  name: refreshAssociation
  inputs:
    associationIds:
    - "{{ associationIds }}"
```

JSON

```
{
  "schemaVersion": "2.2",
  "description": "aws:refreshAssociation",
  "parameters": {
    "associationIds": {
      "description": "(Optional) List of association IDs. If empty, all associations
 bound to the specified target are applied.",
      "type": "StringList"
    }
```

```
  },
  "mainSteps": [
    {
      "action": "aws:refreshAssociation",
      "name": "refreshAssociation",
      "inputs": {
        "associationIds": [
          "{{ associationIds }}"
        ]
      }
    }
  ]
}
```

Entradas

associationIds

Lista de IDs de associação. Se estiver vazia, todas as associações vinculadas ao destino especificado serão aplicadas.

Tipo: StringList

Obrigatório: Não

`aws:runDockerAction`

(Esquema versão 2.0 ou posterior) Executar ações do Docker em contêineres. Esse plugin é executado em sistemas operacionais Linux e Microsoft Windows Server.

Sintaxe

Esquema 2.2

YAML

```
---
mainSteps:
- action: aws:runDockerAction
  name: RunDockerAction
  inputs:
    action: "{{ action }}"
```

```
    container: "{{ container }}"
    image: "{{ image }}"
    memory: "{{ memory }}"
    cpuShares: "{{ cpuShares }}"
    volume: "{{ volume }}"
    cmd: "{{ cmd }}"
    env: "{{ env }}"
    user: "{{ user }}"
    publish: "{{ publish }}"
    workingDirectory: "{{ workingDirectory }}"
    timeoutSeconds: "{{ timeoutSeconds }}"
```

JSON

```
{
   "mainSteps":[
      {
         "action":"aws:runDockerAction",
         "name":"RunDockerAction",
         "inputs":{
            "action":"{{ action }}",
            "container":"{{ container }}",
            "image":"{{ image }}",
            "memory":"{{ memory }}",
            "cpuShares":"{{ cpuShares }}",
            "volume":"{{ volume }}",
            "cmd":"{{ cmd }}",
            "env":"{{ env }}",
            "user":"{{ user }}",
            "publish":"{{ publish }}",
            "workingDirectory": "{{ workingDirectory }}",
            "timeoutSeconds": "{{ timeoutSeconds }}"
         }
      }
   ]
}
```

Entradas

action

O tipo de ação a ser executada.

Tipo: string

Obrigatório: Sim

container

O ID do contêiner do Docker.

Tipo: String

Obrigatório: Não

image

O nome da imagem do Docker.

Tipo: String

Obrigatório: Não

cmd

O comando do contêiner.

Tipo: String

Obrigatório: Não

memória

O limite de memória do contêiner.

Tipo: String

Obrigatório: Não

cpuShares

Os compartilhamentos da CPU do contêiner (peso relativo).

Tipo: String

Obrigatório: Não

volume

O volume em que o contêiner é montado.

Tipo: StringList

Obrigatório: Não

env

As variáveis de ambiente do contêiner.

Tipo: String

Obrigatório: Não

usuário

O nome de usuário do contêiner.

Tipo: String

Obrigatório: Não

publicar

As portas do contêiner publicadas.

Tipo: String

Obrigatório: Não

workingDirectory

O caminho para o diretório de trabalho em seu nó gerenciado.

Tipo: String

Obrigatório: Não

timeoutSeconds

O tempo em segundos para um comando ser concluído antes de ser considerado como tendo falhado.

Tipo: String

Obrigatório: Não

`aws:runDocument`

(Esquema versão 2.0 ou posterior) Executa documentos do SSM armazenados no Systems Manager ou em um compartilhamento local. Você pode usar esse plugin com o plugin

`aws:downloadContent` para baixar um documento do SSM de um local remoto para um compartilhamento local e em seguida executá-lo. Este plugin é compatível com os sistemas operacionais Linux e Windows Server. Este plugin não suporta a execução do documento `AWS-UpdateSSMAgent` ou de qualquer documento que use o plugin `aws:updateSsmAgent`.

Sintaxe

Esquema 2.2

YAML

```
---
schemaVersion: '2.2'
description: aws:runDocument
parameters:
  documentType:
    description: "(Required) The document type to run."
    type: String
    allowedValues:
    - LocalPath
    - SSMDocument
mainSteps:
- action: aws:runDocument
  name: runDocument
  inputs:
    documentType: "{{ documentType }}"
```

JSON

```
{
  "schemaVersion": "2.2",
  "description": "aws:runDocument",
  "parameters": {
    "documentType": {
      "description": "(Required) The document type to run.",
      "type": "String",
      "allowedValues": [
        "LocalPath",
        "SSMDocument"
      ]
    }
  },
  "mainSteps": [
```

```
    {
      "action": "aws:runDocument",
      "name": "runDocument",
      "inputs": {
        "documentType": "{{ documentType }}"
      }
    }
  ]
}
```

Entradas

documentType

O tipo de documento a ser executado. Você pode executar documentos locais (`LocalPath`) ou documentos armazenados no Systems Manager (`SSMDocument`).

Tipo: string

Obrigatório: Sim

documentPath

O caminho para o documento. Se `documentType` for `LocalPath`, especifique o caminho para o documento no compartilhamento local. Se `documentType` for `SSMDocument`, especifique o nome do documento.

Tipo: String

Obrigatório: Não

documentParameters

Os parâmetros para o documento.

Tipo: StringMap

Obrigatório: Não

`aws:runPowerShellScript`

Executar scripts PowerShell ou especificar o caminho para um script a ser executado. Esse plugin é executado em sistemas operacionais Microsoft Windows Server e Linux.

Sintaxe

Esquema 2.2

YAML

```
---
schemaVersion: '2.2'
description: aws:runPowerShellScript
parameters:
  commands:
    type: String
    description: "(Required) The commands to run or the path to an existing script
      on the instance."
    default: Write-Host "Hello World"
mainSteps:
- action: aws:runPowerShellScript
  name: runPowerShellScript
  inputs:
    timeoutSeconds: '60'
    runCommand:
    - "{{ commands }}"
```

JSON

```
{
  "schemaVersion": "2.2",
  "description": "aws:runPowerShellScript",
  "parameters": {
    "commands": {
      "type": "String",
      "description": "(Required) The commands to run or the path to an existing
 script on the instance.",
      "default": "Write-Host \"Hello World\""
    }
  },
  "mainSteps": [
    {
      "action": "aws:runPowerShellScript",
      "name": "runPowerShellScript",
      "inputs": {
        "timeoutSeconds": "60",
        "runCommand": [
```

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

Esquema 1.2

YAML

```
---
runtimeConfig:
  aws:runPowerShellScript:
    properties:
    - id: 0.aws:runPowerShellScript
      runCommand: "{{ commands }}"
      workingDirectory: "{{ workingDirectory }}"
      timeoutSeconds: "{{ executionTimeout }}"
```

JSON

```
{
   "runtimeConfig":{
      "aws:runPowerShellScript":{
         "properties":[
            {
               "id":"0.aws:runPowerShellScript",
               "runCommand":"{{ commands }}",
               "workingDirectory":"{{ workingDirectory }}",
               "timeoutSeconds":"{{ executionTimeout }}"
            }
         ]
      }
   }
}
```

Propriedades

runCommand

Especifique os comandos a serem executados ou o caminho para um script existente na instância.

Tipo: StringList

Obrigatório: Sim

timeoutSeconds

O tempo em segundos para um comando ser concluído antes de ser considerado como tendo falhado. Quando o tempo limite é atingido, o Systems Manager interrompe a execução do comando.

Tipo: String

Obrigatório: Não

workingDirectory

O caminho para o diretório de trabalho em sua instância.

Tipo: String

Obrigatório: Não

`aws:runShellScript`

Executar scripts shell do Linux ou especificar o caminho para um script a ser executado. Esse plugin só é executado em sistemas operacionais Linux.

Sintaxe

Esquema 2.2

YAML

```
---
schemaVersion: '2.2'
```

```
description: aws:runShellScript
parameters:
  commands:
    type: String
    description: "(Required) The commands to run or the path to an existing script
      on the instance."
    default: echo Hello World
mainSteps:
- action: aws:runShellScript
  name: runShellScript
  inputs:
    timeoutSeconds: '60'
    runCommand:
    - "{{ commands }}"
```

JSON

```
{
  "schemaVersion": "2.2",
  "description": "aws:runShellScript",
  "parameters": {
    "commands": {
      "type": "String",
      "description": "(Required) The commands to run or the path to an existing
 script on the instance.",
      "default": "echo Hello World"
    }
  },
  "mainSteps": [
    {
      "action": "aws:runShellScript",
      "name": "runShellScript",
      "inputs": {
        "timeoutSeconds": "60",
        "runCommand": [
          "{{ commands }}"
        ]
      }
    }
  ]
}
```

Esquema 1.2

YAML

```
---
runtimeConfig:
  aws:runShellScript:
    properties:
    - runCommand: "{{ commands }}"
      workingDirectory: "{{ workingDirectory }}"
      timeoutSeconds: "{{ executionTimeout }}"
```

JSON

```
{
   "runtimeConfig":{
      "aws:runShellScript":{
         "properties":[
            {
               "runCommand":"{{ commands }}",
               "workingDirectory":"{{ workingDirectory }}",
               "timeoutSeconds":"{{ executionTimeout }}"
            }
         ]
      }
   }
}
```

Propriedades

runCommand

Especifique os comandos a serem executados ou o caminho para um script existente na instância.

Tipo: StringList

Obrigatório: Sim

timeoutSeconds

O tempo em segundos para um comando ser concluído antes de ser considerado como tendo falhado. Quando o tempo limite é atingido, o Systems Manager interrompe a execução do comando.

Tipo: String

Obrigatório: Não

workingDirectory

O caminho para o diretório de trabalho em sua instância.

Tipo: String

Obrigatório: Não

`aws:softwareInventory`

(Esquema versão 2.0 ou posterior) Coletar metadados sobre aplicativos, arquivos e configurações em suas instâncias gerenciadas. Esse plugin é executado em sistemas operacionais Linux e Microsoft Windows Server. Quando você configura a coleta de inventário, primeiro cria uma associação do AWS Systems Manager State Manager. O coleta os dados de inventário quando a associação é executada. Se você não criar a associação primeiro e tentar invocar o plugin aws:softwareInventory usando, por exemplo, o `aws:softwareInventory`, o sistema retornará o seguinte erro:

```
The aws:softwareInventory plugin can only be invoked via ssm-associate.
```

Uma instância pode ter apenas uma associação de inventário configurada por vez. Se você configurar uma instância com duas ou mais associações, a associação de inventário não será executada, e os dados de inventário não serão coletados. Para obter mais informações sobre como coletar o inventário, consulte AWS Systems Manager Inventory.

Sintaxe

Esquema 2.2

YAML

```
---
```

```
mainSteps:
- action: aws:softwareInventory
  name: collectSoftwareInventoryItems
  inputs:
    applications: "{{ applications }}"
    awsComponents: "{{ awsComponents }}"
    networkConfig: "{{ networkConfig }}"
    files: "{{ files }}"
    services: "{{ services }}"
    windowsRoles: "{{ windowsRoles }}"
    windowsRegistry: "{{ windowsRegistry}}"
    windowsUpdates: "{{ windowsUpdates }}"
    instanceDetailedInformation: "{{ instanceDetailedInformation }}"
    customInventory: "{{ customInventory }}"
```

JSON

```
{
   "mainSteps":[
      {
         "action":"aws:softwareInventory",
         "name":"collectSoftwareInventoryItems",
         "inputs":{
            "applications":"{{ applications }}",
            "awsComponents":"{{ awsComponents }}",
            "networkConfig":"{{ networkConfig }}",
            "files":"{{ files }}",
            "services":"{{ services }}",
            "windowsRoles":"{{ windowsRoles }}",
            "windowsRegistry":"{{ windowsRegistry}}",
            "windowsUpdates":"{{ windowsUpdates }}",
            "instanceDetailedInformation":"{{ instanceDetailedInformation }}",
            "customInventory":"{{ customInventory }}"
         }
      }
   ]
}
```

Entradas

aplicações

(Opcional) Coletar metadados para aplicativos instalados.

Tipo: String

Obrigatório: Não

awsComponents

(Opcional) Colete metadados para componentes da AWS, como o amazon-ssm-agent.

Tipo: String

Obrigatório: Não

files

(Opcional, requer o SSM Agent versão 2.2.64.0 ou posterior) Colete metadados para arquivos, incluindo, por exemplo, nomes de arquivos, a hora em que os arquivos foram criados, a hora em que os arquivos foram modificados e acessados pela última vez e os tamanhos dos arquivos. Para obter mais informações sobre como coletar inventário de arquivos, consulte Trabalhar com o inventário de arquivos e do Registro do Windows.

Tipo: String

Obrigatório: Não

networkConfig

(Opcional) Coletar metadados para configurações de rede.

Tipo: String

Obrigatório: Não

billingInfo

(Opcional) Colete metadados para obter detalhes da plataforma associados ao código de cobrança da AMI.

Tipo: String

Obrigatório: Não

windowsUpdates

(Opcional) Coletar metadados para todas as atualizações do Windows.

Tipo: String

Obrigatório: Não

instanceDetailedInformation

(Opcional) Colete mais informações da instância do que as fornecidas pelo plugin do inventário padrão (`aws:instanceInformation`), incluindo o modelo da CPU, a velocidade e o número de núcleos.

Tipo: String

Obrigatório: Não

serviços

(Opcional, somente o SO Windows, requer o SSM Agent versão 2.2.64.0 ou posterior) Colete metadados para configurações de serviço.

Tipo: String

Obrigatório: Não

windowsRegistry

(Opcional, somente o SO Windows, requer o SSM Agent versão 2.2.64.0 ou posterior) Colete chaves e valores do Registro do Windows. Você pode escolher um caminho de chaves e coletar todos os valores e chaves recursivamente. Você pode também coletar determinada chave de registro e o respectivo valor para um caminho específico. O inventário coleta o caminho da chave, o nome, o tipo e o valor. Para obter mais informações sobre como coletar inventário do Registro do Windows, consulte Trabalhar com o inventário de arquivos e do Registro do Windows.

Tipo: String

Obrigatório: Não

windowsRoles

(Opcional, somente o SO Windows, requer o SSM Agent versão 2.2.64.0 ou posterior) Colete metadados para configurações de função do Microsoft Windows.

Tipo: String

Obrigatório: Não

customInventory

(Opcional) Coletar dados de inventário personalizado. Para obter mais informações sobre inventário personalizado, consulte Trabalhar com inventário personalizado

Tipo: String

Obrigatório: Não

customInventoryDirectory

(Opcional) Colete dados de inventário personalizados do diretório especificado. Para obter mais informações sobre inventário personalizado, consulte Trabalhar com inventário personalizado

Tipo: String

Obrigatório: Não

`aws:updateAgent`

Atualizar o serviço EC2Config para a versão mais recente ou especificar uma versão mais antiga. Este plugin executado somente em sistemas operacionais Microsoft Windows Server. Para obter mais informações sobre o serviço EC2Config, consulte Configurar uma instância do Windows usando o serviço EC2Config (legado) no Guia de usuário do Amazon EC2.

Sintaxe

Esquema 2.2

YAML

```
---
schemaVersion: '2.2'
description: aws:updateAgent
mainSteps:
- action: aws:updateAgent
  name: updateAgent
  inputs:
```

```
    agentName: Ec2Config
    source: https://s3.{Region}.amazonaws.com/aws-ssm-{Region}/manifest.json
```

JSON

```
{
  "schemaVersion": "2.2",
  "description": "aws:updateAgent",
  "mainSteps": [
    {
      "action": "aws:updateAgent",
      "name": "updateAgent",
      "inputs": {
        "agentName": "Ec2Config",
        "source": "https://s3.{Region}.amazonaws.com/aws-ssm-{Region}/manifest.json"
      }
    }
  ]
}
```

Esquema 1.2

YAML

```
---
runtimeConfig:
  aws:updateAgent:
    properties:
      agentName: Ec2Config
      source: https://s3.{Region}.amazonaws.com/aws-ssm-{Region}/manifest.json
      allowDowngrade: "{{ allowDowngrade }}"
      targetVersion: "{{ version }}"
```

JSON

```
{
   "runtimeConfig":{
      "aws:updateAgent":{
         "properties":{
            "agentName":"Ec2Config",
            "source":"https://s3.{Region}.amazonaws.com/aws-ssm-{Region}/
manifest.json",
```

```
            "allowDowngrade":"{{ allowDowngrade }}",
            "targetVersion":"{{ version }}"
         }
      }
   }
}
```

Propriedades

agentName

EC2Config. Esse é o nome do agente que executa o serviço EC2Config.

Tipo: string

Obrigatório: Sim

allowDowngrade

Permitir que o serviço EC2Config seja atualizado para uma versão anterior. Se definido como falso, o serviço poderá ser atualizado apenas para versões mais novas (padrão). Se definido como verdadeiro, especifique a versão anterior.

Tipo: booliano

Obrigatório: Não

origem

O local em que o Systems Manager copia a versão do EC2Config a ser instalada. Não é possível mudar esse local.

Tipo: string

Obrigatório: Sim

targetVersion

Uma versão específica do serviço EC2Config a ser instalado. Se não for especificado, o serviço será atualizado para a versão mais recente.

Tipo: String

Obrigatório: Não

`aws:updateSsmAgent`

Atualizar o SSM Agent para a versão mais recente ou especificar uma versão mais antiga. Esse plugin é executado em sistemas operacionais Linux e Windows Server. Para obter mais informações, consulte Trabalhar com o SSM Agent.

Sintaxe

Esquema 2.2

YAML

```
---
schemaVersion: '2.2'
description: aws:updateSsmAgent
parameters:
  allowDowngrade:
    default: 'false'
    description: "(Optional) Allow the Amazon SSM Agent service to be downgraded to
      an earlier version. If set to false, the service can be upgraded to newer
 versions
      only (default). If set to true, specify the earlier version."
    type: String
    allowedValues:
    - 'true'
    - 'false'
mainSteps:
- action: aws:updateSsmAgent
  name: updateSSMAgent
  inputs:
    agentName: amazon-ssm-agent
    source: https://s3.{Region}.amazonaws.com/amazon-ssm-{Region}/ssm-agent-
manifest.json
    allowDowngrade: "{{ allowDowngrade }}"
```

JSON

```
{
  "schemaVersion": "2.2",
  "description": "aws:updateSsmAgent",
```

```
  "parameters": {
    "allowDowngrade": {
      "default": "false",
      "description": "(Required) Allow the Amazon SSM Agent service to be downgraded
 to an earlier version. If set to false, the service can be upgraded to newer
 versions only (default). If set to true, specify the earlier version.",
      "type": "String",
      "allowedValues": [
        "true",
        "false"
      ]
    }
  },
  "mainSteps": [
    {
      "action": "aws:updateSsmAgent",
      "name": "awsupdateSsmAgent",
      "inputs": {
        "agentName": "amazon-ssm-agent",
        "source": "https://s3.{Region}.amazonaws.com/amazon-ssm-{Region}/ssm-agent-
manifest.json",
        "allowDowngrade": "{{ allowDowngrade }}"
      }
    }
  ]
}
```

Esquema 1.2

YAML

```
---
runtimeConfig:
  aws:updateSsmAgent:
    properties:
    - agentName: amazon-ssm-agent
      source: https://s3.{Region}.amazonaws.com/aws-ssm-{Region}/manifest.json
      allowDowngrade: "{{ allowDowngrade }}"
```

JSON

```
{
```

```
   "runtimeConfig":{
      "aws:updateSsmAgent":{
         "properties":[
            {
               "agentName":"amazon-ssm-agent",
               "source":"https://s3.{Region}.amazonaws.com/aws-ssm-{Region}/
manifest.json",
               "allowDowngrade":"{{ allowDowngrade }}"
            }
         ]
      }
   }
}
```

Propriedades

agentName

amazon-ssm-agent. Esse é o nome do Systems Manager Agent que processa solicitações e executa comandos na instância.

Tipo: string

Obrigatório: Sim

allowDowngrade

Permitir que o SSM Agent seja atualizado para uma versão anterior. Se definido como falso, o agente poderá ser atualizado apenas para versões mais novas (padrão). Se definido como verdadeiro, especifique a versão anterior.

Tipo: booliano

Obrigatório: Sim

origem

O local em que o Systems Manager copia a versão do SSM Agent a ser instalada. Não é possível mudar esse local.

Tipo: string

Obrigatório: Sim

targetVersion

Uma versão específica do SSM Agent a ser instalada. Se não for especificado, o agente será atualizado para a versão mais recente.

Tipo: String

Obrigatório: Não

Criar conteúdo de documento do SSM

Se os documentos públicos do AWS Systems Manager, não executarem todas as ações que você deseja executar em seus recursos da AWS, você poderá criar seus próprios documentos do SSM. Você também pode clonar documentos do SSM usando o console do. A clonagem de documentos copia o conteúdo de um documento existente em um novo documento que você pode modificar. Ao criar ou clonar um documento, o conteúdo do documento não deve exceder 64 KB. Essa cota também inclui o conteúdo especificado para parâmetros de entrada em runtime. Ao criar um novo `Command` ou documento de `Policy`, recomendamos que você use o esquema versão 2.2 ou posterior para que possa aproveitar os recursos mais recentes, como edição de documentos, versionamento automático, sequenciamento e muito mais.

Escrever conteúdo de documento do

Para criar seu próprio conteúdo de documentos do , é importante entender os diferentes esquemas, recursos, plugins e sintaxe disponíveis para documentos do . Recomendamos familiarizar-se com os recursos a seguir.

- Escrever seus próprios documentos do AWS Systems Manager
- Elementos e parâmetros de dados
- Esquemas, atributos e exemplos
- Referência de plug-ins de documentos de comando
- Referência de ações do Systems Manager Automation
- Variáveis de sistema de automação
- Exemplos adicionais de runbook
- Trabalhar com runbooks do Automation do Systems Manager usando o AWS Toolkit for Visual Studio Code

- Experiência de design visual para runbooks de automação
- Uso de scripts em runbooks

Documentos SSM predefinidos da AWS podem executar algumas das ações exigidas. Você pode chamar esses documentos usando os plugins `aws:runDocument`, `aws:runCommand` ou `aws:executeAutomation` em seu documento SSM personalizado, dependendo do tipo de documento. Também é possível copiar partes desses documentos em um documento SSM personalizado e editar o conteúdo para atender às suas necessidades.

Tip

Ao criar conteúdo de documentos SSM, é possível alterar o conteúdo e atualizar o documento SSM várias vezes durante o teste. Os comandos a seguir atualizam o documento do com o conteúdo mais recente e atualizam a versão padrão do documento para a versão mais recente do documento.

Note

Os comandos do Linux e do Windows usam a ferramenta de linha de comando `jq` para filtrar os dados da resposta JSON.

Linux & macOS

```
latestDocVersion=$(aws ssm update-document \
    --content file://path/to/file/documentContent.json \
    --name "ExampleDocument" \
    --document-format JSON \
    --document-version '$LATEST' \
    | jq -r '.DocumentDescription.LatestVersion')

aws ssm update-document-default-version \
    --name "ExampleDocument" \
    --document-version $latestDocVersion
```

Windows

```
latestDocVersion=$(aws ssm update-document ^
    --content file://C:\path\to\file\documentContent.json ^
```

```
    --name "ExampleDocument" ^
    --document-format JSON ^
    --document-version "$LATEST" ^
    | jq -r '.DocumentDescription.LatestVersion')

aws ssm update-document-default-version ^
    --name "ExampleDocument" ^
    --document-version $latestDocVersion
```

PowerShell

```
$content = Get-Content -Path "C:\path\to\file\documentContent.json" | Out-
String
$latestDocVersion = Update-SSMDocument `
    -Content $content `
    -Name "ExampleDocument" `
    -DocumentFormat "JSON" `
    -DocumentVersion '$LATEST' `
    | Select-Object -ExpandProperty LatestVersion

Update-SSMDocumentDefaultVersion `
    -Name "ExampleDocument" `
    -DocumentVersion $latestDocVersion
```

Práticas recomendadas seguras para documentos do SSM

Ao criar documentos do SSM, siga estas práticas recomendadas de segurança para ajudar a evitar a injeção de comandos e garantir o tratamento seguro de parâmetros:

- Use a interpolação de variáveis de ambiente para parâmetros de string que serão usados em comandos ou scripts. Adicione a propriedade `interpolationType` com valor `ENV_VAR` aos seus parâmetros de string:

```
{
    "command": {
        "type": "String",
        "description": "Command to execute",
        "interpolationType": "ENV_VAR"
    }
}
```

Você pode melhorar ainda mais a segurança de seus documentos do SSM especificando que aspas duplas não são aceitas nos valores fornecidos por interpolação:

```
{
    "command": {
        "type": "String",
        "description": "Command to execute",
        "interpolationType": "ENV_VAR",
            "allowedPattern": "^[^"]*$"
    }
}
```

- Ao usar linguagens interpretadas, como Python, Ruby ou Node.js, faça referência aos parâmetros usando a sintaxe apropriada da variável de ambiente:

```
# Python example
import os
command = os.environ['SSM_Message']
```

- Para compatibilidade com as versões anteriores do SSM Agent (anteriores à versão 3.3.2746.0), inclua a lógica de fallback para variáveis de ambiente:

```
if [ -z "${SSM_command+x}" ]; then
    export SSM_command="{{command}}"
fi
```

- Combine a interpolação de variáveis de ambiente com `allowedPattern` para validação de entrada adicional. No exemplo abaixo, o valor `^[^"]*$` de `allowedPattern` evita especificamente aspas duplas no valor da string:

```
{
    "command": {
        "type": "String",
        "interpolationType": "ENV_VAR",
        "allowedPattern": "^[a-zA-Z0-9_-]+$"
    }
}
```

- Antes de implementar seu documento do SSM, verifique as seguintes considerações de segurança:

- Todos os parâmetros de string que aceitam a entrada do usuário usam a interpolação de variáveis de ambiente quando apropriado.
- A validação de entrada é implementada usando `allowedPattern` sempre que possível.
- O documento inclui o tratamento adequado de erros para o processamento de parâmetros.
- A compatibilidade com versões anteriores é mantida para ambientes que usam as versões anteriores do SSM Agent.

Clonar um documento do SSM

Você pode clonar documentos do AWS Systems Manager usando o console de documentos do Systems Manager para criar documentos do SSM. A clonagem de documentos do SSM copia o conteúdo de um documento existente em um novo documento que você pode modificar. Não é possível clonar um documento de mais de 64 KB.

Para clonar um documento do SSM

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Documents.
3. Na caixa de pesquisa, insira o nome do documento que você deseja clonar.
4. Selecione o nome do documento que você quiser clonar e escolha Clone document (Clonar documento) na lista suspensa Actions (Ações).
5. Modifique o documento como preferir e escolhaCriar documentoPara salvar o documento.

Depois que escrever o conteúdo do documento SSM, você poderá usar o conteúdo para criar um documento do SSM usando um dos métodos a seguir.

Criar documentos SSM

- Criar documentos compostos

Criar documentos compostos

Um documento composite do AWS Systems Manager (SSM) é um documento personalizado que realiza uma série de ações ao executar um ou mais documentos secundários do SSM. Os documentos compostos promovem a infraestrutura como código ao permitir que você crie um conjunto padrão de documentos do SSM para tarefas comuns, como inicialização de software ou

ingresso de instâncias no domínio. Depois, você pode compartilhar esses documentos em Contas da AWS da mesma Região da AWS para reduzir a manutenção e garantir a consistência do documento SSM.

Por exemplo, você pode criar um documento composto que execute as seguintes ações:

1. Instala todos os patches na lista de permissões.
2. Instala software antivírus.
3. Baixa scripts do GitHub e os executa.

Neste exemplo, seu documento inclui os seguintes plugins de seu documento personalizado do SSM para executar as seguintes ações:

1. O plug-in `aws:runDocument` para executar o documento `AWS-RunPatchBaseline`, que instala todos os patches permitidos listados.
2. O plugin `aws:runDocument` para executar o documento `AWS-InstallApplication`, que instala o software antivírus.
3. O plugin `aws:downloadContent` para baixar scripts do GitHub e executá-los.

Os documentos compostos e secundários podem ser armazenados no Systems Manager, no GitHub (repositórios públicos e privados) ou no Amazon S3. Os documentos compostos e os documentos secundários podem ser criados em JSON ou YAML.

> **Note**
>
> Os documentos compostos só podem executar em um nível máximo de três documentos. Isso significa que um documento composto pode chamar um documento filho e que o documento filho pode chamar um último documento.

Para criar um documento composta, adicione o plugin `aws:runDocument` a um documento personalizado do SSM e especifique as entradas necessárias. A seguir encontra-se um exemplo de documento composto que executa as seguintes ações:

1. Executa o plugin `aws:downloadContent` para baixar um documento do SSM de um repositório público do GitHub para um diretório local chamado bootstrap. O documento do SSM é chamado StateManagerBootstrap.yml (um documento YAML).

2. Executa o plugin `aws:runDocument` para executar o documento StateManagerBootstrap.yml. Nenhum parâmetro é especificado.
3. Executa o plugin `aws:runDocument` para executar o documento `AWS-ConfigureDocker pre-defined` do MUS. Os parâmetros especificados para instalar o docker na instância.

```
{
  "schemaVersion": "2.2",
  "description": "My composite document for bootstrapping software and installing
 Docker.",
  "parameters": {
  },
  "mainSteps": [
    {
      "action": "aws:downloadContent",
      "name": "downloadContent",
      "inputs": {
        "sourceType": "GitHub",
        "sourceInfo": "{\"owner\":\"TestUser1\",\"repository\":\"TestPublic\", \"path
\":\"documents/bootstrap/StateManagerBootstrap.yml\"}",
        "destinationPath": "bootstrap"
      }
    },
    {
      "action": "aws:runDocument",
      "name": "runDocument",
      "inputs": {
        "documentType": "LocalPath",
        "documentPath": "bootstrap",
        "documentParameters": "{}"
      }
    },
    {
      "action": "aws:runDocument",
      "name": "configureDocker",
      "inputs": {
        "documentType": "SSMDocument",
        "documentPath": "AWS-ConfigureDocker",
        "documentParameters": "{\"action\":\"Install\"}"
      }
    }
  ]
```

```
}
```

Mais informações

- Para obter mais informações sobre reinicialização dos servidores e instâncias ao usar Run Command para chamar scripts, consulte Tratamento de reinicializações ao executar comandos.
- Para obter mais informações sobre os plugins que podem ser adicionados a um documento personalizado do SSM, consulte Referência de plug-ins de documentos de comando.
- Se você simplesmente desejar executar um documento em um local remoto (sem criar um documento composto), consulte Executar documentos do em locais remotos.

Trabalhar com documentos

Esta seção contém informações sobre como usar e trabalhar com documentos do SSM.

Tópicos

- Comparar versões de documentos do SSM
- Criar um documento do SSM.
- Excluir documentos do SSM personalizados
- Executar documentos do em locais remotos
- Compartilhar documentos do Systems Manager
- Pesquisando documentos do SSM

Comparar versões de documentos do SSM

Você pode comparar as diferenças de conteúdo entre versões doAWS Systems Manager(SSM) no console Documents do Systems Manager. Ao comparar versões de um documento SSM, as diferenças entre o conteúdo das versões são realçadas.

Para comparar o conteúdo do documento SSM (console)

1. Abra o console do AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Documents.
3. Na lista de documentos, escolha o documento cujo conteúdo você deseja comparar.

4. NoConteúdo, selecioneComparar versõesE escolha a versão do documento com a qual você deseja comparar o conteúdo.

Criar um documento do SSM.

Depois que criar o conteúdo do documento SSM personalizado, conforme descrito em Escrever conteúdo de documento do , você poderá usar o console do Systems Manager para criar um documento do SSM usando seu conteúdo.

Para criar um documento do SSM

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Documents.
3. Escolha Create command or session (Criar comando ou sessão).
4. Insira um nome descritivo para o documento.
5. (Opcional) No Tipo de destino, especifique o tipo de recursos nos quais o documento pode ser executado.
6. Na lista Document type (Tipo de documento), escolha o tipo de documento que você deseja criar.
7. Exclua os parênteses no campo Content (Conteúdo) e cole o conteúdo do documento que você criou anteriormente.
8. (Opcional) Na seção Tags de documento, aplique um ou mais pares de nome/valor de chave de tag ao documento.

   Tags são metadados opcionais que você atribui a um recurso. As tags permitem categorizar um recurso de diferentes formas, como por finalidade, proprietário ou ambiente. Por exemplo, você pode querer marcar um documento para identificar o tipo de tarefas que ele executa, o tipo de sistemas operacionais que ele direciona e o ambiente em que ele é executado. Nesse caso, você pode especificar os seguintes pares de nome/valor:

   - `Key=TaskType,Value=MyConfigurationUpdate`
   - `Key=OS,Value=AMAZON_LINUX_2`
   - `Key=Environment,Value=Production`
9. Escolha Create document (Criar documento) para salvar o documento.

Excluir documentos do SSM personalizados

Caso não deseje mais usar um documento do SSM personalizado, você poderá excluí-lo usando o console do AWS Systems Manager.

Para excluir um documento do SSM

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Documents.
3. Selecione a função que você deseja excluir.
4. SelectExcluir. Quando solicitado a excluir o documento, selecioneExcluir.

Para obter exemplos do uso de ferramentas de linha de comando ou SDKs para excluir documentos do SSM, consulte Usar `DeleteDocument` com o AWS SDK ou a CLI.

Executar documentos do em locais remotos

Você pode executar documentos do AWS Systems Manager (SSM) em locais remotos usando o documento SSM predefinido `AWS-RunDocument`. Este documento suporta a execução de documentos SSM armazenados nos seguintes locais:

- Repositórios públicos e privados do GitHub (não há suporte ao GitHub Enterprise)
- Buckets do Amazon S3
- Systems Manager

Embora também seja possível executar documentos remotos usando o State Manager ou o Automation, ferramentas do AWS Systems Manager, o procedimento a seguir descreve apenas como executar documentos do SSM remotos usando o AWS Systems Manager Run Command no console do Systems Manager.

> **Note**
>
> O `AWS-RunDocument` pode ser usado para executar apenas documentos SSM do tipo de comando, não outros tipos, como runbooks de automação. O `AWS-RunDocument` usa o plugin `aws:downloadContent`. Para obter mais informações sobre o plugin `aws:downloadContent`, consulte `aws:downloadContent`.

Antes de começar

Para executar um documento remoto, você deve primeiro concluir as tarefas a seguir.

- Crie um documento do Comando SSM e salve-o em um local remoto. Para obter mais informações, consulte Criar conteúdo de documento do SSM.
- Se você planeja executar um documento remoto armazenado em um repositório privado do GitHub, é necessário criar um parâmetro `SecureString` do Systems Manager para o token de acesso de segurança do GitHub. Não é possível acessar um documento remoto em um repositório privado do GitHub transmitindo manualmente o token via SSH. O token de acesso deve ser passado como um parâmetro `SecureString` do Systems Manager. Para obter mais informações sobre como criar um parâmetro `SecureString`, consulte Criar parâmetros do Parameter Store no Systems Manager.

Executar um documento remoto (console)

Para executar um documento remoto

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Run Command.
3. Selecione Run command.
4. Na lista Document (Documento), escolha **`AWS-RunDocument`**.
5. Em Command parameters (Parâmetros de comando), em Source Type (Tipo de origem), escolha uma opção.
   - Se você escolher o GitHub, especifique as Informações da origem no seguinte formato:

```
{
    "owner": "owner_name",
    "repository": "repository_name",
    "path": "path_to_document",
    "getOptions":"branch:branch_name",
    "tokenInfo": "{{ssm-secure:secure-string-token}}"
}
```

Por exemplo:

```
{
```

```
    "owner":"TestUser",
    "repository":"GitHubTestExamples",
    "path":"scripts/python/test-script",
    "getOptions":"branch:exampleBranch",
    "tokenInfo":"{{ssm-secure:my-secure-string-token}}"
}
```

> **Note**
>
> `getOptions` são opções extras para recuperar conteúdo de uma ramificação diferente da principal ou de uma confirmação específica no repositório. O `getOptions` poderá ser omitido se você estiver usando a confirmação mais recente na ramificação principal. O parâmetro `branch` é necessário somente se o documento do SSM estiver armazenado em uma ramificação diferente de `master`.
> Para usar a versão do documento do SSM em uma confirmação específica no repositório, use `commitID` com `getOptions` em vez de `branch`. Por exemplo:
>
> ```
> "getOptions": "commitID:bbc1ddb94...b76d3bEXAMPLE",
> ```

- Se escolher S3, especifique as informações em Source Info no formato a seguir:

  ```
  {"path":"URL_to_document_in_S3"}
  ```

  Por exemplo:

  ```
  {"path":"https://s3.amazonaws.com/amzn-s3-demo-bucket/scripts/ruby/
  mySSMdoc.json"}
  ```

- Se escolher SSMDocument, especifique as informações em Source Info no formato a seguir:

  ```
  {"name": "document_name"}
  ```

  Por exemplo:

  ```
  {"name": "mySSMdoc"}
  ```

6. No campo Document Parameters (Parâmetros do documento), digite os parâmetros para o documento remoto do SSM. Por exemplo, se você executar o documento `AWS-RunPowerShell`, poderá especificar:

```
{"commands": ["date", "echo \"Hello World\""]}
```

Se você executar o documento `AWS-ConfigureAWSPack`, poderá especificar:

```
{
   "action":"Install",
   "name":"AWSPVDriver"
}
```

7. Na seção Targets (Destinos), escolha os nós gerenciados nos quais você quer executar essa operação, especificando as tags, selecionando as instâncias ou dispositivos de borda manualmente ou especificando um grupo de recursos.

> **Tip**
>
> Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

8. Para Other parameters (Outros parâmetros):

   - Em Comment (Comentário), digite as informações sobre esse comando.
   - Em Timeout (seconds) (Tempo limite [segundos]), especifique o número de segundos para o sistema aguardar até a falha de execução do comando total.

9. Para Rate control (Controle de taxa):

   - Em Concurrency (Concorrência), especifique um número ou uma porcentagem de nós gerenciados nos quais executar o comando ao mesmo tempo.

> **Note**
>
> Se você selecionou destinos especificando tags aplicadas a instâncias a nós gerenciados ou especificando grupos de recursos da AWS, e não tiver certeza de

quantas instâncias são direcionadas, restrinja o número de instâncias que poderão executar o documento ao mesmo tempo, especificando uma porcentagem.

- Em Error threshold (Limite de erro), especifique quando parar de executar o comando em outros nós depois de falhar em alguns ou em uma porcentagem de nós. Por exemplo, se você especificar três erros, o Systems Manager deixará de enviar o comando quando o 4° erro for recebido. Os nós gerenciados que continuam processando o comando também podem enviar erros.

10. (Opcional) Em Output options (Opções de saída), para salvar a saída do comando em um arquivo, selecione a caixa Write command output to an S3 bucket (Gravar saída do comando em um bucket do S3). Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

> **Note**
>
> As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil de instância (para instâncias do EC2) ou perfil de serviço do IAM (máquinas ativadas para ambientes híbridos) atribuído à instância, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço do IAM associado ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

11. Na seção SNS notifications (Notificações do SNS), se quiser enviar notificações sobre o status da execução do comando, marque a caixa de seleção Enable SNS notifications (Habilitar notificações do SNS).

    Para obter mais informações sobre a configuração de notificações do Amazon SNS para o Run Command, consulte Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS.

12. Escolha Executar.

> **Note**
>
> Para obter mais informações sobre reinicialização dos servidores e instâncias ao usar Run Command para chamar scripts, consulte Tratamento de reinicializações ao executar comandos.

Compartilhar documentos do Systems Manager

É possível compartilhar documentos do AWS Systems Manager (SSM) de forma privada ou pública com contas na mesma região da Região da AWS. Para compartilhar um documento específico, modifique as permissões do documento e permita que pessoas específicas acessem o mesmo de acordo com o ID da Conta da AWS. Para compartilhar um documento SSM publicamente, modifique as permissões do documento e especifique `All`. Não é possível compartilhar os documentos de forma pública e privada simultaneamente.

> **Warning**
>
> Use documentos SSM compartilhados apenas de fontes confiáveis. Ao usar qualquer documento compartilhado, revise cuidadosamente o conteúdo do documento antes de usá-lo para que você entenda como ele mudará a configuração da sua instância. Para obter mais informações sobre melhores práticas de documentos compartilhados, consulte Práticas recomendadas para documentos compartilhados do SSM.

Limitações

Ao começar a trabalhar com documentos do SSM, lembre-se das seguintes limitações.

- Somente o proprietário pode compartilhar um documento.
- É preciso interromper o compartilhamento de um documento antes de excluí-lo. Para obter mais informações, consulte Modificar permissões para um documento compartilhado do .
- Você pode compartilhar um documento com um máximo de 1000 contas da Contas da AWS. Você pode solicitar um aumento desse limite no Suporte Center. Em Limit type (Tipo de limite), escolha EC2 Systems Manager e descreva o motivo para a solicitação.
- Você pode compartilhar publicamente um máximo de cinco documentos do . Você pode solicitar um aumento desse limite no Suporte Center. Em Limit type (Tipo de limite), escolha EC2 Systems Manager e descreva o motivo para a solicitação.

- Os documentos podem ser compartilhados somente com outras contas na mesma Região da AWS. O compartilhamento entre regiões não é compatível.

> **Important**
>
> No Systems Manager, um documento SSM pertencente à Amazon é um documento criado e gerenciado pela própria Amazon Web Services. Os documentos pertencentes à Amazon incluem um prefixo como `AWS-*` no nome do documento. O proprietário do documento é considerado a Amazon, e não uma conta de usuário específica em AWS. Esses documentos estão disponíveis publicamente para uso de todos.

Para obter mais informações sobre as cotas de serviço do Systems Manager, consulte Service Quotas do AWS Systems Manager.

Conteúdo

- Práticas recomendadas para documentos compartilhados do SSM
- Bloquear compartilhamento público para documentos do SSM
- Compartilhar um documento do
- Modificar permissões para um documento compartilhado do
- Usar documentos compartilhados do

Práticas recomendadas para documentos compartilhados do SSM

Reveja as seguintes diretrizes antes de compartilhar ou usar um documento compartilhado.

Remover informações confidenciais

Reveja seu documento do AWS Systems Manager cuidadosamente e remova todas as informações confidenciais. Por exemplo, verifique se o documento não inclui suas credenciais da AWS. Se você compartilhar um documento com pessoas específicas, esses usuários poderão visualizar as informações no documento. Se você compartilhar um documento publicamente, qualquer pessoa poderá visualizar as informações no documento.

Bloquear compartilhamento público de documentos

Analise todos os documentos do SSM compartilhados publicamente em sua conta e confirme se você deseja continuar compartilhando-os. Para parar de compartilhar um documento com o

público, é necessário modificar a configuração de permissões do documento, conforme descrito na seção Modificar permissões para um documento compartilhado do deste tópico. Ativar a configuração de bloqueio de compartilhamento público não afeta nenhum documento que você esteja compartilhando atualmente com o público. A menos que seu caso de uso exija que você compartilhe documentos com o público, recomendamos ativar a configuração de bloqueio de compartilhamento público para seus documentos do SSM na seção Preferências do console de documentos do Systems Manager. Ativar essa configuração impede o acesso indesejado aos documentos do SSM. A configuração de compartilhamento público de bloco é uma configuração de nível de conta que pode diferir para cada Região da AWS.

Restringir ações do Run Command usando uma política de confiança do IAM

Crie uma política do AWS Identity and Access Management (IAM) restritiva para os usuários que terão acesso ao documento. A política do IAM do determina quais documento do SSM um usuário pode ver no console do Amazon Elastic Compute Cloud (Amazon EC2) ou ao chamar `ListDocuments` usando a AWS Command Line Interface (AWS CLI) ou o AWS Tools for Windows PowerShell. A política também restringe as ações que o usuário pode realizar com o documento do . Você pode criar uma política restritiva para que um usuário só possa usar documentos específicos. Para obter mais informações, consulte Exemplos de política gerenciada pelo cliente.

Ter cuidado ao usar documentos compartilhados do

Revise o conteúdo de cada documento compartilhado com você, especialmente documentos públicos, para entender os comandos que serão executados em suas instâncias. Um documento pode ter intencionalmente ou involuntariamente repercussões negativas após sua execução. Se o documento fizer referência a uma rede externa, reveja a origem externa antes de usar o documento.

Enviar comandos usando o hash do documento

Quando você compartilha um documento, o sistema cria um hash Sha-256 e o atribui ao documento. O sistema também salva um snapshot do conteúdo do documento. Quando você envia um comando usando um documento compartilhado, pode especificar o hash no seu comando para garantir que as seguintes condições sejam verdadeiras:

- Você está executando um comando no documento correto do Systems Manager
- O conteúdo do documento não mudou desde que foi compartilhado com você.

Se o hash não corresponder ao documento especificado ou se o conteúdo do documento compartilhado tiver mudado, o comando retornará uma exceção `InvalidDocument`. Observação: o hash não pode verificar o conteúdo do documento de locais externos.

Usar o parâmetro de interpolação para melhorar a segurança

Para parâmetros de tipo `String` em seus documentos do SSM, use o parâmetro e o valor `interpolationType": "ENV_VAR` para melhorar a segurança contra ataques de injeção de comando, tratando as entradas de parâmetros como literais de string em vez de comandos potencialmente executáveis. Nesse caso, o agente cria uma variável de ambiente denominada `SSM_`*`parameter-name`* com o valor do parâmetro. Recomendamos atualizar todos os seus documentos existentes do SSM que incluam parâmetros de tipo `String` para incluir `"interpolationType": "ENV_VAR"`. Para obter mais informações, consulte the section called “Escrever conteúdo de documento do ”.

Bloquear compartilhamento público para documentos do SSM

Antes de começar, analise todos os documentos do SSM compartilhados publicamente na sua Conta da AWS e confirme se você deseja continuar compartilhando-os. Para parar de compartilhar um documento do SSM com o público, é necessário modificar a configuração de permissões do documento, conforme descrito na seção Modificar permissões para um documento compartilhado do deste tópico. Ativar a configuração de bloqueio de compartilhamento público não afeta nenhum documento do SSM que você esteja compartilhando atualmente com o público. Com a configuração de bloqueio de compartilhamento público ativada, você não poderá compartilhar nenhum documento do SSM adicional com o público.

A menos que seu caso de uso exija que você compartilhe documentos com o público, recomendamos que ative a configuração de bloqueio de compartilhamento público para os seus documentos do SSM. Ativar essa configuração impede o acesso indesejado aos documentos do SSM. A configuração de compartilhamento público de bloco é uma configuração de nível de conta que pode diferir para cada Região da AWS. Conclua as tarefas a seguir para bloquear o compartilhamento público de quaisquer documentos do SSM que você esteja compartilhando no momento.

Bloquear compartilhamento público (console)

Para bloquear o compartilhamento público de seus documentos do SSM

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.

2. No painel de navegação, escolha Documents.
3. Selecione Preferences (Preferências) e, em seguida, escolha Edit (Editar) na seção Block public sharing (Bloquear compartilhamento público).
4. Selecione Bloquear compartilhamento público e, em seguida, selecione Salvar.

Bloquear compartilhamento público (linha de comando)

Abrir o AWS Command Line Interface (AWS CLI) ou AWS Tools for Windows PowerShell no computador local e execute o comando a seguir para bloquear o compartilhamento público de documentos SSM.

Linux & macOS

```
aws ssm update-service-setting  \
    --setting-id /ssm/documents/console/public-sharing-permission \
    --setting-value Disable \
    --region 'The Região da AWS you want to block public sharing in'
```

Windows

```
aws ssm update-service-setting ^
    --setting-id /ssm/documents/console/public-sharing-permission ^
    --setting-value Disable ^
    --region "The Região da AWS you want to block public sharing in"
```

PowerShell

```
Update-SSMServiceSetting `
    -SettingId /ssm/documents/console/public-sharing-permission `
    -SettingValue Disable `
    -Region The Região da AWS you want to block public sharing in
```

Confirme se o valor da configuração foi atualizado usando o comando a seguir.

Linux & macOS

```
aws ssm get-service-setting   \
    --setting-id /ssm/documents/console/public-sharing-permission \
```

```
    --region The Região da AWS you blocked public sharing in
```

Windows

```
aws ssm get-service-setting  ^
    --setting-id /ssm/documents/console/public-sharing-permission ^
    --region "The Região da AWS you blocked public sharing in"
```

PowerShell

```
Get-SSMServiceSetting `
    -SettingId /ssm/documents/console/public-sharing-permission `
    -Region The Região da AWS you blocked public sharing in
```

Restringir o acesso para bloquear o compartilhamento público com o IAM

Você pode criar políticas do AWS Identity and Access Management (IAM) que impedem os usuários de modificar a configuração de compartilhamento público de blocos. Isso impede que os usuários permitam acesso indesejado aos documentos do SSM.

Veja a seguir um exemplo de uma política do IAM que impede que os usuários atualizem a configuração de compartilhamento público de bloco. Para usar este exemplo, você deve substituir o exemplo de ID de conta da Amazon Web Services pelo seu próprio ID de conta.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "ssm:UpdateServiceSetting",
            "Resource": "arn:aws:ssm:*:444455556666:servicesetting/ssm/documents/
console/public-sharing-permission"
        }
    ]
}
```

Compartilhar um documento do

Você pode compartilhar documentos do AWS Systems Manager (SSM) usando o console do Systems Manager. Ao compartilhar documentos do console, é possível compartilhar somente a versão padrão do documento. Também é possível compartilhar documentos SSM de forma programática chamando a operação da API `ModifyDocumentPermission` usando a AWS Command Line Interface (AWS CLI), o AWS Tools for Windows PowerShell ou o AWS SDK. Antes de compartilhar um documento, obtenha os IDs das Conta da AWS das pessoas com quem deseja compartilhar. Você especificará esses IDs de conta quando compartilhar o documento.

Compartilhar um documento (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Documents.
3. Na lista de documentos, selecione o documento que você deseja compartilhar e escolha View details (Visualizar detalhes). Na guia Permissions, verifique se você é o proprietário do documento. Somente o proprietário de um documento pode compartilhá-lo.
4. Escolha Editar.
5. Para compartilhar o comando publicamente, escolha Public (Público) e depois Save (Salvar). Para compartilhar o comando de forma privada, escolha Private (Privado), insira o ID da Conta da AWS e escolha Add permission (Adicionar permissão) e Save (Salvar).

Compartilhar um documento (linha de comando)

O procedimento a seguir requer que você especifique uma Região da AWS para sua sessão de linha de comando.

1. Abra a AWS CLI ou o AWS Tools for Windows PowerShell no computador local e execute o comando a seguir para especificar suas credenciais.

   No comando a seguir, substitua *`region`* por suas próprias informações. Para ver uma lista dos valores de *`região`* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

   Linux & macOS

   ```
   aws config
   ```

```
AWS Access Key ID: [your key]
AWS Secret Access Key: [your key]
Default region name: region
Default output format [None]:
```

Windows

```
aws config

AWS Access Key ID: [your key]
AWS Secret Access Key: [your key]
Default region name: region
Default output format [None]:
```

PowerShell

```
Set-AWSCredentials –AccessKey your key –SecretKey your key
Set-DefaultAWSRegion -Region region
```

2. Use o seguinte comando para listar todos os documentos do disponíveis para você. A lista inclui os documentos que você criou e os documentos que foram compartilhados com você.

   Linux & macOS

   ```
   aws ssm list-documents
   ```

   Windows

   ```
   aws ssm list-documents
   ```

   PowerShell

   ```
   Get-SSMDocumentList
   ```

3. Use o seguinte comando para obter um documento específico.

   Linux & macOS

   ```
   aws ssm get-document \
       --name document name
   ```

Windows

```
aws ssm get-document ^
    --name document name
```

PowerShell

```
Get-SSMDocument `
    -Name document name
```

4. Use o seguinte comando para obter uma descrição do documento.

Linux & macOS

```
aws ssm describe-document \
    --name document name
```

Windows

```
aws ssm describe-document ^
    --name document name
```

PowerShell

```
Get-SSMDocumentDescription `
    -Name document name
```

5. Use o seguinte comando para visualizar as permissões do documento.

Linux & macOS

```
aws ssm describe-document-permission \
    --name document name \
    --permission-type Share
```

Windows

```
aws ssm describe-document-permission ^
    --name document name ^
```

```
    --permission-type Share
```

PowerShell

```
Get-SSMDocumentPermission `
    -Name document name `
    -PermissionType Share
```

6. Use o seguinte comando para modificar as permissões do documento e compartilhá-lo. Você deve ser o proprietário do documento para editar as permissões. Opcionalmente, para documentos compartilhados com IDs de Conta da AWS específicos, é possível especificar uma versão do documento que deseja compartilhar usando o parâmetro `--shared-document-version`. Se você não especificar a versão, o sistema compartilhará a versão `Default` do documento. Se você compartilhar um documento publicamente (com `all`), o comportamento padrão é compartilhar todas as versões do documento especificado. O exemplo de comando a seguir compartilha o documento de forma privada com uma pessoa específica com base no ID da Conta da AWS dessa pessoa.

   Linux & macOS

   ```
   aws ssm modify-document-permission \
       --name document name \
       --permission-type Share \
       --account-ids-to-add Conta da AWS ID
   ```

   Windows

   ```
   aws ssm modify-document-permission ^
       --name document name ^
       --permission-type Share ^
       --account-ids-to-add Conta da AWS ID
   ```

   PowerShell

   ```
   Edit-SSMDocumentPermission `
       -Name document name `
       -PermissionType Share `
       -AccountIdsToAdd Conta da AWS ID
   ```

7. Use o seguinte comando para compartilhar um documento publicamente.

> **Note**
>
> Se você compartilhar um documento publicamente (com `all`), o comportamento padrão é compartilhar todas as versões do documento especificado.

Linux & macOS

```
aws ssm modify-document-permission \
    --name document name \
    --permission-type Share \
    --account-ids-to-add 'all'
```

Windows

```
aws ssm modify-document-permission ^
    --name document name ^
    --permission-type Share ^
    --account-ids-to-add "all"
```

PowerShell

```
Edit-SSMDocumentPermission `
    -Name document name `
    -PermissionType Share `
    -AccountIdsToAdd ('all')
```

Modificar permissões para um documento compartilhado do

Se você compartilhar um comando, os usuários poderão visualizar e usar esse comando até que você remova o acesso ao documento do AWS Systems Manager (SSM) ou exclua esse documento do SSM. No entanto, não é possível excluir um documento que esteja compartilhado. Você deve parar de compartilhá-lo primeiro e depois excluí-lo.

Parar de compartilhar um documento (console)

Parar de compartilhar um documento

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.

2. No painel de navegação, escolha Documents.
3. Na lista de documentos, selecione o documento que deseja parar de compartilhar e escolha Detalhes. Na guia Permissões, verifique se você é o proprietário do documento. Somente o proprietário de um documento pode parar de compartilhá-lo.
4. Escolha Editar.
5. Selecione X para excluir o ID da Conta da AWS que não deve mais ter acesso ao comando e escolha Save (Salvar).

Parar de compartilhar um documento (linha de comando)

Abra a AWS CLI ou o AWS Tools for Windows PowerShell no computador local e execute o comando a seguir para parar o compartilhamento de um comando.

Linux & macOS

```
aws ssm modify-document-permission \
    --name document name \
    --permission-type Share \
    --account-ids-to-remove 'Conta da AWS ID'
```

Windows

```
aws ssm modify-document-permission ^
    --name document name ^
    --permission-type Share ^
    --account-ids-to-remove "Conta da AWS ID"
```

PowerShell

```
Edit-SSMDocumentPermission `
    -Name document name `
    -PermissionType Share `
    –AccountIdsToRemove Conta da AWS ID
```

Usar documentos compartilhados do

Quando você compartilha um documento do AWS Systems Manager, o sistema gera um nome do recurso da Amazon (ARN) e o atribui ao comando. Se você selecionar e executar um documento

compartilhado do console do Systems Manager, não verá o ARN. Porém, para executar um documento SSM compartilhado usando um outro método, não o console do Systems Manager, você deve especificar o ARN completo do documento para o parâmetro de solicitação `DocumentName`. Você visualiza o ARN completo de um documento do SSM ao executar o comando para listar documentos.

> **Note**
>
> Não é necessário especificar ARNs para documentos públicos da AWS (documentos que começam com `AWS-*`) ou comandos de sua propriedade.

Usar um documento compartilhado do (linha de comando)

Para listar todos os documentos públicos do Systems Manager

Linux & macOS

```
aws ssm list-documents \
    --filters Key=Owner,Values=Public
```

Windows

```
aws ssm list-documents ^
    --filters Key=Owner,Values=Public
```

PowerShell

```
$filter = New-Object Amazon.SimpleSystemsManagement.Model.DocumentKeyValuesFilter
$filter.Key = "Owner"
$filter.Values = "Public"

Get-SSMDocumentList `
    -Filters @($filter)
```

Para listar documentos SSM privados que foram compartilhados com você

Linux & macOS

```
aws ssm list-documents \
```

```
    --filters Key=Owner,Values=Private
```

Windows

```
aws ssm list-documents ^
    --filters Key=Owner,Values=Private
```

PowerShell

```
$filter = New-Object Amazon.SimpleSystemsManagement.Model.DocumentKeyValuesFilter
$filter.Key = "Owner"
$filter.Values = "Private"

Get-SSMDocumentList `
    -Filters @($filter)
```

Para listar todos os documentos SSM disponíveis para você

Linux & macOS

```
aws ssm list-documents
```

Windows

```
aws ssm list-documents
```

PowerShell

```
Get-SSMDocumentList
```

Para obter informações sobre um documento SSM que foi compartilhado com você

Linux & macOS

```
aws ssm describe-document \
    --name arn:aws:ssm:us-east-2:12345678912:document/documentName
```

Windows

```
aws ssm describe-document ^
    --name arn:aws:ssm:us-east-2:12345678912:document/documentName
```

PowerShell

```
Get-SSMDocumentDescription `
    -Name arn:aws:ssm:us-east-2:12345678912:document/documentName
```

Para executar um documento SSM compartilhado

Linux & macOS

```
aws ssm send-command \
    --document-name arn:aws:ssm:us-east-2:12345678912:document/documentName \
    --instance-ids ID
```

Windows

```
aws ssm send-command ^
    --document-name arn:aws:ssm:us-east-2:12345678912:document/documentName ^
    --instance-ids ID
```

PowerShell

```
Send-SSMCommand `
    -DocumentName arn:aws:ssm:us-east-2:12345678912:document/documentName `
    -InstanceIds ID
```

Pesquisando documentos do SSM

Você pode pesquisar aAWS Systems Manager(SSM) para documentos SSM usando pesquisa de texto livre ou uma pesquisa baseada em filtro. Também é possível marcar documentos como favoritos para ajudar você a encontrar documentos do SSM usados com frequência. As seções a seguir descrevem como usar esses recursos.

Usando a pesquisa de texto livre

A caixa de pesquisa na página Documents (Documentos) do Systems Manager oferece suporte à pesquisa de texto livre. A pesquisa de texto livre compara o termo ou termos de pesquisa inseridos com o nome do documento em cada documento do SSM. Se você inserir um único termo de pesquisa, por exemplo**`ansible`**, o Systems Manager retorna todos os documentos SSM onde esse termo foi descoberto. Se você inserir vários termos de pesquisa, o Systems Manager pesquisará usando um0Rinstrução. Por exemplo, se você especificar **`ansible`** e **`linux`**, a pesquisa retornará todos os documentos com qualquer uma das palavras-chave em seu nome.

Se você inserir um termo de pesquisa de texto livre e escolher uma opção de pesquisa, comoTipo de plataforma, em seguida, a pesquisa usa umANDe retorna todos os documentos com a palavra-chave em seu nome e o tipo de plataforma especificado.

> **Note**
>
> Observe os seguintes detalhes sobre a pesquisa de texto livre.
>
> - A pesquisa de texto livre não diferencia maiúsculas de minúsculas.
> - Os termos de pesquisa exigem um mínimo de três caracteres e têm um máximo de 20 caracteres.
> - A pesquisa de texto livre aceita até cinco termos de pesquisa.
> - Se você inserir um espaço entre termos de pesquisa, o sistema incluirá o espaço durante a pesquisa.
> - Você pode combinar a pesquisa de texto livre com outras opções de pesquisa, comoTipo de documentoouTipo de plataforma.
> - OPrefixo do Nome do documentoO filtro e a pesquisa de texto livre não podem ser usados juntos. Eles são mutuamente exclusivos.

Para pesquisar um documento do SSM

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Documents.
3. Insira os termos da pesquisa na caixa de pesquisa e pressione Enter.

Executando pesquisa de documentos de texto livre usando o comandoAWS CLI

Para executar uma pesquisa de documento de texto livre usando a CLI

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.
2. Para executar a pesquisa de documento de texto livre com um único termo, execute o comando a seguir. Neste comando, substitua*search_term*com suas próprias informações.

   ```
   aws ssm list-documents --filters Key="SearchKeyword",Values="search_term"
   ```

   Aqui está um exemplo.

   ```
   aws ssm list-documents --filters Key="SearchKeyword",Values="aws-asg" --region us-
   east-2
   ```

   Para pesquisar usando vários termos que criam umAND, execute o seguinte comando. Neste comando, substitua*search_term_1*e*search_term_2*com suas próprias informações.

   ```
   aws ssm list-documents --filters
    Key="SearchKeyword",Values="search_term_1","search_term_2","search_term_3" --
   region us-east-2
   ```

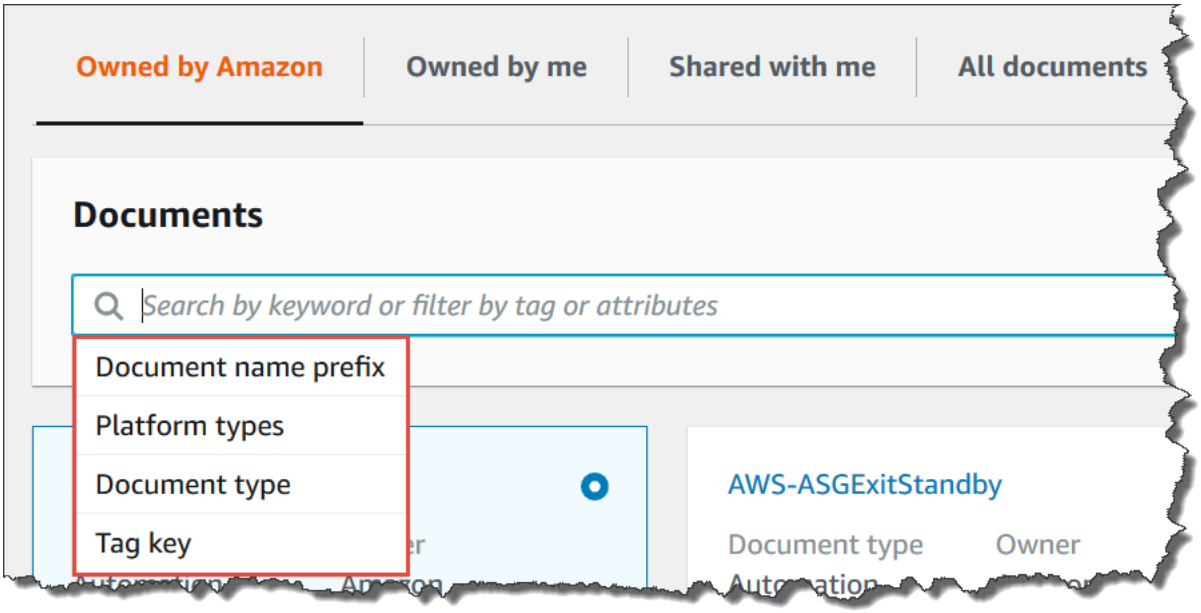
   Aqui está um exemplo.

   ```
   aws ssm list-documents --filters Key="SearchKeyword",Values="aws-asg","aws-
   ec2","restart" --region us-east-2
   ```

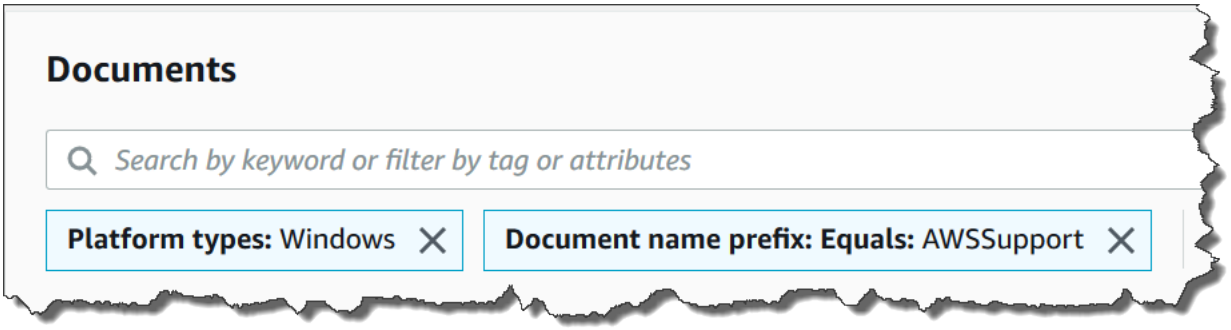
Usar filtros

A página Documents (Documentos) do Systems Manager exibe automaticamente os filtros a seguir quando você escolhe a caixa de pesquisa.

- Prefixo do nome do documento
- Tipos de plataforma
- Tipo de documento
- Chave de tag



Você pode procurar documentos SSM usando um único filtro. Se você quiser retornar um conjunto mais específico de documentos SSM, você pode aplicar vários filtros. Veja a seguir um exemplo de pesquisa que usa os Tipos de plataforma e os filtros do Prefixo do nome do documento.



Se você aplicar vários filtros, o Systems Manager criará instruções de pesquisa diferentes com base nos filtros escolhidos:

- Se você aplicar olgualfiltrar várias vezes, por exemploPrefixo do nome do documentoe, em seguida, o Systems Manager procura utilizando um0Rinstrução. Por exemplo, se você especificar um filtro Document name prefix=**`AWS`** e um segundo filtro Document name prefix=**`Lambda`**, a pesquisa retornará todos os documentos com o prefixo "`AWS`" e todos os documentos com o prefixo "`Lambda`".
- Se você aplicar filtros diferentes, por exemplo Document name prefix (Prefixo do nome do documento) e Platform types (Tipos de plataforma), o Systems Manager fará a busca utilizando uma instrução `AND`. Por exemplo, se você especificar um filtro Document name prefix=**`AWS`** (Prefixo

do nome do documento=) e um filtro Platform types=**`Linux`** (Tipo de plataforma=), a pesquisa retornará todos os documentos com o prefixo "`AWS`" que forem específicos da plataforma Linux.

> **Note**
>
> As pesquisas que usam filtros diferenciam maiúsculas de minúsculas

Como adicionar documentos aos seus favoritos

Para ajudar você a encontrar documentos do SSM usados com frequência, adicione-os aos seus favoritos. É possível adicionar até 20 documentos como favoritos por tipo de documento, Conta da AWS e Região da AWS. Você pode escolher, modificar e visualizar seus favoritos nos documentos do AWS Management Console. Os procedimentos a seguir descrevem como escolher, modificar e visualizar seus favoritos.

Como adicionar um documento do SSM aos favoritos

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Documents.
3. Selecione o ícone de estrela ao lado do nome do documento que deseja adicionar como favorito.

Como remover um documento do SSM dos seus favoritos

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Documents.
3. Desmarque o ícone de estrela ao lado do nome do documento que deseja remover dos seus favoritos.

Como visualizar seus favoritos nos documentos do AWS Management Console

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Documents.
3. Selecione a guia Favoritos.

Solução de problemas de tratamento de parâmetros

Problemas comuns de tratamento de parâmetros

Variáveis de ambiente não disponíveis durante a execução

Problema: os comandos falham porque as variáveis de ambiente (`SSM_parameter-name`) não foram encontradas.

Causas possíveis:

- A versão do SSM Agent não é compatível com a interpolação de variáveis de ambiente
- `interpolationType` não está definido como `ENV_VAR`
- O nome do parâmetro não corresponde ao nome da variável de ambiente esperada

Solução:

- Verifique se a versão do SSM Agent é 3.3.2746.0 ou posterior
- Adicione uma lógica de fallback para as versões mais antigas do agente:

```
if [ -z "${SSM_parameterName+x}" ]; then
    export SSM_parameterName="{{parameterName}}"
fi
```

Valores de parâmetros com caracteres especiais

Problema: os comandos falham quando os valores dos parâmetros contêm espaços, aspas ou outros caracteres especiais.

Solução:

- Use as aspas adequadas ao referenciar variáveis de ambiente:

```
# Correct
echo "$SSM_parameter-name"

# Incorrect
echo $SSM_parameter-name
```

- Adicione validação de entrada usando `allowedPattern` para restringir caracteres especiais

Comportamento inconsistente entre plataformas

Problema: o tratamento de parâmetros funciona de forma diferente nos sistemas Linux e Windows Server.

Solução:

- Use a sintaxe da variável de ambiente específica da plataforma:

```
# PowerShell
$env:SSM_parameter-name

# Bash
$SSM_parameter-name
```

- Use verificações de pré-condições específicas da plataforma em seu documento

Valores dos parâmetros não foram devidamente escapados

Problema: vulnerabilidades de injeção de comando, apesar de usar a interpolação de variáveis de ambiente.

Solução:

- Sempre use o escape adequado ao incluir valores de parâmetros nos comandos:

```
# Correct
mysql_command="mysql -u \"$SSM_username\" -p\"$SSM_password\""

# Incorrect
mysql_command="mysql -u $SSM_username -p$SSM_password"
```

Dicas de validação de parâmetros

Use estas técnicas para validar o tratamento de parâmetros:

1. Verifique a disponibilidade de variáveis de ambiente:

```
#!/bin/bash
# Print all SSM_ environment variables
env | grep ^SSM_

# Test specific parameter
if [ -n "$SSM_parameter" ]; then
```

```
    echo "Parameter is available"
else
    echo "Parameter is not available"
fi
```

2. Verifique os padrões dos parâmetros:

```
parameters:
  myParameter:
    type: String
    allowedPattern: "^[a-zA-Z0-9_-]+$"
    description: "Test this pattern with sample inputs"
```

3. Inclua o tratamento de erros:

```
if [[ ! "$SSM_parameter" =~ ^[a-zA-Z0-9_-]+$ ]]; then
    echo "Parameter validation failed"
    exit 1
fi
```

AWS Systems Manager Maintenance Windows

O Maintenance Windows, uma ferramenta do AWS Systems Manager, ajuda você a definir um agendamento de quando executar ações que possivelmente causarão interrupções aos nós, como aplicar patches a um sistema operacional, atualizar drivers ou instalar um software ou patches.

> **Note**
>
> O State Manager e Maintenance Windows podem executar alguns tipos semelhantes de atualizações nos seus nós gerenciados. Qual deles escolher depende se você precisa automatizar a conformidade do sistema ou executar tarefas de alta prioridade e sensíveis ao tempo durante os períodos especificados.
> Para obter mais informações, consulte Selecionar entre State Manager e Maintenance Windows.

Com o Maintenance Windows, você pode agendar ações em vários outros tipos de recursos da AWS, como buckets do Amazon Simple Storage Service (Amazon S3), filas do Amazon Simple Queue Service (Amazon SQS), chaves do AWS Key Management Service (AWS KMS) e muito mais.

Para obter uma lista completa dos tipos de recursos compatíveis que você pode incluir em um destino da janela de manutenção, consulte Recursos que você pode usar com o AWS Resource Groups e o Editor de tags no Guia do usuárioAWS Resource Groups. Para começar a usar o Maintenance Windows, abra o Systems Manager console (Console do gerenciador de sistemas). No painel de navegação, escolha Maintenance Windows.

Cada janela de manutenção tem uma programação, uma duração máxima, um conjunto de destinos registrados (os nós gerenciados ou outros recursos da AWS nas quais as ações ocorrem) e um conjunto de tarefas registradas. Você pode adicionar tags às suas janelas de manutenção quando você criá-las ou atualizá-las. (Tags são chaves que ajudam a identificar e classificar os recursos na sua organização). Também é possível especificar datas antes ou depois das quais uma janela de manutenção não deve ser executada e especificar o fuso horário internacional no qual basear a programação da janela de manutenção.

Para obter uma explicação de como as várias opções relacionadas à programação de janelas de manutenção se relacionam entre si, consulte Opções de programação da janela de manutenção e do período ativo.

Para obter mais informações sobre como trabalhar com a opção `--schedule`, consulte Referência: Expressões cron e rate para o Systems Manager.

Tipos de tarefas compatíveis

Com as janelas de manutenção, você pode executar quatro tipos de tarefas:

- Comandos no Run Command, uma ferramenta do Systems Manager

  Para obter mais informações sobre o Run Command, consulte AWS Systems Manager Run Command.
- Fluxos de trabalho no Automation, uma ferramenta do Systems Manager

  Para obter mais informações sobre fluxos de trabalho de Automação, consulte AWS Systems Manager Automation.
- Funções no AWS Lambda

  Para obter mais informações sobre como criar uma função do Lambda, consulte Conceitos básicos do Lambda no Guia do desenvolvedor do AWS Lambda.
- Tarefas no AWS Step Functions

> **Note**
>
> As tarefas da janela de manutenção oferecem suporte somente aos fluxos de trabalho da máquina de estado Step Functions Standard. Elas não oferecem suporte a fluxos de trabalho de máquinas de estado Express. Para obter informações sobre os tipos de fluxo de trabalho da máquina de estado, consulte Fluxos de trabalho Standard vs. Express no Guia do desenvolvedor do AWS Step Functions.

Para obter mais informações sobre o Steps Functions, consulte o Guia do desenvolvedor do AWS Step Functions.

Isso significa que você pode usar a janela de manutenção para executar tarefas como a seguinte em seus destinos selecionados.

- Instalar ou atualizar aplicativos.
- Aplicar patches.
- Instalar ou atualizar o SSM Agent.
- Execute comandos do PowerShell e scripts de shell do Linux usando uma tarefa Run Command do Systems Manager.
- Crie Amazon Machine Images (AMIs), faça o bootstrapping do software e configure os nós usando uma tarefa do Systems Manager Automation.
- Execute funções do AWS Lambda que invoquem outras ações, como verificar nós para a atualizações de patches.
- Execute máquinas de estado do AWS Step Functions para executar tarefas como remover um nó de um ambiente do Elastic Load Balancing, aplicar patch a um nó e, em seguida, adicionar o nó de volta ao ambiente do Elastic Load Balancing.
- Nós de destino que estiverem offline especificando um grupo de recursos da AWS como destino.

> **Note**
>
> Especifique um ou mais destinos para as tarefas da janela de manutenção do tipo Run Command. Dependendo da tarefa, os destinos serão opcionais para outros tipos de tarefas da janela de manutenção (Automation, AWS Lambda e AWS Step Functions). Para obter

mais informações sobre como executar tarefas que não especificam destinos, consulte Registrar tarefas da janela de manutenção sem destinos.

Suporte ao EventBridge

Essa ferramenta do Systems Manager pode ser usada como um tipo de evento nas regras do Amazon EventBridge. Para obter informações, consulte Monitorar eventos do Systems Manager com o Amazon EventBridge e Referência: padrões e tipos de eventos do Amazon EventBridge para o Systems Manager.

Conteúdo

- Configurar o Maintenance Windows
- Crie e gerencie janelas de manutenção usando o console
- Tutoriais
- Usar pseudoparâmetros ao registrar tarefas da janela de manutenção
- Opções de programação da janela de manutenção e do período ativo
- Registrar tarefas da janela de manutenção sem destinos
- Solução de problemas das janelas

Configurar o Maintenance Windows

Antes que os usuários na sua Conta da AWS possam criar e programar tarefas da janela de manutenção usando o Maintenance Windows, uma ferramenta do AWS Systems Manager, eles devem receber as permissões necessárias para isso. Além disso, você deve criar um perfil de serviço do IAM para janelas de manutenção e a política do IAM a ser anexada a ele.

Antes de começar

Além das permissões que você configura nesta seção, as Entidades do IAM (usuários, perfis ou grupos) que funcionarão com janelas de manutenção já devem ter permissões gerais da janela de manutenção. Faça isso ao atribuir a política do IAM `AmazonSSMFullAccess` às Entidades, ou ao atribuir uma política do IAM personalizada que forneça um conjunto menor de permissões de acesso para o Systems Manager que abranja as tarefas da janela de manutenção.

Tópicos

- Control access to maintenance windows using the console
- Control access to maintenance windows using the AWS CLI

Control access to maintenance windows using the console

Os procedimentos a seguir descrevem como usar o console do AWS Systems Manager para criar as permissões e perfis necessários para janelas de manutenção.

Tópicos

- Tarefa 1: crie uma política personalizada para seu perfil de serviço de janela de manutenção usando o console
- Tarefa 2: crie um perfil de serviço personalizado para janelas de manutenção usando o console
- Tarefa 3: conceda permissões a usuários especificados para que registrem tarefas da janela de manutenção usando o console
- Tarefa 4: evite que usuários especificados registrem tarefas da janela de manutenção usando o console

Tarefa 1: crie uma política personalizada para seu perfil de serviço de janela de manutenção usando o console

As tarefas de janela de manutenção exigem um perfil do IAM para fornecer as permissões necessárias para serem executadas nos recursos de destino. As permissões são fornecidas por meio de uma política do IAM anexada ao perfil. Os tipos de tarefas executadas e seus outros requisitos operacionais determinam o conteúdo dessa política. Fornecemos uma política básica que você pode adaptar às suas necessidades. Dependendo das tarefas e dos tipos de tarefas que as janelas de manutenção executam, talvez você não precise de todas as permissões nesta política, e talvez seja necessário incluir permissões adicionais. Você anexará essa política ao perfil que criar posteriormente na Tarefa 2: crie um perfil de serviço personalizado para janelas de manutenção usando o console.

Para criar uma política personalizada usando o console

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Políticas e, em seguida, Criar política.
3. Na área Editor de políticas, escolha JSON.
4. Substitua o conteúdo padrão pelo seguinte:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand",
                "ssm:CancelCommand",
                "ssm:ListCommands",
                "ssm:ListCommandInvocations",
                "ssm:GetCommandInvocation",
                "ssm:GetAutomationExecution",
                "ssm:StartAutomationExecution",
                "ssm:ListTagsForResource",
                "ssm:DescribeInstanceInformation",
                "ssm:GetParameters"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "states:DescribeExecution",
                "states:StartExecution"
            ],
            "Resource": [
                "arn:aws:states:*:*:execution:*:*",
                "arn:aws:states:*:*:stateMachine:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": [
                "arn:aws:lambda:*:*:function:*"
            ]
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "resource-groups:ListGroups",
                "resource-groups:ListGroupResources"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/maintenance-window-
role-name",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ssm.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

5. Modifique o conteúdo JSON conforme necessário para as tarefas de manutenção executadas na sua conta. As alterações feitas são específicas para suas operações planejadas.

   Por exemplo:

   - Você pode fornecer nomes do recurso da Amazon (ARNs) para funções específicas e máquinas de estado em vez de usar qualificadores-curinga (*).
   - Se você não planeja executar tarefas do AWS Step Functions, é possível remover as permissões `states` e os ARNs.

- Se você não planeja executar tarefas do AWS Lambda, é possível remover as permissões `lambda` e os ARNs.
- Se você não planeja executar tarefas do Automation, é possível remover as permissões `ssm:GetAutomationExecution` e `ssm:StartAutomationExecution`.
- Adicione mais permissões que podem ser necessárias para que as tarefas sejam executadas. Por exemplo, algumas ações da Automação trabalham com pilhas do AWS CloudFormation. Portanto, as permissões `cloudformation:CreateStack`, `cloudformation:DescribeStacks`, e `cloudformation:DeleteStack` são necessárias.

  Outro exemplo: o runbook `AWS-CopySnapshot` do Automation requer permissão para criar um snapshot do Amazon Elastic Block Store (Amazon EBS). Portanto, o perfil de serviço precisa da permissão `ec2:CreateSnapshot`.

  Para obter informações sobre as permissões de perfil necessárias para os runbooks de automação, consulte as descrições de runbooks em Referência de runbooks do AWS Systems Manager Automation.

6. Depois de concluir as revisões da política, escolha Próximo.
7. Em Nome da política, insira um nome que identifique isso como a política anexada pelo perfil de serviço que você cria. Por exemplo: **`my-maintenance-window-role-policy`**.
8. (Opcional) Na área Adicionar tags, adicione um ou mais pares de chave-valor de etiqueta para organizar, monitorar ou controlar acesso para essa política.
9. Escolha Criar política.

   Anote o nome que você especificou para a política. Você fará referência a ele no próximo procedimento, Tarefa 2: crie um perfil de serviço personalizado para janelas de manutenção usando o console.

Tarefa 2: crie um perfil de serviço personalizado para janelas de manutenção usando o console

A política que você criou na tarefa anterior está anexada ao perfil de serviço da janela de manutenção criado na mesma. Quando os usuários registram uma tarefa de janela de manutenção, eles especificam esse perfil do IAM como parte da configuração da tarefa. As permissões nesse perfil permitem que o Systems Manager execute tarefas em janelas de manutenção em seu nome.

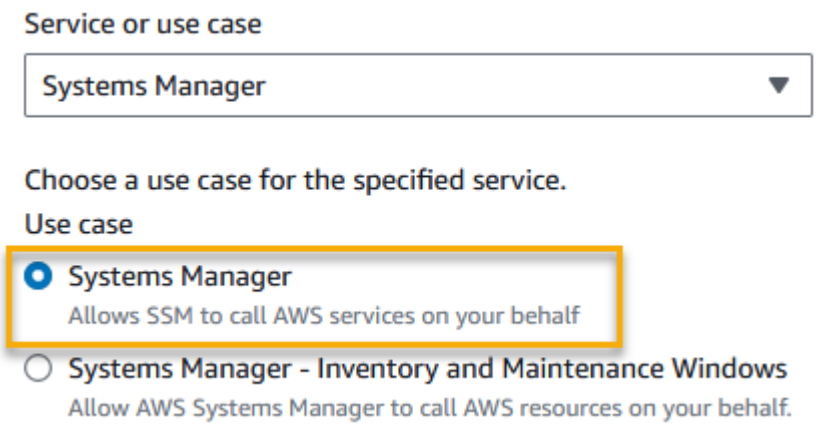
> **Important**
>
> Anteriormente, o console do Systems Manager permitia a você escolher o perfil `AWSServiceRoleForAmazonSSM` vinculado ao serviço do IAM gerenciado pela AWS para usar como perfil de manutenção para suas tarefas. O uso desse perfil e sua política associada, `AmazonSSMServiceRolePolicy`, para tarefas de janela de manutenção não é mais recomendado. Se estiver usando esse perfil para tarefas de janela de manutenção agora, recomendamos parar de usá-lo. Em vez disso, crie seu próprio perfil do IAM para permitir a comunicação entre o Systems Manager e outros Serviços da AWS quando as tarefas da janela de manutenção são executadas.

Use o procedimento a seguir para criar um perfil de serviço personalizado para as Maintenance Windows, para que o Systems Manager possa executar tarefas de Maintenance Windows em seu nome. Basta anexar a política criada na tarefa anterior ao perfil de serviço personalizado criado.

Para criar um perfil de serviço personalizado para janelas de manutenção usando o console

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Roles e Create role.
3. Em Select trusted entity (Selecionar entidade confiável), faça as seguintes escolhas:
    1. Em Tipo de entidade confiável, escolha Serviços da AWS.
    2. Em Caso de uso, escolha Systems Manager
    3. Escolha Systems Manager.

    A imagem a seguir destaca a localização da opção Systems Manager.



4. Escolha Próximo.
5. Na área Políticas de permissões, na caixa de pesquisa, insira o nome da política que você criou na Tarefa 1: crie uma política personalizada para seu perfil de serviço de janela de manutenção usando o console, marque a caixa ao lado do nome e escolha Próximo.
6. Em Role name (Nome da regra), insira um nome que identifique essa função como uma função da Maintenance Windows. Por exemplo: **`my-maintenance-window-role`**.
7. (Opcional) Altere a descrição da função padrão para refletir a finalidade dessa função. Por exemplo: **`Performs maintenance window tasks on your behalf`**.
8. Na Etapa 1: Selecionar entidades confiáveis, verifique se a política a seguir é exibida na caixa Política confiável.

   JSON

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "",
         "Effect": "Allow",
         "Principal": {
           "Service": "ssm.amazonaws.com"
         },
         "Action": "sts:AssumeRole"
       }
     ]
   }
   ```

9. Na Etapa 2: Adicionar permissões, verifique se a política que você criou Tarefa 1: crie uma política personalizada para seu perfil de serviço de janela de manutenção usando o console está presente.
10. (Opcional) Em Etapa 3: Adicionar tags, adicione um ou mais pares de valor tag-chave para organizar, monitorar ou controlar acesso para esse perfil.
11. Selecione Create role (Criar função). O sistema faz com que você retorne para a página Roles.
12. Escolha o nome do perfil do IAM que você acabou de criar.
13. Copie ou anote o nome do perfil e o valor do ARN na área Summary (Resumo). Os usuários em sua conta especificam essas informações ao criarem janelas de manutenção.

Tarefa 3: conceda permissões a usuários especificados para que registrem tarefas da janela de manutenção usando o console

Fornecer aos usuários permissões para acessar o perfil de serviço da janela de manutenção personalizado permite que eles a usem com suas tarefas de janelas de manutenção. Isso é além das permissões que você concedeu a eles para trabalhar com os comandos da API do Systems Manager para a ferramenta do Maintenance Windows. Esse perfil do IAM transmite as permissões necessárias para executar uma tarefa de janela de manutenção. Como resultado, um usuário não poderá registrar tarefas a uma janela de manutenção usando seu perfil de serviço personalizado sem a capacidade de passar essas permissões do IAM.

Ao registrar uma tarefa em uma janela de manutenção, você especifica um perfil de serviço para executar as operações de tarefas reais. Esta é a função que o serviço assumirá quando executar tarefas em seu nome. Antes disso, para registrar a própria tarefa, é necessário atribuir a política do IAM `PassRole` a uma entidade do IAM (como um usuário ou grupo). Isso permite que a entidade do IAM especifique, como parte do registro dessas tarefas na janela de manutenção, o perfil que deve ser usado ao executar as tarefas. Para obter informações, consulte Conceda permissões a um usuário para passar um perfil para um AWS service (Serviço da AWS), no Guia do usuário do IAM.

Para configurar permissões para permitir que usuários registrem tarefas da janela de manutenção

Se uma entidade do IAM (usuário, perfil ou grupo) for configurada com permissões de administrador, o usuário ou o perfil do IAM terá acesso às janelas de manutenção. Para entidades do IAM que não têm permissões de administrador, um administrador deve conceder as permissões a seguir à entidade do IAM. Estas são as permissões mínimas requeridas para o registro de tarefas em uma janela de manutenção:

- A política gerenciada `AmazonSSMFullAccess` ou uma política que forneça permissões comparáveis.
- O seguinte `iam:PassRole` e as permissões `iam:ListRoles`.

  JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
```

```
            "Resource": "arn:aws:iam::111122223333:role/my-maintenance-window-
role"
        },
        {
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "arn:aws:iam::111122223333:role/"
        },
        {
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "arn:aws:iam::111122223333:role/aws-service-role/
ssm.amazonaws.com/"
        }
    ]
}
```

my-maintenance-window-role representa o nome do perfil de serviço da janela de manutenção personalizada criada anteriormente.

account-id representa o ID da Conta da AWS. Adicionar essa permissão para o recurso `arn:aws:iam::`*account-id*`:role/` permite que um usuário visualize e escolha entre funções de cliente no console ao criar uma tarefa da janela de manutenção. A adição dessa permissão para `arn:aws:iam::`*account-id*`:role/aws-service-role/ssm.amazonaws.com/` permite que um usuário escolha a função vinculada ao serviço do Systems Manager no console ao criar uma tarefa da janela de manutenção.

Para conceder acesso, adicione as permissões aos seus usuários, grupos ou perfis:

- Usuários e grupos no AWS IAM Identity Center:

  Crie um conjunto de permissões. Siga as instruções em Criação de um conjunto de permissões no Guia do usuário do AWS IAM Identity Center.

- Usuários gerenciados no IAM com provedor de identidades:

  Crie um perfil para a federação de identidades. Siga as instruções em Criando um perfil para um provedor de identidades de terceiros (federação) no Guia do Usuário do IAM.

- Usuários do IAM:
  - Crie um perfil que seu usuário possa assumir. Siga as instruções em Criação de um perfil para um usuário do IAM no Guia do usuário do IAM.

- (Não recomendado) Vincule uma política diretamente a um usuário ou adicione um usuário a um grupo de usuários. Siga as instruções em Adição de permissões a um usuário (console) no Guia do usuário do IAM.

Para configurar permissões para grupos que tiverem permissão para registrar tarefas da janela de manutenção usando o console

1. Abra o console do IAM, em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione User groups (Grupos de usuários).
3. Na lista de grupos, selecione o nome do grupo do qual você deseja atribuir a permissão `iam:PassRole`, ou crie primeiro um novo grupo, se necessário.
4. Na guia Permissions (Permissões), escolha Add permissions, Create inline policy (Adicionar permissões, Criar política em linha).
5. Na área Editor de políticas, escolha JSON, e substitua o conteúdo padrão da caixa, pelo seguinte:

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/my-maintenance-
window-role"
        },
        {
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "arn:aws:iam::111122223333:role/"
        },
        {
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "arn:aws:iam::111122223333:role/aws-service-role/
ssm.amazonaws.com/"
        }
    ]
```

```
}
```

my-maintenance-window-role representa o nome da função da janela de manutenção personalizada criada anteriormente.

account-id representa o ID da Conta da AWS. Adicionar essa permissão para o recurso `arn:aws:iam::`*account-id*`:role/` permite que um usuário visualize e escolha entre funções de cliente no console ao criar uma tarefa da janela de manutenção. A adição dessa permissão para `arn:aws:iam::`*account-id*`:role/aws-service-role/ssm.amazonaws.com/` permite que um usuário escolha a função vinculada ao serviço do Systems Manager no console ao criar uma tarefa da janela de manutenção.

6. Escolha Próximo.
7. Na página Analisar e criar, insira um nome na caixa Nome da política para identificar essa política de `PassRole`, como **`my-group-iam-passrole-policy`**, e selecione Criar política.

Tarefa 4: evite que usuários especificados registrem tarefas da janela de manutenção usando o console

É possível negar a permissão `ssm:RegisterTaskWithMaintenanceWindow` aos usuários da sua Conta da AWS para os quais você não quer registrar tarefas com janelas de manutenção. Isso fornece uma camada extra de prevenção para usuários que não devem registrar tarefas da janela de manutenção.

Para configurar permissões para grupos sem permissão para registrar tarefas da janela de manutenção usando o console

1. Abra o console do IAM, em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione User groups (Grupos de usuários).
3. Na lista de grupos, selecione o nome do grupo do qual você deseja recusar a permissão `ssm:RegisterTaskWithMaintenanceWindow`, ou crie primeiro um novo grupo, se necessário.
4. Na guia Permissions (Permissões), escolha Add permissions, Create inline policy (Adicionar permissões, Criar política em linha).
5. Na área Editor de políticas, escolha JSON, em seguida substitua o conteúdo padrão da caixa, pelo seguinte:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "ssm:RegisterTaskWithMaintenanceWindow",
            "Resource": "*"
        }
    ]
}
```

6. Escolha Próximo.
7. Na página Analisar e criar, em Nome da política, insira um nome para identificar a política, como **`my-groups-deny-mw-tasks-policy`**, e selecione Criar política.

Control access to maintenance windows using the AWS CLI

Os procedimentos a seguir descrevem como usar a AWS Command Line Interface (AWS CLI) para criar os perfis e permissões necessários para o Maintenance Windows, uma ferramenta do AWS Systems Manager.

Tópicos

- Tarefa 1: crie uma política de confiança e arquivos de políticas gerenciados pelo cliente no formato JSON
- Tarefa 2: crie um perfil de serviço personalizado para tarefas de janela de manutenção usando a AWS CLI
- Tarefa 3: conceda permissões a usuários especificados para que registrem tarefas da janela de manutenção usando a AWS CLI
- Tarefa 4: evite que usuários especificados registrem tarefas da janela de manutenção usando a AWS CLI

Tarefa 1: crie uma política de confiança e arquivos de políticas gerenciados pelo cliente no formato JSON

As tarefas de janela de manutenção exigem um perfil do IAM para fornecer as permissões necessárias para serem executadas nos recursos de destino. As permissões são fornecidas por meio de uma política do IAM anexada ao perfil. Os tipos de tarefas executadas e seus outros requisitos operacionais determinam o conteúdo dessa política. Fornecemos uma política básica que você pode adaptar às suas necessidades. Dependendo das tarefas e dos tipos de tarefas que as janelas de manutenção executam, talvez você não precise de todas as permissões nesta política, e talvez seja necessário incluir permissões adicionais.

Nessa tarefa, você especifica as permissões necessárias para o seu perfil da janela de manutenção personalizada em um par de arquivos JSON. Você anexará essa política ao perfil que criar posteriormente na Tarefa 2: crie um perfil de serviço personalizado para tarefas de janela de manutenção usando a AWS CLI.

Para criar uma política de confiança e arquivos da política gerenciada pelo cliente

1. Copie e cole a política de confiança a seguir em um arquivo de texto. Salve o arquivo com o seguinte nome e extensão: **`mw-role-trust-policy.json`**.

   JSON

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Effect": "Allow",
               "Principal": {
                   "Service": "ssm.amazonaws.com"
               },
               "Action": "sts:AssumeRole"
           }
       ]
   }
   ```

2. Copie e cole a política JSON a seguir em um arquivo de texto distinto. No mesmo diretório em que você criou o primeiro arquivo, salve-o com o seguinte nome e extensão de arquivo: **`mw-role-custom-policy.json`**.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand",
                "ssm:CancelCommand",
                "ssm:ListCommands",
                "ssm:ListCommandInvocations",
                "ssm:GetCommandInvocation",
                "ssm:GetAutomationExecution",
                "ssm:StartAutomationExecution",
                "ssm:ListTagsForResource",
                "ssm:GetParameters"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "states:DescribeExecution",
                "states:StartExecution"
            ],
            "Resource": [
                "arn:aws:states:*:*:execution:*:*",
                "arn:aws:states:*:*:stateMachine:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": [
                "arn:aws:lambda:*:*:function:*"
            ]
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "resource-groups:ListGroups",
                "resource-groups:ListGroupResources"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/maintenance-window-
role-name",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ssm.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

3. Modifique o conteúdo do `mw-role-custom-policy.json` conforme necessário para as tarefas de manutenção executadas na sua conta. As alterações feitas são específicas para suas operações planejadas.

   Por exemplo:

   - Você pode fornecer nomes do recurso da Amazon (ARNs) para funções específicas e máquinas de estado em vez de usar qualificadores-curinga (*).
   - Se você não planeja executar tarefas do AWS Step Functions, é possível remover as permissões `states` e os ARNs.

- Se você não planeja executar tarefas do AWS Lambda, é possível remover as permissões `lambda` e os ARNs.
- Se você não planeja executar tarefas do Automation, é possível remover as permissões `ssm:GetAutomationExecution` e `ssm:StartAutomationExecution`.
- Adicione mais permissões que podem ser necessárias para que as tarefas sejam executadas. Por exemplo, algumas ações da Automação trabalham com pilhas do AWS CloudFormation. Portanto, as permissões `cloudformation:CreateStack`, `cloudformation:DescribeStacks`, e `cloudformation:DeleteStack` são necessárias.

  Outro exemplo: o runbook `AWS-CopySnapshot` do Automation requer permissão para criar um snapshot do Amazon Elastic Block Store (Amazon EBS). Portanto, o perfil de serviço precisa da permissão `ec2:CreateSnapshot`.

  Para obter informações sobre as permissões de perfil necessárias para os runbooks de automação, consulte as descrições de runbooks em Referência de runbooks do AWS Systems Manager Automation.

Salve o arquivo novamente depois de fazer as alterações necessárias.

Tarefa 2: crie um perfil de serviço personalizado para tarefas de janela de manutenção usando a AWS CLI

A política que você criou na tarefa anterior está anexada ao perfil de serviço da janela de manutenção criado na mesma. Quando os usuários registram uma tarefa de janela de manutenção, eles especificam esse perfil do IAM como parte da configuração da tarefa. As permissões nesse perfil permitem que o Systems Manager execute tarefas em janelas de manutenção em seu nome.

> **Important**
>
> Anteriormente, o console do Systems Manager permitia a você escolher o perfil `AWSServiceRoleForAmazonSSM` vinculado ao serviço do IAM gerenciado pela AWS para usar como perfil de manutenção para suas tarefas. O uso desse perfil e sua política associada, `AmazonSSMServiceRolePolicy`, para tarefas de janela de manutenção não é mais recomendado. Se estiver usando esse perfil para tarefas de janela de manutenção agora, recomendamos parar de usá-lo. Em vez disso, crie seu próprio perfil do IAM para

permitir a comunicação entre o Systems Manager e outros Serviços da AWS quando as tarefas da janela de manutenção são executadas.

Nessa tarefa, você executa comandos CLI para criar seu perfil de serviço de janelas de manutenção, adicionando o conteúdo da política dos arquivos JSON que você criou.

Crie um perfil de serviço personalizado para janelas de manutenção usando a AWS CLI

1. Abra a AWS CLI e execute o seguinte comando no diretório emm que você colocou `mw-role-custom-policy.json` e `mw-role-trust-policy.json`. O comando cria um perfil de serviço da janela de manutenção chamado `my-maintenance-window-role` e anexa a política de confiança a ela.

   Linux & macOS

   ```
   aws iam create-role \
       --role-name "my-maintenance-window-role" \
       --assume-role-policy-document file://mw-role-trust-policy.json
   ```

   Windows

   ```
   aws iam create-role ^
       --role-name "my-maintenance-window-role" ^
       --assume-role-policy-document file://mw-role-trust-policy.json
   ```

   O sistema retorna informações semelhantes às seguintes.

   ```
   {
       "Role": {
           "AssumeRolePolicyDocument": {
               "Version": "2012-10-17",
               "Statement": [
                   {
                       "Action": "sts:AssumeRole",
                       "Effect": "Allow",
                       "Principal": {
                           "Service": "ssm.amazonaws.com"
                       }
                   }
   ```

```
            ]
        },
        "RoleId": "AROAIIZKPBKS2LEXAMPLE",
        "CreateDate": "2024-08-19T03:40:17.373Z",
        "RoleName": "my-maintenance-window-role",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/my-maintenance-window-role"
    }
}
```

> **Note**
>
> Anote o `RoleName` e os valores de `Arn`. Você os inclui no próximo comando.

2. Execute o comando a seguir para anexar a política gerenciada pelo cliente ao perfil. Substitua o espaço reservado *`account-id`* pelo ID de sua Conta da AWS

   Linux & macOS

   ```
   aws iam attach-role-policy \
       --role-name "my-maintenance-window-role" \
       --policy-arn "arn:aws:iam::account-id:policy/mw-role-custom-policy.json"
   ```

   Windows

   ```
   aws iam attach-role-policy ^
       --role-name "my-maintenance-window-role" ^
       --policy-arn "arn:aws:iam::account-id:policy/mw-role-custom-policy.json"
   ```

3. Execute o comando a seguir para verificar se seu perfil foi criado e se a política de confiança foi anexada.

   ```
   aws iam get-role --role-name my-maintenance-window-role
   ```

   O comando retorna informações semelhantes às seguintes:

   ```
   {
       "Role": {
           "Path": "/",
           "RoleName": "my-maintenance-window-role",
   ```

```
        "RoleId": "AROA123456789EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:role/my-maintenance-window-role",
        "CreateDate": "2024-08-19T14:13:32+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "ssm.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        },
        "MaxSessionDuration": 3600,
        "RoleLastUsed": {
            "LastUsedDate": "2024-08-19T14:30:44+00:00",
            "Region": "us-east-2"
        }
    }
}
```

4. Execute o comando a seguir para verificar se a política gerenciada pelo cliente foi anexada ao perfil.

```
aws iam list-attached-role-policies --role-name my-maintenance-window-role
```

O comando retorna informações semelhantes às seguintes:

```
{
    "AttachedPolicies": [
        {
            "PolicyName": "mw-role-custom-policy",
            "PolicyArn": "arn:aws:iam::123456789012:policy/mw-role-custom-policy"
        }
    ]
}
```

Tarefa 3: conceda permissões a usuários especificados para que registrem tarefas da janela de manutenção usando a AWS CLI

Fornecer aos usuários permissões para acessar o perfil de serviço da janela de manutenção personalizado permite que eles a usem com suas tarefas de janelas de manutenção. Isso é além das permissões que você concedeu a eles para trabalhar com os comandos da API do Systems Manager para a ferramenta do Maintenance Windows. Esse perfil do IAM transmite as permissões necessárias para executar uma tarefa de janela de manutenção. Como resultado, um usuário não poderá registrar tarefas a uma janela de manutenção usando seu perfil de serviço personalizado sem a capacidade de passar essas permissões do IAM.

Ao registrar uma tarefa em uma janela de manutenção, você especifica um perfil de serviço para executar as operações de tarefas reais. Esta é a função que o serviço assumirá quando executar tarefas em seu nome. Antes disso, para registrar a própria tarefa, é necessário atribuir a política do IAM `PassRole` a uma entidade do IAM (como um usuário ou grupo). Isso permite que a entidade do IAM especifique, como parte do registro dessas tarefas na janela de manutenção, o perfil que deve ser usado ao executar as tarefas. Para obter informações, consulte Conceda permissões a um usuário para passar um perfil para um AWS service (Serviço da AWS), no Guia do usuário do IAM.

Para configurar permissões para usuários que tiverem permissão para registrar tarefas da janela de manutenção usando a AWS CLI

1. Copie e cole a seguinte política do AWS Identity and Access Management (IAM) em um editor de texto e salve-a com o seguinte nome e extensão de arquivo: `mw-passrole-policy.json`.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/my-maintenance-
window-role"
        },
        {
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "arn:aws:iam::111122223333:role/"
```

```
        },
        {
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "arn:aws:iam::111122223333:role/aws-service-role/
ssm.amazonaws.com/"
        }
    ]
}
```

Substituir*my-maintenance-window-role* pelo nome da função da janela de manutenção personalizada criada anteriormente.

Substitua *account-id* pelo ID da sua Conta da AWS. Adicionar essa permissão para o recurso `arn:aws:iam::`*account-id*`:role/` permite que os usuários do grupo visualizem e escolham entre funções de cliente no console ao criar uma tarefa da janela de manutenção. Adicionar essa permissão para `arn:aws:iam::`*account-id*`:role/aws-service-role/ssm.amazonaws.com/` permite que os usuários do grupo escolham a função vinculada ao serviço do Systems Manager no console ao criar uma tarefa da janela de manutenção.

2. Abra a AWS CLI.
3. Se você estiver atribuindo a permissão a uma entidade do IAM (usuário ou grupo), execute um dos seguintes comandos.

   - Para uma entidade do IAM:

     Linux & macOS

     ```
     aws iam put-user-policy \
         --user-name "user-name" \
         --policy-name "policy-name" \
         --policy-document file://path-to-document
     ```

     Windows

     ```
     aws iam put-user-policy ^
         --user-name "user-name" ^
         --policy-name "policy-name" ^
         --policy-document file://path-to-document
     ```

Em *user-name*, especifique o usuário do que atribui tarefas às janelas de manutenção. Em *policy-name*, especifique o nome que deseja usar para identificar a política, como **my-iam-passrole-policy**. Para *path-to-document*, especifique o caminho para o arquivo que você salvou na etapa 1. Por exemplo: `file://C:\Temp\mw-passrole-policy.json`

> **Note**
>
> Para conceder acesso a um usuário para registrar tarefas de janelas de manutenção usando o console do Systems Manager, também será necessário atribuir a política `AmazonSSMFullAccess` ao seu usuário (ou uma política do IAM que forneça um conjunto menor de permissões de acesso ao Systems Manager que abranja tarefas da janela de manutenção). Execute o seguinte comando para atribuir a política `AmazonSSMFullAccess` ao seu usuário.
>
> Linux & macOS
>
> ```
> aws iam attach-user-policy \
>     --policy-arn "arn:aws:iam::aws:policy/AmazonSSMFullAccess" \
>     --user-name "user-name"
> ```
>
> Windows
>
> ```
> aws iam attach-user-policy ^
>     --policy-arn "arn:aws:iam::aws:policy/AmazonSSMFullAccess" ^
>     --user-name "user-name"
> ```

- Para um grupo IAM:

  Linux & macOS

  ```
  aws iam put-group-policy \
      --group-name "group-name" \
      --policy-name "policy-name" \
      --policy-document file://path-to-document
  ```

  Windows

  ```
  aws iam put-group-policy ^
      --group-name "group-name" ^
  ```

```
    --policy-name "policy-name" ^
    --policy-document file://path-to-document
```

Em *group-name*, especifique o grupo do cujos membros atribuem tarefas a janelas de manutenção. Em *policy-name*, especifique o nome que deseja usar para identificar a política, como **my-iam-passrole-policy**. Para *path-to-document*, especifique o caminho para o arquivo que você salvou na etapa 1. Por exemplo: `file://C:\Temp\mw-passrole-policy.json`

> **Note**
>
> Para conceder acesso a membros de um grupo para registrar tarefas das janelas de manutenção usando o console do Systems Manager, também será necessário atribuir a política `AmazonSSMFullAccess` ao grupo. Execute o seguinte comando para atribuir essa política ao seu grupo.
>
> Linux & macOS
>
> ```
> aws iam attach-group-policy \
>     --policy-arn "arn:aws:iam::aws:policy/AmazonSSMFullAccess" \
>     --group-name "group-name"
> ```
>
> Windows
>
> ```
> aws iam attach-group-policy ^
>     --policy-arn "arn:aws:iam::aws:policy/AmazonSSMFullAccess" ^
>     --group-name "group-name"
> ```

4. Execute o seguinte comando para verificar se a política foi atribuída ao grupo.

   Linux & macOS

   ```
   aws iam list-group-policies \
       --group-name "group-name"
   ```

   Windows

   ```
   aws iam list-group-policies ^
       --group-name "group-name"
   ```

Tarefa 4: evite que usuários especificados registrem tarefas da janela de manutenção usando a AWS CLI

É possível negar a permissão `ssm:RegisterTaskWithMaintenanceWindow` aos usuários da sua Conta da AWS para os quais você não quer registrar tarefas com janelas de manutenção. Isso fornece uma camada extra de prevenção para usuários que não devem registrar tarefas da janela de manutenção.

Se você estiver recusando a permissão `ssm:RegisterTaskWithMaintenanceWindow` a um usuário individual ou a um grupo, use um dos procedimentos a seguir para impedir que os usuários registrem tarefas com uma janela de manutenção.

Para configurar permissões para usuários sem permissão para registrar tarefas da janela de manutenção usando a AWS CLI

1. Copie e cole a seguinte política do IAM em um editor de texto e salve-a com o seguinte nome e extensão de arquivo: **`deny-mw-tasks-policy.json`**.

   JSON

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Effect": "Deny",
               "Action": "ssm:RegisterTaskWithMaintenanceWindow",
               "Resource": "*"
           }
       ]
   }
   ```

2. Abra a AWS CLI.
3. Se você estiver atribuindo a permissão a uma entidade do IAM (usuário ou grupo), execute um dos seguintes comandos.

   - Para um usuário:

     Linux & macOS

     ```
     aws iam put-user-policy \
         --user-name "user-name" \
     ```

```
    --policy-name "policy-name" \
    --policy-document file://path-to-document
```

Windows

```
aws iam put-user-policy ^
    --user-name "user-name" ^
    --policy-name "policy-name" ^
    --policy-document file://path-to-document
```

Em *user-name*, especifique o usuário para impedir a atribuição de tarefas às janelas de manutenção. Em *policy-name*, especifique o nome que deseja usar para identificar a política, como **my-deny-mw-tasks-policy**. Para *path-to-document*, especifique o caminho para o arquivo que você salvou na etapa 1. Por exemplo: `file://C:\Temp\deny-mw-tasks-policy.json`

- Para um grupo:

  Linux & macOS

```
aws iam put-group-policy \
    --group-name "group-name" \
    --policy-name "policy-name" \
    --policy-document file://path-to-document
```

  Windows

```
aws iam put-group-policy ^
    --group-name "group-name" ^
    --policy-name "policy-name" ^
    --policy-document file://path-to-document
```

  Em *group-name*, especifique o grupo para impedir a atribuição de tarefas às janelas de manutenção. Em *policy-name*, especifique o nome que deseja usar para identificar a política, como **my-deny-mw-tasks-policy**. Para *path-to-document*, especifique o caminho para o arquivo que você salvou na etapa 1. Por exemplo: `file://C:\Temp\deny-mw-tasks-policy.json`

4. Execute o seguinte comando para verificar se a política foi atribuída ao grupo.

Linux & macOS

```
aws iam list-group-policies \
    --group-name "group-name"
```

Windows

```
aws iam list-group-policies ^
    --group-name "group-name"
```

Crie e gerencie janelas de manutenção usando o console

Esta seção descreve como criar, configurar, atualizar e excluir janelas de manutenção usando o console do AWS Systems Manager. Esta seção também fornece informações sobre como gerenciar os destinos e tarefas de uma janela de manutenção.

> **Important**
>
> Recomendamos que você crie e configure inicialmente as janelas de manutenção em um ambiente de teste.

Antes de começar

Antes de criar uma janela de manutenção, você deve configurar o acesso ao Maintenance Windows, uma ferramenta do AWS Systems Manager. Para ter mais informações, consulte Configurar o Maintenance Windows.

Tópicos

- Crie uma janela de manutenção usando o console
- Atribua destinos a uma janela de manutenção usando o console
- Atribua tarefas a uma janela de manutenção usando o console
- Desabilite ou habilite uma janela de manutenção usando o console
- Atualize ou exclua recursos da janela de manutenção usando o console

Crie uma janela de manutenção usando o console

Neste procedimento, você cria uma janela de manutenção na Maintenance Windows, uma ferramenta do AWS Systems Manager. Especifique as opções básicas, como nome, programação e duração. Nas etapas posteriores, você escolherá os destinos, ou recursos, que ela atualizará e as tarefas que serão executadas quando a janela de manutenção for executada.

> **Note**
>
> Para obter uma explicação de como as várias opções relacionadas à programação de janelas de manutenção se relacionam entre si, consulte Opções de programação da janela de manutenção e do período ativo.
> Para obter mais informações sobre como trabalhar com a opção `--schedule`, consulte Referência: Expressões cron e rate para o Systems Manager.

Para criar uma janela de manutenção usando o console

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Maintenance Windows.
3. Escolha Create maintenance window (Criar janela de manutenção).
4. Em Name (Nome), insira um nome descritivo para ajudar você a identificar essa janela de manutenção.
5. (Opcional) Em Description (Descrição), insira uma descrição para identificar como essa janela de manutenção será usada.
6. (Opcional) se você deseja permitir que uma tarefa de janela de manutenção seja executada em nós gerenciados, mesmo que você não tenha registrado esses nós como destinos, escolha Allow unregistered targets (Permitir destinos não registrados).

   Se você escolher essa opção, poderá escolher os nós não registrados (por ID do nó) quando registrar uma tarefa na janela de manutenção.

   Se você não escolher essa opção, deverá escolher destinos anteriormente registrados quando registrar uma tarefa na janela de manutenção.
7. Especifique uma programação para a janela de manutenção usando uma das opções de programação.

Para obter mais informações sobre criar expressões cron/rate, consulte Referência: Expressões cron e rate para o Systems Manager.

8. Em Duration (Duração), insira o número de horas que a janela de manutenção será executada. O valor especificado determina a hora de término específica para a janela de manutenção com base no horário em que ela começa. Nenhuma tarefa da janela de manutenção tem permissão para iniciar após a hora de término resultante menos o número de horas especificado para Stop initiating tasks (Parar de iniciar tarefas) na próxima etapa.

   Por exemplo, se a janela de manutenção começar às 15h, a duração for de três horas e o valor Stop initiating tasks (Parar de iniciar tarefas) for uma hora, nenhuma tarefa da janela de manutenção poderá ser iniciada depois das 17h.

9. Em Stop initiating tasks (Para de iniciar tarefas), insira o número de horas antes do final da janela de manutenção que o sistema deve parar de agendar novas tarefas para execução.

10. (Opcional) Em Window start date (Data de início da janela), especifique uma data e hora no formato ISO-8601 estendido para quando você deseja que a janela de manutenção se torne ativa. Isso permite que você atrase a ativação da janela de manutenção até a data futura especificada.

    > **Note**
    >
    > Não é possível especificar uma data e hora de início que ocorreram no passado.

11. (Opcional) Em Window end date (Data de término da janela), especifique uma data e hora no formato ISO-8601 estendido para quando você deseja que a janela de manutenção se torne inativa. Isso permite que você defina uma data e hora no futuro após a qual a janela de manutenção não será mais executada.

12. (Opcional) Em Schedule time zone (Fuso horário da programação), especifique o fuso horário a ser usado como base para quando a janela de manutenção for executada, no formato IANA (Internet Assigned Numbers Authority). Por exemplo: "America/Los_Angeles", "etc/UTC" ou "Ásia/Seul".

    Para obter mais informações sobre os formatos válidos, consulte o Banco de dados de fusos horários no site da IANA.

13. (Opcional) em Schedule offset (Deslocamento de programação), insira o número de dias de espera após a data e a hora especificadas por uma expressão de cron ou rate antes de executar a janela de manutenção. Você pode especificar entre um e seis dias.

> **Note**
>
> Essa opção estará disponível somente se você tiver especificado uma programação inserindo uma expressão cron ou rate manualmente.

14. (Opcional) Na área Manage tags (Gerenciar tags), aplique um ou mais pares de nome/valor de chave de tag à janela de manutenção.

    Tags são metadados opcionais que você atribui a um recurso. As tags permitem categorizar um recurso de diferentes formas, como por finalidade, proprietário ou ambiente. Por exemplo, você pode querer marcar uma janela de manutenção para identificar o tipo de tarefa que ela executa, os tipos de destinos e o ambiente em que ela é executada. Nesse caso, você pode especificar os seguintes pares de nome/valor:

    - `Key=TaskType,Value=AgentUpdate`
    - `Key=OS,Value=Windows`
    - `Key=Environment,Value=Production`
15. Escolha Create maintenance window (Criar janela de manutenção). O sistema fará com que você retorne para a página de janela de manutenção. O estado da janela de manutenção que você acabou de criar é Enabled (Habilitada).

Atribua destinos a uma janela de manutenção usando o console

Neste procedimento, você registra um destino em uma janela de manutenção. Em outras palavras, você especifica em quais recursos a janela de manutenção executa ações.

> **Note**
>
> Se uma única tarefa da janela de manutenção for registrada com vários destinos, suas chamadas de tarefa ocorrerão sequencialmente e não em paralelo. Se a tarefa precisar ser executada em vários destinos ao mesmo tempo, registre uma tarefa para cada destino individualmente e atribua a cada uma o mesmo nível de prioridade.

Para atribuir destinos a uma janela de manutenção usando o console

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.

2. No painel de navegação, escolha Maintenance Windows.
3. Na lista de janelas de manutenção, escolha a janela de manutenção à qual adicionar destinos.
4. Escolha Actions (Ações) e depois Register targets (Registrar destinos).
5. (Opcional) Em Target name (Nome do destino), insira um nome para os destinos.
6. (Opcional) Em Description (Descrição), insira uma descrição.
7. (Opcional) Em Owner Information (Informações do proprietário), especifique informações para incluir em qualquer evento do Amazon EventBridge gerado durante a execução de tarefas para esses destinos nessa janela de manutenção.

   Para obter informações sobre como usar o EventBridge para monitorar eventos do Systems Manager, consulte Monitorar eventos do Systems Manager com o Amazon EventBridge.
8. Na área Targets (Destinos), escolha uma das opções descritas na tabela a seguir.

Opção	Descrição
Especificar tags de instâncias	Nas caixas Specify instance tags (Especificar tags de instâncias), especifique uma ou mais chaves de tags e valores (opcional) que foram ou serão adicionados aos nós gerenciados em sua conta. Quando a janela de manutenção for executada, ele tentará executar tarefas em todos os nós gerenciados aos quais essas tags foram adicionadas. Se você especificar mais de uma chave de tag, um nó deverá ser marcado com todas as chaves de tag e os valores especificados para serem incluídos no grupo de destino.
Escolher instâncias manualmente	Na lista, marque a caixa de seleção para cada nó que você deseja incluir na janela de manutenção de destino. A lista inclui todos os nós da sua conta que estão configurados para uso com o Systems Manager.

Opção	Descrição
	Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas. Para dispositivos de borda, servidores e máquinas virtuais (VMs) on-premises, consulte Gerenciar nós em ambientes híbridos e multinuvem com o Systems Manager

Opção	Descrição
Escolher um grupo de recursos	Em Resource group (Grupo de recursos), escolha o nome de um grupo de recursos existente em sua conta na lista. Para obter informações sobre como criar e trabalhar com grupos de recursos, consulte os seguintes tópicos: • O que são grupos de recursos? no Guia do usuário do AWS Resource Groups • Resource Groups and Tagging for AWS (Grupos de recursos e marcação para a AWS) no Blog de notícias da Em Resource types (Tipos de recursos) , selecione até cinco tipos de recursos disponíveis ou escolha All resource types (Todos os tipos de recursos). Se as tarefas que você atribui para a janela de manutenção não atuar em um dos tipos de recurso que você adicionou ao destino, o sistema poderá relatar um erro. As tarefas para as quais um tipo de recurso compatíve l é encontrado continuam a ser executadas apesar desses erros. Por exemplo, suponha que você adicione os seguintes tipos de recurso para este destino: • `AWS::S3::Bucket` • `AWS::DynamoDB::Table` • `AWS::EC2::Instance`

Opção	Descrição
	Mas posteriormente, quando você adicionar tarefas à janela de manutenção, você incluirá apenas as tarefas que executam ações em nós, como a aplicação de uma lista de referência de patches ou a reinicialização de um nó. No log da janela de manutenção, um erro poderá ser relatado sobre os buckets do Amazon Simple Storage Service (Amazon S3) ou as tabelas do Amazon DynamoDB que foram encontrados. No entanto, a janela de manutenção ainda executará tarefas em nós do grupo de recursos.

9. Escolha Register target.

Se quiser atribuir mais destinos a essa janela de manutenção, escolha a guia Targets (Destinos) e escolha Register new targets (Registrar destinos novos). Com essa opção, você pode escolher um meio diferente de direcionamento. Por exemplo, se você definiu nós gerenciados de destino anteriormente por ID do nó, poderá registrar novos destinos e nós de destino especificando as tags aplicadas aos nós gerenciados ou escolhendo tipos de recurso de um grupo de recursos.

Atribua tarefas a uma janela de manutenção usando o console

Neste procedimento, você adicionará uma tarefa a uma janela de manutenção. Tarefas são as ações realizadas quando uma janela de manutenção é executada.

Os quatro tipos de tarefas a seguir podem ser adicionados a uma janela de manutenção:

- Comandos do Run Command do AWS Systems Manager
- Fluxos de trabalho do Systems Manager Automation
- Tarefas do AWS Step Functions
- Funções do AWS Lambda

> **Important**
>
> A política do Maintenance Windows para IAM requer a adição do prefixo `SSM` aos nomes das funções Lambda (ou alias) . Antes de prosseguir com o registro desse tipo de tarefa, atualize o nome no AWS Lambda para incluir `SSM`. Por exemplo, se o nome da função Lambda for `MyLambdaFunction`, altere-o para `SSMMyLambdaFunction`.

Para atribuir tarefas a uma janela de manutenção

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Maintenance Windows.
3. Na lista de janelas de manutenção, escolha uma janela de manutenção.
4. Escolha Actions (Ações) e selecione a opção para o tipo de tarefa que deseja registrar na janela de manutenção.
   - Escolha Register run command task (Registrar tarefa de comando de execução).
   - Escolha Register Automation task (Registrar tarefa de automação).
   - Register Lambda task (Registrar tarefa do Lambda)
   - Register Step Functions task (Registrar tarefa do Step Functions)

   > **Note**
   >
   > As tarefas da janela de manutenção oferecem suporte somente aos fluxos de trabalho da máquina de estado Step Functions Standard. Elas não oferecem suporte a fluxos de trabalho de máquinas de estado Express. Para obter informações sobre os tipos de fluxo de trabalho da máquina de estado, consulte Fluxos de trabalho Standard vs. Express no Guia do desenvolvedor do AWS Step Functions.
5. (Opcional) Em Name (Nome), insira um nome para a tarefa.
6. (Opcional) Em Description (Descrição), insira uma descrição.
7. Para New task invocation cutoff (Novo corte de invocação de tarefas), se você não quiser que novas invocações de tarefas sejam iniciadas após o tempo de corte da janela de manutenção ser atingido, escolha Enabled (Habilitado).

Quando esta opção não for ativada, a tarefa continua sendo executada quando o tempo limite for atingido e inicia novas invocações de tarefa até a conclusão.

> **Note**
>
> O status das tarefas que não são concluídas quando você habilita essa opção é `TIMED_OUT`.

8. Para essa etapa, siga as subetapas aplicáveis ao tipo de tarefa selecionado.

   Executar comando

   1. Na lista Documento de comandos, escolha o documento do Systems Manager Command (documento do SSM) que define as tarefas que devem ser executadas.
   2. Em Document version (Versão do documento), escolha a versão do documento a ser usada.
   3. Em Task priority (Prioridade da tarefa), especifique uma prioridade para essa tarefa. Zero (`0`) é a prioridade mais alta. As tarefas em uma janela de manutenção são programadas em ordem de prioridade, com as tarefas que têm a mesma prioridade programada em paralelo.

   Automation

   1. Na lista Documento do Automation, escolha o runbook do Automation que define as tarefas que devem ser executadas.
   2. Em Document version (Versão do documento), escolha a versão do runbook a ser usada.
   3. Em Task priority (Prioridade da tarefa), especifique uma prioridade para essa tarefa. Zero (`0`) é a prioridade mais alta. As tarefas em uma janela de manutenção são programadas em ordem de prioridade, com as tarefas que têm a mesma prioridade programada em paralelo.

   Lambda

   1. Na área Parâmetros do Lambda) escolha uma função do Lambda na lista.
   2. (Opcional) Forneça qualquer conteúdo para Payload (Carga útil), Client Context (Contexto do cliente) ou Qualifier (Qualificador) que você deseje incluir.

> **Note**
>
> Em alguns casos, você pode usar um pseudoparâmetro como parte do valor de `Payload`. Quando a tarefa de janela de manutenção é executada, ela envia os valores corretos em vez dos espaços reservados do pseudoparâmetro. Para mais informações, consulte Usar pseudoparâmetros ao registrar tarefas da janela de manutenção.

3. Em Task priority (Prioridade da tarefa), especifique uma prioridade para essa tarefa. Zero (`0`) é a prioridade mais alta. As tarefas em uma janela de manutenção são programadas em ordem de prioridade, com as tarefas que têm a mesma prioridade programada em paralelo.

Step Functions

1. Na área Parâmetros de Step Functions, escolha uma máquina de estado na lista.
2. (Opcional) Forneça um nome para a execução da máquina de estado e qualquer conteúdo para a Input (Entrada) que você deseja incluir.

> **Note**
>
> Em alguns casos, você pode usar um pseudoparâmetro como parte do valor de `Input`. Quando a tarefa de janela de manutenção é executada, ela envia os valores corretos em vez dos espaços reservados do pseudoparâmetro. Para mais informações, consulte Usar pseudoparâmetros ao registrar tarefas da janela de manutenção.

3. Em Task priority (Prioridade da tarefa), especifique uma prioridade para essa tarefa. Zero (`0`) é a prioridade mais alta. As tarefas em uma janela de manutenção são programadas em ordem de prioridade, com as tarefas que têm a mesma prioridade programada em paralelo.

9. Na área Targets (Destinos), escolha uma das seguintes opções:

- Selecione grupos de destinos registrados: selecione um ou mais destinos da janela de manutenção que estejam registrados na janela de manutenção atual.

- Selecione destinos não registrados: escolha os recursos disponíveis um por um, como destinos para a tarefa.

  Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

- Destino da tarefa não obrigatório: os destinos para a tarefa já podem estar especificados em outras funções para todos, menos para as tarefas do tipo Run Command.

  Especifique um ou mais destinos para as tarefas da janela de manutenção do tipo Run Command. Dependendo da tarefa, os destinos serão opcionais para outros tipos de tarefas da janela de manutenção (Automation, AWS Lambda e AWS Step Functions). Para obter mais informações sobre como executar tarefas que não especificam destinos, consulte Registrar tarefas da janela de manutenção sem destinos.

  > **Note**
  >
  > Em muitos casos, você não precisa especificar explicitamente um destino para uma tarefa de automação. Por exemplo, digamos que você esteja criando uma tarefa do tipo Automation para atualizar uma Amazon Machine Image (AMI) para Linux, usando o runbook `AWS-UpdateLinuxAmi`. Quando a tarefa for executada, a AMI será atualizada com os pacotes de distribuição Linux e o software da Amazon mais recentes disponível. As novas instâncias criadas na AMI já têm essas atualizações instaladas. Como o ID da AMI a ser atualizado é especificado nos parâmetros de entrada para o runbook, não há necessidade de especificar um destino novamente na tarefa da janela de manutenção.

10. Somente tarefas do Automation:

    Na área Input parameters (Parâmetros de entrada), forneça valores para quaisquer parâmetros obrigatórios ou opcionais necessários para executar sua tarefa.

    > **Note**
    >
    > Em alguns casos, você pode usar um pseudoparâmetro para determinados valores de parâmetros de entrada. Quando a tarefa de janela de manutenção é executada, ela envia os valores corretos em vez dos espaços reservados do pseudoparâmetro. Para

mais informações, consulte Usar pseudoparâmetros ao registrar tarefas da janela de manutenção.

11. Para Rate control (Controle de taxa):

    - Em Concurrency (Concorrência), especifique um número ou uma porcentagem de nós gerenciados nos quais executar o comando ao mesmo tempo.

      > **Note**
      >
      > Se você selecionou destinos especificando tags aplicadas a instâncias a nós gerenciados ou especificando grupos de recursos da AWS, e não tiver certeza de quantas instâncias são direcionadas, restrinja o número de instâncias que poderão executar o documento ao mesmo tempo, especificando uma porcentagem.

    - Em Error threshold (Limite de erro), especifique quando parar de executar o comando em outros nós depois de falhar em alguns ou em uma porcentagem de nós. Por exemplo, se você especificar três erros, o Systems Manager deixará de enviar o comando quando o 4° erro for recebido. Os nós gerenciados que continuam processando o comando também podem enviar erros.

12. (Opcional) Em Perfil de serviço do IAM, escolha um perfil para fornecer permissões ao Systems Manager para assumir quando executar uma tarefa da janela de manutenção.

    Se você não especificar um ARN de perfil de serviço, o Systems Manager usará um perfil vinculado ao serviço em sua conta. Este perfil não está listado no menu suspenso. Se nenhum perfil vinculado ao serviço apropriado para Systems Manager existir em sua conta, ele será criado quando a tarefa for registrada com êxito.

    > **Note**
    >
    > Para melhorar a postura de segurança, é altamente recomendável criar uma política personalizada e um perfil de serviço personalizado para executar as tarefas da janela de manutenção. A política pode ser criada para fornecer somente as permissões necessárias para as tarefas da sua janela de manutenção específica. Para obter mais informações, consulte Configurar o Maintenance Windows.

13. Somente tarefas do Run Command:

(Opcional) Em Output options (Opções de saída), faça o seguinte:

- Selecione a opção Enable writing to S3 (Ativar gravação no S3) para salvar a saída do comando em um arquivo. Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.
- Selecione a caixa de verificação saída do CloudWatch para gravar a saída completa no Amazon CloudWatch Logs Insira o nome do grupo de logs do CloudWatch Logs.

> **Note**
>
> As permissões que garantem a possibilidade de gravar dados em um bucket do S3 ou no CloudWatch Logs são as do perfil da instância atribuído ao nó, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager. Além disso, se o bucket do S3 ou grupo de logs especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância associada ao nó tem as permissões necessárias para gravar nesse bucket.

14. Somente tarefas do Run Command:

    Na seção SNS notifications (Notificações do SNS), se quiser enviar notificações sobre o status da execução do comando, marque a caixa de seleção Enable SNS notifications (Habilitar notificações do SNS).

    Para obter mais informações sobre a configuração de notificações do Amazon SNS para o Run Command, consulte Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS.

15. Somente tarefas do Run Command:

    Na área Parameters (Parâmetros), especifique os parâmetros para o documento.

> **Note**
>
> Em alguns casos, você pode usar um pseudoparâmetro para determinados valores de parâmetros de entrada. Quando a tarefa de janela de manutenção é executada, ela envia os valores corretos em vez dos espaços reservados do pseudoparâmetro. Para mais informações, consulte Usar pseudoparâmetros ao registrar tarefas da janela de manutenção.

16. Somente tarefas do Run Command e do Automation:

    (Opcional) Na área Alarme do CloudWatch, em Nome do alarme, escolha um alarme do CloudWatch existente para aplicar à sua tarefa para monitoramento.

    Se o alarme for ativado, a tarefa será interrompida.

    > **Note**
    >
    > Para anexar um alarme do CloudWatch à sua tarefa, a entidade principal do IAM que executa a tarefa deve ter permissão para a ação `iam:createServiceLinkedRole`. Para obter mais informações sobre alarmes do CloudWatch, consulte Usar alarmes do Amazon CloudWatch.

17. Dependendo do tipo da tarefa, escolha uma das seguintes opções:

    - Escolha Register run command task (Registrar tarefa de comando de execução).
    - Escolha Register Automation task (Registrar tarefa de automação).
    - Register Lambda task (Registrar tarefa do Lambda)
    - Register Step Functions task (Registrar tarefa do Step Functions)

Desabilite ou habilite uma janela de manutenção usando o console

Você pode desabilitar ou habilitar uma janela de manutenção no Maintenance Windows, uma ferramenta do AWS Systems Manager. Você pode escolher uma janela de manutenção por vez para desabilitar ou habilitar a execução da janela de manutenção. Também é possível selecionar várias ou todas as janelas de manutenção para habilitar e desabilitar.

Esta seção descreve como desabilitar ou habilitar uma janela de manutenção usando o console do Systems Manager. Para obter exemplos de como fazer isso usando a AWS Command Line Interface (AWS CLI), consulte Tutorial: Atualizar uma janela de manutenção usando a AWS CLI.

Tópicos

- Como desabilitar a janela de manutenção usando o console
- Habilite a janela de manutenção usando o console

Como desabilitar a janela de manutenção usando o console

Você pode desativar uma janela de manutenção para pausar uma tarefa por um período especificado, e ela continuará disponível para ser habilitada novamente mais tarde.

Para desabilitar uma janela de manutenção

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Maintenance Windows.
3. Usando a caixa de seleção ao lado da janela de manutenção que você quer desabilitar, selecione uma ou mais janelas de manutenção.
4. Escolha Desabilitar janela de manutenção no menu Ações. O sistema solicitará que você confirme suas ações.

Habilite a janela de manutenção usando o console

É possível habilitar uma janela de manutenção para retomar uma tarefa.

> **Note**
>
> Se a janela de manutenção usar uma tabela de tarifas e a data de início estiver atualmente definida como data e hora passadas, a data e a hora atuais serão usadas como a data de início da janela de manutenção. É possível alterar a data de início da janela de manutenção antes ou depois de habilitá-la. Para ter mais informações, consulte Atualize ou exclua recursos da janela de manutenção usando o console.

Para habilitar uma janela de manutenção

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Maintenance Windows.
3. Marque a caixa de seleção próxima à janela de manutenção a ser habilitada.
4. Escolha Ações, Habilitar janela de manutenção. O sistema solicitará que você confirme suas ações.

Atualize ou exclua recursos da janela de manutenção usando o console

Você pode atualizar ou excluir uma janela de manutenção no Maintenance Windows, uma ferramenta do AWS Systems Manager. Também pode atualizar ou excluir os destinos ou as tarefas de uma janela de manutenção. Se você editar os detalhes de uma janela de manutenção, poderá alterar o a programação, os destinos e as tarefas. Você também pode especificar nomes e descrições para janelas, destinos e tarefas, o que o ajuda a entender melhor a finalidade e facilita o gerenciamento da sua fila de janelas.

Esta seção descreve como atualizar ou excluir uma janela de manutenção, destinos e tarefas usando o console do Systems Manager. Para obter exemplos de como fazer isso usando a AWS Command Line Interface (AWS CLI), consulte Tutorial: Atualizar uma janela de manutenção usando a AWS CLI.

Tópicos

- Atualização ou exclusão da janela de manutenção usando o console
- Atualização ou cancelamento de registros de destinos da janela de manutenção usando o console
- Atualização ou cancelamento de registros de tarefas da janela de manutenção usando o console

Atualização ou exclusão da janela de manutenção usando o console

É possível atualizar uma janela de manutenção para alterar o nome, a descrição e a programação e se a janela de manutenção deve permitir destinos não registrados.

Para atualizar ou excluir uma janela de manutenção

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Maintenance Windows.
3. Escolha o botão próximo à janela de manutenção que você quer atualizar ou excluir e execute uma das seguintes ações:

   - Escolha Excluir. O sistema solicitará que você confirme suas ações.
   - Escolha Editar. Na página Edit maintenance window (Editar janela de manutenção), altere as opções e os valores desejados e escolha Save changes (Salvar alterações).

     Para obter informações sobre as opções de configuração que podem ser feitas, consulte Crie uma janela de manutenção usando o console.

Atualização ou cancelamento de registros de destinos da janela de manutenção usando o console

Você pode atualizar ou cancelar registros dos destinos de uma janela de manutenção. Se você optar por atualizar um destino da janela de manutenção, poderá especificar um novo nome, uma descrição e um proprietário para esse destino. Também é possível escolher diferentes destinos.

Para atualizar ou excluir os destinos de uma janela de manutenção

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Maintenance Windows.
3. Escolha o nome da janela de manutenção que deseja atualizar, escolha a guia Targets (Destinos) e execute uma das seguintes ações:
   - Para atualizar os destinos, selecione o botão ao lado do destino a ser atualizado e escolha Edit (Editar).
   - Para excluir registros de destinos, selecione o botão ao lado do destino a cancelar o registro e, em seguida, escolha Deregister targets (Cancelar registro do destino). Na caixa de diálogo Deregister maintenance windows target (Cancelar destino da janela de manutenção), escolha Deregister (Cancelar registro).

Atualização ou cancelamento de registros de tarefas da janela de manutenção usando o console

Você pode atualizar ou cancelar registros das tarefas de uma janela de manutenção. Se você optar por atualizar, poderá especificar um novo nome de tarefa, uma descrição e um proprietário. Para tarefas do Run Command e do Automation, você pode escolher um documento do SSM diferente para as tarefas. No entanto, não é possível editar uma tarefa para alterar seu tipo. Por exemplo, se você tiver criado uma tarefa do Automation, não poderá editá-la e alterá-la para uma tarefa do Run Command.

Para atualizar ou excluir as tarefas de uma janela de manutenção usando o console

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Maintenance Windows.
3. Escolha o nome a janela de manutenção que você deseja atualizar.
4. Escolha a guia Tasks (Tarefas) e, em seguida, selecione o botão ao lado da tarefa a ser atualizada.
5. Execute um destes procedimentos:

- Para cancelar o registro de uma tarefa, escolha Deregister task (Cancelar o registro de tarefa).
- Para editar a tarefa, escolha Edit (Editar). Altere os valores e as opções desejados e escolha Edit Task (Editar tarefa).

Tutoriais

Os tutoriais nesta seção mostram como realizar tarefas comuns por meio de janelas de manutenção.

Concluir os pré-requisitos

Antes de tentar estes tutoriais, conclua os seguintes pré-requisitos.

- Configure a AWS CLI em sua máquina local: antes de poder executar comandos da AWS CLI, você deve instalar e configurar a CLI em sua máquina local. Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI e Instalar o Ferramentas da AWS para PowerShell.
- Verifique as funções e as permissões da janela de manutenção: um administrador da AWS em sua conta deve conceder a você as permissões do AWS Identity and Access Management (IAM) necessárias para gerenciar as janelas de manutenção usando a CLI. Para ter mais informações, consulte Configurar o Maintenance Windows.
- Crie ou configure uma instância que é compatível com o Systems Manager: é necessário, no mínimo, uma instância do Amazon Elastic Compute Cloud (Amazon EC2) configurada para uso com o Systems Manager para concluir os tutoriais. Isso significa que o SSM Agent é instalado na instância e um perfil de instância do IAM para o Systems Manager é anexado à instância.

  Recomendamos iniciar uma instância a partir de uma Amazon Machine Image (AMI) gerenciada pela AWS com o agente pré-instalado. Para ter mais informações, consulte Encontrar AMIs com o SSM Agent pré-instalado.

  Para obter informações sobre como instalar o SSM Agent em uma instância, consulte os seguintes tópicos:

  - Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Windows Server
  - Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux

  Para obter informações sobre como configurar as permissões do IAM para o Systems Manager em sua instância, consulte Configurar permissões de instância obrigatórias para o Systems Manager.

- Crie recursos adicionais, conforme necessário: Run Command, uma ferramenta do Systems Manager, inclui diversas tarefas que não exigem a criação de recursos diferentes dos listados neste tópico de pré-requisitos. Por esse motivo, nós fornecemos uma tarefa simples de Run Command para você usar pela primeira vez nos tutoriais. Você também precisará de uma instância do EC2 configurada para uso com o Systems Manager, conforme descrito anteriormente neste tópico. Depois de configurar essa instância, você pode registrar uma tarefa simples do Run Command.

  A ferramenta Maintenance Windows do Systems Manager oferece suporte à execução dos quatro tipos de tarefas a seguir:

  - Comandos Run Command
  - Fluxos de trabalho do Systems Manager Automation
  - Funções do AWS Lambda
  - Tarefas do AWS Step Functions

  No geral, se uma tarefa de janela de manutenção que você deseja executar exigir recursos adicionais, você deverá criá-los primeiro. Por exemplo, se você quiser uma janela de manutenção que execute uma função do AWS Lambda, crie a função do Lambda antes de começar. Para uma tarefa do Run Command, crie o bucket do S3 no qual você pode salvar a saída do comando (se pretender fazer isso), e assim por diante.

Tutoriais

- Tutorials: Create and manage maintenance windows using the AWS CLI
- Tutorial: Create a maintenance window for patching using the console

Tutorials: Create and manage maintenance windows using the AWS CLI

Esta seção inclui tutoriais que ensinarão como usar a AWS Command Line Interface (AWS CLI) para fazer o seguinte:

- Criar e configurar uma janela de manutenção
- Visualizar informações sobre uma janela de manutenção
- Visualizar informações sobre tarefas de janelas de manutenção e execuções de tarefas
- Atualizar uma janela de manutenção
- Excluir uma janela de manutenção

Acompanhar os IDs de recursos

À medida que as tarefas nestes tutoriais da AWS CLI forem concluídas, acompanhe os IDs de recursos gerados pelos comandos executados. Muitos deles são usados como entrada para comandos subsequentes. Por exemplo, ao criar a janela de manutenção, o sistema fornece um ID de janela de manutenção no formato a seguir.

```
{
   "WindowId":"mw-0c50858d01EXAMPLE"
}
```

Anote os seguintes IDs gerados pelo sistema, pois são usados pelos tutoriais desta seção:

- `WindowId`
- `WindowTargetId`
- `WindowTaskId`
- `WindowExecutionId`
- `TaskExecutionId`
- `InvocationId`
- `ExecutionId`

Você também precisa do ID da instância do EC2 que planeja usar nos tutoriais. Por exemplo: `i-02573cafcfEXAMPLE`

Tutoriais

- Tutorial: Criar e configurar uma janela de manutenção usando a AWS CLI
- Tutorial: Visualizar informações sobre janelas de manutençãousando a AWS CLI
- Tutorial: Visualizar informações sobre tarefas e execuções de tarefas usando a AWS CLI
- Tutorial: Atualizar uma janela de manutenção usando a AWS CLI
- Tutorial: Excluir uma janela de manutenção usando a AWS CLI

Tutorial: Criar e configurar uma janela de manutenção usando a AWS CLI

Este tutorial demonstra como usar a AWS Command Line Interface (AWS CLI) para criar e configurar uma janela de manutenção, seus destinos e suas tarefas. O caminho principal do tutorial consiste em etapas simples. Crie uma única janela de manutenção, identifique um único destino e configure

uma tarefa simples para a execução da janela de manutenção. Durante o processo, forneceremos informações que poderão ser usadas para testar cenários mais complicados.

Ao seguir as etapas neste tutorial, substitua os valores em texto itálico *vermelho* por suas próprias opções e IDs. Por exemplo, substitua o ID da janela de manutenção *mw-0c50858d01EXAMPLE* e o ID da instância*i-02573cafcfEXAMPLE* pelos IDs de recursos criados.

Conteúdo

- Etapa 1: Criar a janela de manutenção usando a AWS CLI
- Etapa 2: Registrar um nó de destino na janela de manutenção usando a AWS CLI
- Etapa 3: Registrar uma tarefa na janela de manutenção usando a AWS CLI

Etapa 1: Criar a janela de manutenção usando a AWS CLI

Nesta etapa, crie uma janela de manutenção e especifique suas opções básicas, como nome, programação e duração. Nas etapas posteriores, você escolherá a instância que ela atualiza e a tarefa que ela executa.

No nosso exemplo, você criará uma janela de manutenção que é executada a cada cinco minutos. Normalmente, uma janela de manutenção não seria executada com essa frequência. No entanto, essa taxa permite visualizar os resultados do tutorial rapidamente. Mostraremos como alterar para uma taxa menos frequente após a execução bem-sucedida da tarefa.

> **Note**
>
> Para obter uma explicação de como as várias opções relacionadas à programação de janelas de manutenção se relacionam entre si, consulte Opções de programação da janela de manutenção e do período ativo.
> Para obter mais informações sobre como trabalhar com a opção `--schedule`, consulte Referência: Expressões cron e rate para o Systems Manager.

Para criar uma janela de manutenção usando a AWS CLI

1. Abra a AWS Command Line Interface (AWS CLI) e execute o seguinte comando na máquina local para criar uma janela de manutenção que faz o seguinte:

   - Executa a cada cinco minutos durante até duas horas (conforme necessário).

- Impede que novas tarefas iniciem a menos de uma hora do final da operação da janela de manutenção.
- Permite destinos não associados (instâncias que não foram registradas na janela de manutenção).
- Indica, por meio do uso de tags personalizadas, que seu criador pretende usá-la em um tutorial.

Linux & macOS

```
aws ssm create-maintenance-window \
    --name "My-First-Maintenance-Window" \
    --schedule "rate(5 minutes)" \
    --duration 2 \
    --cutoff 1 \
    --allow-unassociated-targets \
    --tags "Key=Purpose,Value=Tutorial"
```

Windows

```
aws ssm create-maintenance-window ^
    --name "My-First-Maintenance-Window" ^
    --schedule "rate(5 minutes)" ^
    --duration 2 ^
    --cutoff 1 ^
    --allow-unassociated-targets ^
    --tags "Key"="Purpose","Value"="Tutorial"
```

O sistema retorna informações semelhantes às seguintes.

```
{
   "WindowId":"mw-0c50858d01EXAMPLE"
}
```

2. Agora, execute o comando a seguir para ver detalhes sobre esta e quaisquer outras janelas de manutenção que já estejam na sua conta.

```
aws ssm describe-maintenance-windows
```

O sistema retorna informações semelhantes às seguintes.

```
{
   "WindowIdentities":[
      {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "Enabled": true,
            "Duration": 2,
            "Cutoff": 1,
            "NextExecutionTime": "2019-05-11T16:46:16.991Z"
      }
   ]
}
```

Avance para Etapa 2: Registrar um nó de destino na janela de manutenção usando a AWS CLI.

Etapa 2: Registrar um nó de destino na janela de manutenção usando a AWS CLI

Nesta etapa, você registra um destino com sua nova janela de manutenção. Nesse caso, você especifica qual nó deverá ser atualizado quando a janela de manutenção for executada.

Para obter um exemplo de como registrar mais de um nó por vez usando IDs de nós, exemplos de como usar tags para identificar vários nós e exemplos de como especificar grupos de recursos como destinos, consulte Exemplos: Registrar destinos em uma janela de manutenção.

> **Note**
>
> Você já deve ter criado uma instância do Amazon Elastic Compute Cloud (Amazon EC2) para usar nesta etapa, conforme descrito no tutorial de pré-requisitos do Maintenance Windows.

Para registrar um nó de destino em uma janela de manutenção usando a AWS CLI

1. Execute o seguinte comando na máquina local. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

Linux & macOS

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --resource-type "INSTANCE" \
    --target "Key=InstanceIds,Values=i-02573cafcfEXAMPLE"
```

Windows

```
aws ssm register-target-with-maintenance-window ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --resource-type "INSTANCE" ^
    --target "Key=InstanceIds,Values=i-02573cafcfEXAMPLE"
```

O sistema retorna informações semelhantes às seguintes.

```
{
   "WindowTargetId":"e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
}
```

2. Agora execute o seguinte comando na máquina local para visualizar detalhes sobre o destino da janela de manutenção.

   Linux & macOS

   ```
   aws ssm describe-maintenance-window-targets \
       --window-id "mw-0c50858d01EXAMPLE"
   ```

   Windows

   ```
   aws ssm describe-maintenance-window-targets ^
       --window-id "mw-0c50858d01EXAMPLE"
   ```

   O sistema retorna informações semelhantes às seguintes.

   ```
   {
       "Targets": [
           {
   ```

```
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowTargetId": "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE",
            "ResourceType": "INSTANCE",
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-02573cafcfEXAMPLE"
                    ]
                }
            ]
        }
    ]
}
```

Avance para Etapa 3: Registrar uma tarefa na janela de manutenção usando a AWS CLI.

Exemplos: Registrar destinos em uma janela de manutenção

Você pode registrar um único nó como destino usando o ID do nó, conforme demonstrado em Etapa 2: Registrar um nó de destino na janela de manutenção usando a AWS CLI. Você também pode registrar um ou mais nós como destinos usando os formatos de comando nesta página.

Em geral, existem dois métodos para identificar os nós que você deseja usar como destinos da janela de manutenção: especificando nós individuais e usando tags de recurso. O método de tags de recurso fornece mais opções, conforme mostrado nos exemplos 2 e 3.

Você também pode especificar um ou mais grupos de recursos como o destino de uma janela de manutenção. Um grupo de recursos pode incluir nós e muitos outros tipos de recursos compatíveis da AWS. Os exemplos 4 e 5 seguintes demonstram como adicionar grupos de recursos aos destinos da janela de manutenção.

> **Note**
>
> Se uma única tarefa da janela de manutenção for registrada com vários destinos, suas chamadas de tarefa ocorrerão sequencialmente e não em paralelo. Se a tarefa precisar ser executada em vários destinos ao mesmo tempo, registre uma tarefa para cada destino individualmente e atribua a cada uma o mesmo nível de prioridade.

Para obter mais informações sobre como criar e gerenciar grupos de recursos, consulte O que são grupos de recursos? no Guia do usuário do AWS Resource Groups e Grupos de recursos e marcação para a AWS no Blog de notícias da AWS.

Para obter informações sobre cotas para o Maintenance Windows, uma ferramenta do AWS Systems Manager, além das especificadas nos exemplos a seguir, consulte Cotas de serviço do Systems Manager no Referência geral da Amazon Web Services.

Exemplo 1: Registrar vários destinos usando IDs do nó

Execute o seguinte comando no formato da máquina local para registrar vários nós como destinos usando os IDs de nós deles: Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

Linux & macOS

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --resource-type "INSTANCE" \
    --target
 "Key=InstanceIds,Values=i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE,i-07782c72faEXAMPLE"
```

Windows

```
aws ssm register-target-with-maintenance-window ^
    --window-id "mw-0c50858d01EXAMPLE ^
    --resource-type "INSTANCE" ^
    --target
 "Key=InstanceIds,Values=i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE,i-07782c72faEXAMPLE
```

Uso recomendado: muito útil para registrar um grupo exclusivo de nós em qualquer janela de manutenção pela primeira vez e quando eles não compartilham uma tag de nó comum.

Cotas: você pode especificar um total de até 50 nós para cada destino de janela de manutenção.

Exemplo 2: Registrar destinos usando tags de recursos aplicadas aos nós

Execute o seguinte comando na máquina local para registrar nós que já estejam marcados com um par de chave/valor atribuído por você: Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

Linux & macOS

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --resource-type "INSTANCE" \
    --target "Key=tag:Region,Values=East"
```

Windows

```
aws ssm register-target-with-maintenance-window ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --resource-type "INSTANCE" ^
    --target "Key=tag:Region,Values=East"
```

Uso recomendado: muito útil para registrar um grupo exclusivo de nós em qualquer janela de manutenção pela primeira vez e quando eles compartilham uma tag de nó comum.

Cotas: você pode especificar até cinco pares de chave-valor para cada destino. Se você especificar mais de um par de chave-valor, um nó deverá ser marcado com todas as chaves e valores de tag especificados para inclusão no grupo de destino.

> **Note**
>
> Você pode marcar um grupo de nós com a chave de tag `Patch Group` ou `PatchGroup` e atribuir aos nós um valor de chave comum, como `my-patch-group`. (Você deve usar `PatchGroup`, sem espaço, se tiver tags permitidas nos metadados da instância do EC2.) O Patch Manager, uma ferramenta do Systems Manager, avalia a chave `Patch Group` ou `PatchGroup` nos nós para ajudar a determinar qual a lista de referência de patches se aplica a eles. Se a sua tarefa executar o documento SSM `AWS-RunPatchBaseline` (ou o documento SSM legado `AWS-ApplyPatchBaseline`), você poderá especificar o mesmo par de chave/valor `Patch Group` ou `PatchGroup` ao registrar destinos com uma janela de manutenção. Por exemplo: `--target "Key=tag:PatchGroup,Values=my-patch-group`. Isso permite que você use uma janela de manutenção para atualizar patches em um grupo de nós que já esteja associado à mesma lista de referência de patches. Para ter mais informações, consulte Grupos de patches.

Exemplo 3: Registrar destinos usando um grupo de chaves de tag (sem valores de tag)

Execute o seguinte comando na máquina local para registrar nós que tenham uma ou mais chaves de tags atribuídas, independentemente de seus valores de chave. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

Linux & macOS

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --resource-type "INSTANCE" \
    --target "Key=tag-key,Values=Name,Instance-Type,CostCenter"
```

Windows

```
aws ssm register-target-with-maintenance-window ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --resource-type "INSTANCE" ^
    --target "Key=tag-key,Values=Name,Instance-Type,CostCenter"
```

Uso recomendado: útil quando você quiser marcar nós como destino especificando várias chaves de tag (sem seus valores) em vez de apenas uma tag-chave ou um par chave-valor de tag.

Cotas: você pode especificar até um total de cinco chaves de tag para cada destino. Se você especificar mais de uma chave de tag, um nó deverá ser marcado com todas as chaves de tags especificadas para inclusão no grupo de destino.

Exemplo 4: Registrar destinos usando o nome de grupo de recursos

Execute o seguinte comando na máquina local para registrar um grupo de recursos especificado, independentemente do tipo de recurso que ele contém. Substitua *mw-0c50858d01EXAMPLE* pelas suas próprias informações. Se as tarefas que você atribui à janela de manutenção não atuarem em um tipo de recurso incluído nesse grupo de recursos, o sistema poderá relatar um erro. As tarefas para as quais um tipo de recurso compatível é encontrado continuam a ser executadas apesar desses erros.

Linux & macOS

```
aws ssm register-target-with-maintenance-window \
```

```
    --window-id "mw-0c50858d01EXAMPLE" \
    --resource-type "RESOURCE_GROUP" \
    --target "Key=resource-groups:Name,Values=MyResourceGroup"
```

Windows

```
aws ssm register-target-with-maintenance-window ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --resource-type "RESOURCE_GROUP" ^
    --target "Key=resource-groups:Name,Values=MyResourceGroup"
```

Uso recomendado: útil quando você deseja especificar rapidamente um grupo de recursos como destino sem avaliar se todos os tipos de recurso serão direcionados por uma janela de manutenção, ou quando você sabe que o grupo de recursos contém apenas os tipos de recurso nos quais as tarefas executam ações.

Cotas: você pode especificar apenas um grupo de recursos como destino.

Exemplo 5: Registrar destinos filtrando tipos de recurso em um grupo de recursos

Execute o seguinte comando na máquina local para registrar somente certos tipos de recursos que pertencem a um grupo de recursos que você especificar. Substitua *mw-0c50858d01EXAMPLE* pelas suas próprias informações. Com essa opção, mesmo que você adicione uma tarefa para um tipo de recurso que pertence ao grupo de recursos, a tarefa não será executada se você ainda não tiver adicionado explicitamente o tipo de recurso ao filtro.

Linux & macOS

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --resource-type "RESOURCE_GROUP" \
    --target "Key=resource-groups:Name,Values=MyResourceGroup" \
    "Key=resource-
groups:ResourceTypeFilters,Values=AWS::EC2::Instance,AWS::ECS::Cluster"
```

Windows

```
aws ssm register-target-with-maintenance-window ^
    --window-id "mw-0c50858d01EXAMPLE" ^
```

```
    --resource-type "RESOURCE_GROUP" ^
    --target "Key=resource-groups:Name,Values=MyResourceGroup" ^
    "Key=resource-
groups:ResourceTypeFilters,Values=AWS::EC2::Instance,AWS::ECS::Cluster"
```

Uso recomendado: útil quando você quiser manter um controle rígido sobre os tipos de recurso da AWS nos quais sua janela de manutenção pode executar ações, ou quando o grupo de recursos contiver um grande número de tipos de recurso e você quiser evitar relatórios de erro desnecessários nos logs da janela de manutenção.

Cotas: você pode especificar apenas um grupo de recursos como destino.

Etapa 3: Registrar uma tarefa na janela de manutenção usando a AWS CLI

Nesta etapa do tutorial, você registra uma tarefa do AWS Systems Manager Run Command que executa o `df` na instância do Amazon Elastic Compute Cloud (Amazon EC2) para Linux. Os resultados desse comando padrão do Linux mostram a quantidade de espaço livre e a quantidade usada no sistema de arquivos do disco de sua instância.

- ou -

Se você estiver direcionando uma instância do Amazon EC2 para Windows Server em vez de para Linux, substitua o df no comando a seguir por ipconfig. A saída desse comando lista detalhes sobre o endereço IP, a máscara de sub-rede e o gateway padrão para adaptadores na instância de destino.

Quando você estiver pronto para registrar outros tipos de tarefa ou usar mais opções do Run Command disponíveis no Systems Manager, consulte Exemplos: Registrar tarefas em uma janela de manutenção. Lá fornecemos mais informações sobre os quatro tipos de tarefa e algumas das suas opções mais importante, para ajudar você a se planejar para cenários reais mais abrangentes.

Como registrar uma tarefa em uma janela de manutenção

1. Execute o seguinte comando na máquina local. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações. A versão que será executada a partir de uma máquina Windows local inclui os caracteres de escape ("/") necessários para executar o comando em sua ferramenta de linha de comando.

   Linux & macOS

   ```
   aws ssm register-task-with-maintenance-window \
   ```

```
    --window-id mw-0c50858d01EXAMPLE \
    --task-arn "AWS-RunShellScript" \
    --max-concurrency 1 --max-errors 1 \
    --priority 10 \
    --targets "Key=InstanceIds,Values=i-0471e04240EXAMPLE" \
    --task-type "RUN_COMMAND" \
    --task-invocation-parameters '{"RunCommand":{"Parameters":{"commands":
["df"]}}}'
```

Windows

```
aws ssm register-task-with-maintenance-window ^
    --window-id mw-0c50858d01EXAMPLE ^
    --task-arn "AWS-RunShellScript" ^
    --max-concurrency 1 --max-errors 1 ^
    --priority 10 ^
    --targets "Key=InstanceIds,Values=i-02573cafcfEXAMPLE" ^
    --task-type "RUN_COMMAND" ^
    --task-invocation-parameters={\"RunCommand\":{\"Parameters\":{\"commands\":
[\"df\"]}}}
```

O sistema retorna informações semelhantes às seguintes:

```
{
    "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE"
}
```

2. Agora, execute o seguinte comando para visualizar detalhes sobre a tarefa de janela de manutenção criada:

   Linux & macOS

   ```
   aws ssm describe-maintenance-window-tasks \
       --window-id mw-0c50858d01EXAMPLE
   ```

   Windows

   ```
   aws ssm describe-maintenance-window-tasks ^
       --window-id mw-0c50858d01EXAMPLE
   ```

3. O sistema retorna informações semelhantes às seguintes.

```
{
    "Tasks": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
            "TaskArn": "AWS-RunShellScript",
            "Type": "RUN_COMMAND",
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-02573cafcfEXAMPLE"
                    ]
                }
            ],
            "TaskParameters": {},
            "Priority": 10,
            "ServiceRoleArn": "arn:aws:iam::123456789012:role/
MyMaintenanceWindowServiceRole",
            "MaxConcurrency": "1",
            "MaxErrors": "1"
        }
    ]
}
```

4. Aguarde o runtime da tarefa, com base na programação especificada em Etapa 1: Criar a janela de manutenção usando a AWS CLI. Por exemplo, se você tiver especificado **`--schedule "rate(5 minutes)"`**, aguarde cinco minutos. Depois, execute o seguinte comando para visualizar informações sobre todas as execuções que ocorreram para essa tarefa.

   Linux & macOS

   ```
   aws ssm describe-maintenance-window-executions \
       --window-id mw-0c50858d01EXAMPLE
   ```

   Windows

   ```
   aws ssm describe-maintenance-window-executions ^
       --window-id mw-0c50858d01EXAMPLE
   ```

O sistema retorna informações semelhantes às seguintes.

```
{
    "WindowExecutions": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowExecutionId": "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE",
            "Status": "SUCCESS",
            "StartTime": 1557593493.096,
            "EndTime": 1557593498.611
        }
    ]
}
```

Tip

Depois que a tarefa for executada com êxito, reduza a taxa de execução da janela de manutenção. Por exemplo, execute o comando a seguir para reduzir a frequência para uma vez por semana. Substitua *mw-0c50858d01EXAMPLE* pelas suas próprias informações.

Linux & macOS

```
aws ssm update-maintenance-window \
    --window-id mw-0c50858d01EXAMPLE \
    --schedule "rate(7 days)"
```

Windows

```
aws ssm update-maintenance-window ^
    --window-id mw-0c50858d01EXAMPLE ^
    --schedule "rate(7 days)"
```

Para obter informações sobre como gerenciar programações da janela de manutenção, consulte Referência: Expressões cron e rate para o Systems Manager e Opções de programação da janela de manutenção e do período ativo.

> Para obter informações sobre como usar a AWS Command Line Interface (AWS CLI) para modificar uma janela de manutenção, consulte Tutorial: Atualizar uma janela de manutenção usando a AWS CLI.

Para praticar a execução de comandos da AWS CLI a fim de visualizar mais detalhes sobre sua tarefa de janela de manutenção e suas execuções, continue em Tutorial: Visualizar informações sobre tarefas e execuções de tarefas usando a AWS CLI.

Acesso à saída do comando do tutorial

Está além do escopo deste tutorial usar a AWS CLI para visualizar a saída do comando do Run Command associado às suas execuções da tarefa de janela de manutenção.

No entanto, esses dados podem ser visualizados usando a AWS CLI. (Você também pode visualizar a saída no console do Systems Manager ou em um arquivo de log armazenado em um bucket do Amazon Simple Storage Service (Amazon S3), caso tenha configurado a janela de manutenção para armazenar a saída do comando nele.) Observe que a saída do comando df em uma instância do EC2 para Linux é semelhante à seguinte:

```
Filesystem 1K-blocks Used Available Use% Mounted on

devtmpfs 485716 0 485716 0% /dev

tmpfs 503624 0 503624 0% /dev/shm

tmpfs 503624 328 503296 1% /run

tmpfs 503624 0 503624 0% /sys/fs/cgroup

/dev/xvda1 8376300 1464160 6912140 18% /
```

A saída do comando ipconfig em uma instância do EC2 para Windows Server é semelhante à seguinte:

```
Windows IP Configuration


Ethernet adapter Ethernet 2:
```

```
   Connection-specific DNS Suffix  . : example.com
   IPv4 Address. . . . . . . . . . . : 10.24.34.0/23
   Subnet Mask . . . . . . . . . . . : 255.255.255.255
   Default Gateway . . . . . . . . . : 0.0.0.0

Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : abc1.wa.example.net

Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::100b:c234:66d6:d24f%4
   IPv4 Address. . . . . . . . . . . : 192.0.2.0
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.0.2.0

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
```

Exemplos: Registrar tarefas em uma janela de manutenção

Você pode registrar uma tarefa no Run Command, uma ferramenta do AWS Systems Manager, com uma janela de manutenção usando o AWS Command Line Interface (AWS CLI), conforme demonstrado em Registre tarefas com a janela de manutenção. Você também pode registrar tarefas para fluxos de trabalho do Systems Manager Automation, funções do AWS Lambda, e tarefas do AWS Step Functions, conforme demonstrado abaixo, neste tópico.

> **Note**
>
> Especifique um ou mais destinos para as tarefas da janela de manutenção do tipo Run Command. Dependendo da tarefa, os destinos serão opcionais para outros tipos de tarefas da janela de manutenção (Automation, AWS Lambda e AWS Step Functions). Para obter

mais informações sobre como executar tarefas que não especificam destinos, consulte Registrar tarefas da janela de manutenção sem destinos.

Neste tópico, fornecemos exemplos de como usar o comando AWS Command Line Interface (AWS CLI) da `register-task-with-maintenance-window` para registrar cada um dos quatro tipos de tarefa compatíveis em uma janela de manutenção. Os exemplos são apenas para demonstração, mas você pode modificá-los para criar comandos de registro de tarefa funcionais.

Uso da opção --cli-input-json

Para gerenciar melhor suas opções de tarefas, use a opção de comando `--cli-input-json`, com valores de opção referenciados em um arquivo JSON.

Para usar o conteúdo do arquivo JSON de exemplo fornecido nos exemplos a seguir, faça o seguinte em sua máquina local:

1. Crie um arquivo com um nome, como `MyRunCommandTask.json`, `MyAutomationTask.json` ou outro nome de sua preferência.
2. Copie o conteúdo do nosso exemplo de JSON no arquivo.
3. Modifique o conteúdo do arquivo para o registro de sua tarefa e salve o arquivo.
4. No mesmo diretório em que armazenou o arquivo, execute o seguinte comando. Substitua o nome do arquivo por *`MyFile.json`*.

   Linux & macOS

   ```
   aws ssm register-task-with-maintenance-window \
       --cli-input-json file://MyFile.json
   ```

   Windows

   ```
   aws ssm register-task-with-maintenance-window ^
       --cli-input-json file://MyFile.json
   ```

Pseudoparâmetros em tarefas da janela de manutenção

Em alguns exemplos, usamos pseudoparâmetros como método para enviar informações de ID às tarefas. Por exemplo, `{{TARGET_ID}}` e `{{RESOURCE_ID}}` podem ser usados para passar os

IDs dos recursos da AWS somente para tarefas do Automation, do Lambda e do Step Functions. Para obter mais informações sobre pseudoparâmetros em conteúdo `--task-invocation-parameters`, consulte Usar pseudoparâmetros ao registrar tarefas da janela de manutenção.

Mais informações

- Opções de parâmetros para o comando register-task-with-maintenance-windows.
- register-task-with-maintenance-window na AWS CLI Command Reference
- RegisterTaskWithMaintenanceWindow na Referência de API do AWS Systems Manager

Exemplos de registro de tarefas

As seções a seguir fornecem um exemplo de comando da AWS CLI para registrar um tipo de tarefa compatível e um exemplo de JSON que pode ser usado com a opção `--cli-input-json`.

Registre uma tarefa Run Command do Systems Manager

Os exemplos a seguir demonstram como registrar tarefas Run Command do Systems Manager em uma janela de manutenção usando a AWS CLI:

Linux & macOS

```
aws ssm register-task-with-maintenance-window \
    --window-id mw-0c50858d01EXAMPLE \
    --task-arn "AWS-RunShellScript" \
    --max-concurrency 1 --max-errors 1 --priority 10 \
    --targets "Key=InstanceIds,Values=i-02573cafcfEXAMPLE" \
    --task-type "RUN_COMMAND" \
    --task-invocation-parameters '{"RunCommand":{"Parameters":{"commands":["df"]}}}'
```

Windows

```
aws ssm register-task-with-maintenance-window ^
    --window-id mw-0c50858d01EXAMPLE ^
    --task-arn "AWS-RunShellScript" ^
    --max-concurrency 1 --max-errors 1 --priority 10 ^
    --targets "Key=InstanceIds,Values=i-02573cafcfEXAMPLE" ^
    --task-type "RUN_COMMAND" ^
    --task-invocation-parameters "{\"RunCommand\":{\"Parameters\":{\"commands\":
[\"df\"]}}}"
```

Conteúdo JSON para uso com a opção de arquivo **`--cli-input-json`**:

```
{
    "TaskType": "RUN_COMMAND",
    "WindowId": "mw-0c50858d01EXAMPLE",
    "Description": "My Run Command task to update SSM Agent on an instance",
    "MaxConcurrency": "1",
    "MaxErrors": "1",
    "Name": "My-Run-Command-Task",
    "Priority": 10,
    "Targets": [
        {
            "Key": "WindowTargetIds",
            "Values": [
                "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
            ]
        }
    ],
    "TaskArn": "AWS-UpdateSSMAgent",
    "TaskInvocationParameters": {
        "RunCommand": {
            "Comment": "A TaskInvocationParameters test comment",
            "NotificationConfig": {
                "NotificationArn": "arn:aws:sns:region:123456789012:my-sns-topic-name",
                "NotificationEvents": [
                    "All"
                ],
                "NotificationType": "Invocation"
            },
            "OutputS3BucketName": "amzn-s3-demo-bucket",
            "OutputS3KeyPrefix": "S3-PREFIX",
            "TimeoutSeconds": 3600
        }
    }
}
```

Registre uma tarefa do Systems Manager Automation

Os exemplos a seguir demonstram como registrar tarefas do Systems Manager Automation em uma janela de manutenção usando a AWS CLI:

Comando da AWS CLI:

Linux & macOS

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --task-arn "AWS-RestartEC2Instance" \
    --service-role-arn arn:aws:iam::123456789012:role/MyMaintenanceWindowServiceRole
 \
    --task-type AUTOMATION \
    --task-invocation-parameters
 "Automation={DocumentVersion=5,Parameters={InstanceId='{{RESOURCE_ID}}'}}" \
    --priority 0 --name "My-Restart-EC2-Instances-Automation-Task" \
    --description "Automation task to restart EC2 instances"
```

Windows

```
aws ssm register-task-with-maintenance-window ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --task-arn "AWS-RestartEC2Instance" ^
    --service-role-arn arn:aws:iam::123456789012:role/MyMaintenanceWindowServiceRole
 ^
    --task-type AUTOMATION ^
    --task-invocation-parameters
 "Automation={DocumentVersion=5,Parameters={InstanceId='{{TARGET_ID}}'}}" ^
    --priority 0 --name "My-Restart-EC2-Instances-Automation-Task" ^
    --description "Automation task to restart EC2 instances"
```

Conteúdo JSON para uso com a opção de arquivo **`--cli-input-json`**:

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
        "TaskArn": "AWS-PatchInstanceWithRollback",
    "TaskType": "AUTOMATION","TaskInvocationParameters": {
        "Automation": {
            "DocumentVersion": "1",
            "Parameters": {
                "instanceId": [
                    "{{RESOURCE_ID}}"
                ]
            }
        }
    }
```

```
}
```

Registrar uma tarefa do AWS Lambda

Os exemplos a seguir demonstram como registrar tarefas de função do Lambda em uma janela de manutenção usando a AWS CLI:

Para esses exemplos, o usuário que criou a função do Lambda a nomeou como `SSMrestart-my-instances` e criou dois parâmetros chamados `instanceId` e `targetType`.

> **Important**
>
> A política do Maintenance Windows para IAM requer a adição do prefixo `SSM` aos nomes das funções Lambda (ou alias) . Antes de prosseguir com o registro desse tipo de tarefa, atualize o nome no AWS Lambda para incluir `SSM`. Por exemplo, se o nome da função Lambda for `MyLambdaFunction`, altere-o para `SSMMyLambdaFunction`.

Comando da AWS CLI:

Linux & macOS

> **Important**
>
> Se você estiver usando a versão 2 do AWS CLI, inclua a opção `--cli-binary-format raw-in-base64-out` no comando a seguir se sua carga útil do Lambda não for codificada em base64. A opção `cli_binary_format` está disponível apenas na versão 2. Para obter informações sobre essa e outras configurações do arquivo AWS CLI `config`, consulte Configurações de arquivo `config` compatíveis no Manual do usuário do AWS Command Line Interface.

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE" \
    --priority 2 --max-concurrency 10 --max-errors 5 --name "My-Lambda-Example" \
    --description "A description for my LAMBDA example task" --task-type "LAMBDA" \
    --task-arn "arn:aws:lambda:region:123456789012:function:serverlessrepo-
SSMrestart-my-instances-C4JF9EXAMPLE" \
```

```
    --task-invocation-parameters '{"Lambda":{"Payload":"{\"InstanceId\":
\"{{RESOURCE_ID}}\",\"targetType\":\"{{TARGET_TYPE}}\"}","Qualifier": "$LATEST"}}'
```

PowerShell

> **Important**
>
> Se você estiver usando a versão 2 do AWS CLI, inclua a opção `--cli-binary-format raw-in-base64-out` no comando a seguir se sua carga útil do Lambda não for codificada em base64. A opção `cli_binary_format` está disponível apenas na versão 2. Para obter informações sobre essa e outras configurações do arquivo AWS CLI `config`, consulte Configurações de arquivo `config` compatíveis no Manual do usuário do AWS Command Line Interface.

```
aws ssm register-task-with-maintenance-window `
    --window-id "mw-0c50858d01EXAMPLE" `
    --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE" `
    --priority 2 --max-concurrency 10 --max-errors 5 --name "My-Lambda-Example" `
    --description "A description for my LAMBDA example task" --task-type "LAMBDA" `
    --task-arn "arn:aws:lambda:region:123456789012:function:serverlessrepo-
SSMrestart-my-instances-C4JF9EXAMPLE" `
    --task-invocation-parameters '{\"Lambda\":{\"Payload\":\"{\\\"InstanceId\\\":\\
\"{{RESOURCE_ID}}\\\",\\\"targetType\\\":\\\"{{TARGET_TYPE}}\\\"}\",\"Qualifier\":
 \"$LATEST\"}}'
```

Conteúdo JSON para uso com a opção de arquivo **`--cli-input-json`**:

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
    "Targets": [
        {
            "Key": "WindowTargetIds",
            "Values": [
                "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
            ]
        }
    ],
    "TaskArn": "SSM_RestartMyInstances",
    "TaskType": "LAMBDA",
```

```
    "MaxConcurrency": "10",
    "MaxErrors": "10",
    "TaskInvocationParameters": {
        "Lambda": {
            "ClientContext": "ew0KICAi--truncated--0KIEXAMPLE",
            "Payload": "{ \"instanceId\": \"{{RESOURCE_ID}}\", \"targetType\":
 \"{{TARGET_TYPE}}\" }",
            "Qualifier": "$LATEST"
        }
    },
    "Name": "My-Lambda-Task",
    "Description": "A description for my LAMBDA task",
    "Priority": 5
}
```

Registre uma tarefa do Step Functions

Os exemplos a seguir demonstram como registrar tarefas da máquina de estado do Step Functions em uma janela de manutenção usando a AWS CLI:

> **Note**
>
> As tarefas da janela de manutenção oferecem suporte somente aos fluxos de trabalho da máquina de estado Step Functions Standard. Elas não oferecem suporte a fluxos de trabalho de máquinas de estado Express. Para obter informações sobre os tipos de fluxo de trabalho da máquina de estado, consulte Fluxos de trabalho Standard vs. Express no Guia do desenvolvedor do AWS Step Functions.

Para esses exemplos, o usuário que criou a máquina de estado do Step Functions criou uma máquina de estado chamada `SSMMyStateMachine` com um parâmetro chamado `instanceId`.

> **Important**
>
> A política do AWS Identity and Access Management (IAM) para a Maintenance Windows requer o uso do prefixo Step Functions nos nomes das máquinas de estado do `SSM`. Antes de prosseguir com o registro desse tipo de tarefa, é necessário atualizar o nome no AWS Step Functions a fim de incluir `SSM`. Por exemplo, se o nome da máquina de estado for `MyStateMachine`, altere para `SSMMyStateMachine`.

Comando da AWS CLI:

Linux & macOS

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE" \
    --task-arn arn:aws:states:region:123456789012:stateMachine:SSMMyStateMachine-
MggiqEXAMPLE \
    --task-type STEP_FUNCTIONS \
    --task-invocation-parameters '{"StepFunctions":{"Input":"{\"InstanceId\":
\"{{RESOURCE_ID}}\"}", "Name":"{{INVOCATION_ID}}"}}' \
    --priority 0 --max-concurrency 10 --max-errors 5 \
    --name "My-Step-Functions-Task" --description "A description for my Step
 Functions task"
```

PowerShell

```
aws ssm register-task-with-maintenance-window `
    --window-id "mw-0c50858d01EXAMPLE" `
    --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE" `
    --task-arn arn:aws:states:region:123456789012:stateMachine:SSMMyStateMachine-
MggiqEXAMPLE `
    --task-type STEP_FUNCTIONS `
    --task-invocation-parameters '{\"StepFunctions\":{\"Input\":\"{\\\"InstanceId\\
\":\\\"{{RESOURCE_ID}}\\\"}\", \"Name\":\"{{INVOCATION_ID}}\"}}' `
    --priority 0 --max-concurrency 10 --max-errors 5 `
    --name "My-Step-Functions-Task" --description "A description for my Step
 Functions task"
```

Conteúdo JSON para uso com a opção de arquivo **`--cli-input-json`**:

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
    "Targets": [
        {
            "Key": "WindowTargetIds",
            "Values": [
                "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
            ]
        }
    ],
```

```
    "TaskArn": "SSM_MyStateMachine",
    "TaskType": "STEP_FUNCTIONS",
    "MaxConcurrency": "10",
    "MaxErrors": "10",
    "TaskInvocationParameters": {
        "StepFunctions": {
            "Input": "{ \"instanceId\": \"{{TARGET_ID}}\" }",
            "Name": "{{INVOCATION_ID}}"
        }
    },
    "Name": "My-Step-Functions-Task",
    "Description": "A description for my Step Functions task",
    "Priority": 5
}
```

Opções de parâmetros para o comando register-task-with-maintenance-windows

O comando register-task-with-maintenance-window fornece várias opções para configurar uma tarefa de acordo com as suas necessidades. Algumas são necessárias, algumas são opcionais, outras se aplicam somente a um único tipo de tarefa de janela de manutenção.

Este tópico fornece informações sobre algumas dessas opções para ajudar você a trabalhar com exemplos nesta seção do tutorial. Para obter mais informações sobre as opções de comando, consulte register-task-with-maintenance-window na Referência de comandos do AWS CLI.

Opção de comando: **`--task-arn`**

A opção `--task-arn` é usada para especificar o recurso no qual a tarefa é executada. O valor especificado depende do tipo de tarefa que você estiver registrando, conforme descrito na tabela a seguir.

Formatos de TaskArn para tarefas de janela de manutenção

Tipo de tarefa de janela de manutenção	Valor de TaskArn
`RUN_COMMAND` e **`AUTOMATION`**	O `TaskArn` é o nome do documento SSM ou o Amazon Resource Name (ARN). Por exemplo: `AWS-RunBatchShellScript` - ou -

Tipo de tarefa de janela de manutenção	Valor de TaskArn
	`arn:aws:ssm:` *`region`*`:11112222 3333:document/My-Document` .
`LAMBDA`	`TaskArn` é o nome ou o ARN da função. Por exemplo: `SSMMy-Lambda-Function` - ou - `arn:aws:lambda:` *`region`*`:11112222 3333:function:SSMMyLambdaFu nction` . **Important** A política do Maintenance Windows para IAM requer a adição do prefixo `SSM` aos nomes das funções Lambda (ou alias) . Antes de prosseguir com o registro desse tipo de tarefa, atualize o nome no AWS Lambda para incluir `SSM`. Por exemplo, se o nome da função Lambda for `MyLambdaF unction` , altere-o para `SSMMyLamb daFunction` .

Tipo de tarefa de janela de manutenção	Valor de TaskArn
STEP_FUNCTIONS	`TaskArn` é o ARN da máquina de estado. Por exemplo: `arn:aws:states:us-east-2:111122223333:stateMachine:SSMMyStateMachine` . **Important** A política do IAM para as janelas de manutenção requer o uso do prefixo Step Functions nos nomes das máquinas de estado com o `SSM`. Antes de registrar esse tipo de tarefa, é necessário atualizar o nome no AWS Step Functions a fim de incluir `SSM`. Por exemplo, se o nome da máquina de estado for `MyStateMachine` , altere para `SSMMyStateMachine` .

Opção de comando: **`--service-role-arn`**

A função a ser assumida pelo AWS Systems Manager ao executar a tarefa de janela de manutenção.

Para obter mais informações, consulte Configurar o Maintenance Windows.

Opção de comando: **`--task-invocation-parameters`**

A opção `--task-invocation-parameters` é usada para especificar os parâmetros que são exclusivos para cada um dos quatro tipos de tarefa. Os parâmetros compatíveis com cada um dos quatro tipos de tarefa estão descritos na tabela a seguir.

> **Note**
>
> Para obter informações sobre como usar pseudoparâmetros em conteúdo `--task-invocation-parameters`, como {{TARGET_ID}}, consulte Usar pseudoparâmetros ao registrar tarefas da janela de manutenção.

Opções de parâmetros de invocação de tarefas para tarefas de janela de manutenção

Tipo de tarefa de janela de manutenção	Parâmetros disponíveis	Exemplo

RUN_COMMAND	`Comment`
	`DocumentHash`
	`DocumentHashType`
	`NotificationConfig`
	`OutputS3BucketName`
	`OutPutS3KeyPrefix`
	`Parameters`
	`ServiceRoleArn`
	`TimeoutSeconds`
		`"TaskInvocationParameters": { "RunCommand": { "Comment": "My Run Command task comment", "DocumentHash": "6554ed3d--truncated--5EXAMPLE", "DocumentHashType": "Sha256", "NotificationConfig": { "NotificationArn": "arn:aws:sns: region:123456789012:my-sns-topic-name", "NotificationEvents": ["FAILURE"], "NotificationType": "Invocation" },`

Tipo de tarefa de janela de manutenção	Parâmetros disponíveis	Exemplo
		"OutputS3BucketName": "amzn-s3-demo-bucket", "OutputS3KeyPrefix": " *S3-PREFIX* ", "Parameters": { "commands": ["Get-ChildItem$env:temp-Recurse|Remove-Item-Recurse-force"] }, "ServiceRoleArn": "arn:aws:iam::123456789012:role/MyMaintenanceWindowServiceRole", "TimeoutSeconds": 3600 } }

Tipo de tarefa de janela de manutenção	Parâmetros disponíveis	Exemplo

AUTOMATION	DocumentVersion
	
	Parameters
		"TaskInvocationParameters": {
		"Automation": {
		"DocumentVersion": "3",
		"Parameters": {
		"instanceid": [
		"{{TARGET_ID}}"
		]
		}
		}
		}

LAMBDA	ClientContext
	
	Payload
	
	Qualifier
		"TaskInvocationParameters": {
		"Lambda": {
		"ClientContext": "ew0KICAi--truncated--0KIEXAMPLE",
		"Payload": "{ \"targetId\": \"{{TARGET_ID}}\", \"targetType\": \"{{TARGET_TYPE}}\" }",
		"Qualifier": "$LATEST"
		}
		}

Tipo de tarefa de janela de manutenção	Parâmetros disponíveis	Exemplo

STEP_FUNCTIONS	`Input`
	
	`Name`
		`"TaskInvocationParameters": { "StepFunctions": { "Input": "{ \"targetId\": \"{{TARGET_ID}}\" }", "Name": "{{INVOCATION_ID}}" } }`

Tutorial: Visualizar informações sobre janelas de manutençãousando a AWS CLI

Esta seção inclui comandos para ajudar você a atualizar ou obter informações sobre suas janelas de manutenção, tarefas, execuções e invocações. Os exemplos são organizados por comando para demonstrar como usar as opções de comando para filtrar para o tipo de detalhes que você deseja ver.

Ao seguir as etapas neste tutorial, substitua os valores em texto itálico *vermelho* por suas próprias opções e IDs. Por exemplo, substitua o ID da janela de manutenção *mw-0c50858d01EXAMPLE* e o ID da instância*i-02573cafcfEXAMPLE* pelos IDs de recursos criados.

Para obter informações sobre como definir e configurar o AWS Command Line Interface (AWS CLI), consulte Instalar, atualizar e desinstalar a AWS CLI e Configurar a AWS CLI.

Exemplos de comando

- Exemplos para "describe-maintenance-windows"
- Exemplos para "describe-maintenance-window-targets"
- Exemplos para "describe-maintenance-window-tasks"
- Exemplos para "describe-maintenance-windows-for-target"
- Exemplos para "describe-maintenance-window-executions"
- Exemplos para "describe-maintenance-window-schedule"

Exemplos para "describe-maintenance-windows"

Liste todas as janelas de manutenção em sua conta da Conta da AWS

Execute o seguinte comando:

```
aws ssm describe-maintenance-windows
```

O sistema retorna informações semelhantes às seguintes.

```
{
   "WindowIdentities":[
      {
         "WindowId":"mw-0c50858d01EXAMPLE",
         "Name":"My-First-Maintenance-Window",
         "Enabled":true,
         "Duration":2,
         "Cutoff":0,
         "NextExecutionTime": "2019-05-18T17:01:01.137Z"
      },
      {
         "WindowId":"mw-9a8b7c6d5eEXAMPLE",
         "Name":"My-Second-Maintenance-Window",
         "Enabled":true,
         "Duration":4,
         "Cutoff":1,
         "NextExecutionTime": "2019-05-30T03:30:00.137Z"
      },
   ]
}
```

Listar todas as janelas de manutenção habilitadas

Execute o seguinte comando:

```
aws ssm describe-maintenance-windows --filters "Key=Enabled,Values=true"
```

O sistema retorna informações semelhantes às seguintes.

```
{
   "WindowIdentities":[
```

```
      {
         "WindowId":"mw-0c50858d01EXAMPLE",
         "Name":"My-First-Maintenance-Window",
         "Enabled":true,
         "Duration":2,
         "Cutoff":0,
         "NextExecutionTime": "2019-05-18T17:01:01.137Z"
      },
      {
         "WindowId":"mw-9a8b7c6d5eEXAMPLE",
         "Name":"My-Second-Maintenance-Window",
         "Enabled":true,
         "Duration":4,
         "Cutoff":1,
         "NextExecutionTime": "2019-05-30T03:30:00.137Z"
      },
   ]
}
```

Listar todas as janelas de manutenção desabilitadas

Execute o seguinte comando:

```
aws ssm describe-maintenance-windows --filters "Key=Enabled,Values=false"
```

O sistema retorna informações semelhantes às seguintes.

```
{
    "WindowIdentities": [
        {
            "WindowId": "mw-6e5c9d4b7cEXAMPLE",
            "Name": "My-Disabled-Maintenance-Window",
            "Enabled": false,
            "Duration": 2,
            "Cutoff": 1
        }
    ]
}
```

Listar todas as janelas de manutenção com nomes que começam com um determinado prefixo

Execute o seguinte comando:

```
aws ssm describe-maintenance-windows --filters "Key=Name,Values=My"
```

O sistema retorna informações semelhantes às seguintes.

```
{
    "WindowIdentities": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "Enabled": true,
            "Duration": 2,
            "Cutoff": 0,
            "NextExecutionTime": "2019-05-18T17:01:01.137Z"
        },
        {
            "WindowId": "mw-9a8b7c6d5eEXAMPLE",
            "Name": "My-Second-Maintenance-Window",
            "Enabled": true,
            "Duration": 4,
            "Cutoff": 1,
            "NextExecutionTime": "2019-05-30T03:30:00.137Z"
        },
        {
            "WindowId": "mw-6e5c9d4b7cEXAMPLE",
            "Name": "My-Disabled-Maintenance-Window",
            "Enabled": false,
            "Duration": 2,
            "Cutoff": 1
        }
    ]
}
```

Exemplos para "describe-maintenance-window-targets"

Exibir os destinos para uma janela de manutenção correspondentes ao valor das informações de um proprietário específico

Execute o seguinte comando:

Linux & macOS

```
aws ssm describe-maintenance-window-targets \
```

```
    --window-id "mw-6e5c9d4b7cEXAMPLE" \
    --filters "Key=OwnerInformation,Values=CostCenter1"
```

Windows

```
aws ssm describe-maintenance-window-targets ^
    --window-id "mw-6e5c9d4b7cEXAMPLE" ^
    --filters "Key=OwnerInformation,Values=CostCenter1"
```

> **Note**
>
> As chaves de filtro compatíveis são `Type`, `WindowTargetId` e `OwnerInformation`.

O sistema retorna informações semelhantes às seguintes.

```
{
    "Targets": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowTargetId": "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE",
            "ResourceType": "INSTANCE",
            "Targets": [
                {
                    "Key": "tag:Name",
                    "Values": [
                        "Production"
                    ]
                }
            ],
            "OwnerInformation": "CostCenter1",
            "Name": "Target1"
        }
    ]
}
```

Exemplos para "describe-maintenance-window-tasks"

Mostre todas as tarefas registradas que invocam o documento de comando d**`AWS-RunPowerShellScript`** do SSM

Execute o seguinte comando:

Linux & macOS

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-0c50858d01EXAMPLE" \
    --filters "Key=TaskArn,Values=AWS-RunPowerShellScript"
```

Windows

```
aws ssm describe-maintenance-window-tasks ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --filters "Key=TaskArn,Values=AWS-RunPowerShellScript"
```

O sistema retorna informações semelhantes às seguintes.

```
{
   "Tasks":[
      {
         "ServiceRoleArn": "arn:aws:iam::111122223333:role/
MyMaintenanceWindowServiceRole",
         "MaxErrors":"1",
         "TaskArn":"AWS-RunPowerShellScript",
         "MaxConcurrency":"1",
         "WindowTaskId":"4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
         "TaskParameters":{
            "commands":{
               "Values":[
                  "driverquery.exe"
               ]
            }
         },
         "Priority":3,
         "Type":"RUN_COMMAND",
         "Targets":[
            {
               "TaskTargetId":"i-02573cafcfEXAMPLE",
               "TaskTargetType":"INSTANCE"
            }
         ]
      },
      {
```

```
         "ServiceRoleArn":"arn:aws:iam::111122223333:role/
MyMaintenanceWindowServiceRole",
         "MaxErrors":"1",
         "TaskArn":"AWS-RunPowerShellScript",
         "MaxConcurrency":"1",
         "WindowTaskId":"4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
         "TaskParameters":{
            "commands":{
               "Values":[
                  "ipconfig"
               ]
            }
         },
         "Priority":1,
         "Type":"RUN_COMMAND",
         "Targets":[
            {
               "TaskTargetId":"i-02573cafcfEXAMPLE",
               "TaskTargetType":"WINDOW_TARGET"
            }
         ]
      }
   ]
}
```

Mostrar todas as tarefas registradas que têm uma prioridade de "3"

Execute o seguinte comando:

Linux & macOS

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-9a8b7c6d5eEXAMPLE" \
    --filters "Key=Priority,Values=3"
```

Windows

```
aws ssm describe-maintenance-window-tasks ^
    --window-id "mw-9a8b7c6d5eEXAMPLE" ^
    --filters "Key=Priority,Values=3"
```

O sistema retorna informações semelhantes às seguintes.

```
{
   "Tasks":[
      {
         "ServiceRoleArn":"arn:aws:iam::111122223333:role/
MyMaintenanceWindowServiceRole",
         "MaxErrors":"1",
         "TaskArn":"AWS-RunPowerShellScript",
         "MaxConcurrency":"1",
         "WindowTaskId":"4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
         "TaskParameters":{
            "commands":{
               "Values":[
                  "driverquery.exe"
               ]
            }
         },
         "Priority":3,
         "Type":"RUN_COMMAND",
         "Targets":[
            {
               "TaskTargetId":"i-02573cafcfEXAMPLE",
               "TaskTargetType":"INSTANCE"
            }
         ]
      }
   ]
}
```

Mostrar todas as tarefas registradas que têm uma prioridade de "1" e usam Run Command

Execute o seguinte comando:

Linux & macOS

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-0c50858d01EXAMPLE" \
    --filters "Key=Priority,Values=1" "Key=TaskType,Values=RUN_COMMAND"
```

Windows

```
aws ssm describe-maintenance-window-tasks ^
    --window-id "mw-0c50858d01EXAMPLE" ^
```

```
    --filters "Key=Priority,Values=1" "Key=TaskType,Values=RUN_COMMAND"
```

O sistema retorna informações semelhantes às seguintes.

```
{
    "Tasks": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
            "TaskArn": "AWS-RunShellScript",
            "Type": "RUN_COMMAND",
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-02573cafcfEXAMPLE"
                    ]
                }
            ],
            "TaskParameters": {},
            "Priority": 1,
            "ServiceRoleArn": "arn:aws:iam::111122223333:role/
MyMaintenanceWindowServiceRole",
            "MaxConcurrency": "1",
            "MaxErrors": "1"
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowTaskId": "8a5c4629-31b0-4edd-8aea-33698EXAMPLE",
            "TaskArn": "AWS-UpdateSSMAgent",
            "Type": "RUN_COMMAND",
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-0471e04240EXAMPLE"
                    ]
                }
            ],
            "TaskParameters": {},
            "Priority": 1,
```

```
            "ServiceRoleArn": "arn:aws:iam::111122223333:role/
MyMaintenanceWindowServiceRole",
            "MaxConcurrency": "1",
            "MaxErrors": "1",
            "Name": "My-Run-Command-Task",
            "Description": "My Run Command task to update SSM Agent on an instance"
        }
    ]
}
```

Exemplos para "describe-maintenance-windows-for-target"

Listar informações sobre os destinos da janela de manutenção ou as tarefas associadas a um nó específico.

Execute o seguinte comando:

Linux & macOS

```
aws ssm describe-maintenance-windows-for-target \
    --resource-type INSTANCE \
    --targets "Key=InstanceIds,Values=i-02573cafcfEXAMPLE" \
    --max-results 10
```

Windows

```
aws ssm describe-maintenance-windows-for-target ^
    --resource-type INSTANCE ^
    --targets "Key=InstanceIds,Values=i-02573cafcfEXAMPLE" ^
    --max-results 10
```

O sistema retorna informações semelhantes às seguintes.

```
{
    "WindowIdentities": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "My-First-Maintenance-Window"
        },
        {
            "WindowId": "mw-9a8b7c6d5eEXAMPLE",
            "Name": "My-Second-Maintenance-Window"
```

```
        }
    ]
}
```

Exemplos para "describe-maintenance-window-executions"

Listar todas as tarefas executadas antes de uma determinada data

Execute o seguinte comando:

Linux & macOS

```
aws ssm describe-maintenance-window-executions \
    --window-id "mw-9a8b7c6d5eEXAMPLE" \
    --filters "Key=ExecutedBefore,Values=2019-05-12T05:00:00Z"
```

Windows

```
aws ssm describe-maintenance-window-executions ^
    --window-id "mw-9a8b7c6d5eEXAMPLE" ^
    --filters "Key=ExecutedBefore,Values=2019-05-12T05:00:00Z"
```

O sistema retorna informações semelhantes às seguintes.

```
{
    "WindowExecutions": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowExecutionId": "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE",
            "Status": "FAILED",
            "StatusDetails": "The following SSM parameters are invalid: LevelUp",
            "StartTime": 1557617747.993,
            "EndTime": 1557617748.101
        },
        {
            "WindowId": "mw-9a8b7c6d5eEXAMPLE",
            "WindowExecutionId": "791b72e0-f0da-4021-8b35-f95dfEXAMPLE",
            "Status": "SUCCESS",
            "StartTime": 1557594085.428,
            "EndTime": 1557594090.978
        },
        {
```

```
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowExecutionId": "ecec60fa-6bb0-4d26-98c7-140308EXAMPLE",
            "Status": "SUCCESS",
            "StartTime": 1557593793.483,
            "EndTime": 1557593798.978
        }
    ]
}
```

Listar todas as tarefas executadas depois de uma determinada data

Execute o seguinte comando:

Linux & macOS

```
aws ssm describe-maintenance-window-executions \
    --window-id "mw-9a8b7c6d5eEXAMPLE" \
    --filters "Key=ExecutedAfter,Values=2018-12-31T17:00:00Z"
```

Windows

```
aws ssm describe-maintenance-window-executions ^
    --window-id "mw-9a8b7c6d5eEXAMPLE" ^
    --filters "Key=ExecutedAfter,Values=2018-12-31T17:00:00Z"
```

O sistema retorna informações semelhantes às seguintes.

```
{
    "WindowExecutions": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowExecutionId": "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE",
            "Status": "FAILED",
            "StatusDetails": "The following SSM parameters are invalid: LevelUp",
            "StartTime": 1557617747.993,
            "EndTime": 1557617748.101
        },
        {
            "WindowId": "mw-9a8b7c6d5eEXAMPLE",
            "WindowExecutionId": "791b72e0-f0da-4021-8b35-f95dfEXAMPLE",
            "Status": "SUCCESS",
            "StartTime": 1557594085.428,
```

```
            "EndTime": 1557594090.978
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowExecutionId": "ecec60fa-6bb0-4d26-98c7-140308EXAMPLE",
            "Status": "SUCCESS",
            "StartTime": 1557593793.483,
            "EndTime": 1557593798.978
        }
    ]
}
```

Exemplos para "describe-maintenance-window-schedule"

Exiba as próximas dez execuções da janela de manutenção programadas para um nó específico

Execute o seguinte comando:

Linux & macOS

```
aws ssm describe-maintenance-window-schedule \
    --resource-type INSTANCE \
    --targets "Key=InstanceIds,Values=i-07782c72faEXAMPLE" \
    --max-results 10
```

Windows

```
aws ssm describe-maintenance-window-schedule ^
    --resource-type INSTANCE ^
    --targets "Key=InstanceIds,Values=i-07782c72faEXAMPLE" ^
    --max-results 10
```

O sistema retorna informações semelhantes às seguintes.

```
{
    "ScheduledWindowExecutions": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "ExecutionTime": "2019-05-18T23:35:24.902Z"
        },
        {
```

```
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "ExecutionTime": "2019-05-25T23:35:24.902Z"
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "ExecutionTime": "2019-06-01T23:35:24.902Z"
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "ExecutionTime": "2019-06-08T23:35:24.902Z"
        },
        {
            "WindowId": "mw-9a8b7c6d5eEXAMPLE",
            "Name": "My-Second-Maintenance-Window",
            "ExecutionTime": "2019-06-15T23:35:24.902Z"
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "ExecutionTime": "2019-06-22T23:35:24.902Z"
        },
        {
            "WindowId": "mw-9a8b7c6d5eEXAMPLE",
            "Name": "My-Second-Maintenance-Window",
            "ExecutionTime": "2019-06-29T23:35:24.902Z"
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "ExecutionTime": "2019-07-06T23:35:24.902Z"
        },
        {
            "WindowId": "mw-9a8b7c6d5eEXAMPLE",
            "Name": "My-Second-Maintenance-Window",
            "ExecutionTime": "2019-07-13T23:35:24.902Z"
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "ExecutionTime": "2019-07-20T23:35:24.902Z"
        }
```

```
    ],
    "NextToken": "AAEABUXdceT92FvtKld/dGHELj5Mi+GKW/EXAMPLE"
}
```

Exiba a programação da janela de manutenção para nós marcados com um determinado par de chave/valor

Execute o seguinte comando:

Linux & macOS

```
aws ssm describe-maintenance-window-schedule \
    --resource-type INSTANCE \
    --targets "Key=tag:prod,Values=rhel7"
```

Windows

```
aws ssm describe-maintenance-window-schedule ^
    --resource-type INSTANCE ^
    --targets "Key=tag:prod,Values=rhel7"
```

O sistema retorna informações semelhantes às seguintes.

```
{
    "ScheduledWindowExecutions": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "DemoRateStartDate",
            "ExecutionTime": "2019-10-20T05:34:56-07:00"
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "DemoRateStartDate",
            "ExecutionTime": "2019-10-21T05:34:56-07:00"
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "DemoRateStartDate",
            "ExecutionTime": "2019-10-22T05:34:56-07:00"
        },
        {
```

```
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "DemoRateStartDate",
            "ExecutionTime": "2019-10-23T05:34:56-07:00"
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "DemoRateStartDate",
            "ExecutionTime": "2019-10-24T05:34:56-07:00"
        }
    ],
    "NextToken": "AAEABccwSXqQRGKiTZ1yzGELR6cxW4W/EXAMPLE"
}
```

Exibir os horários de início da próximas quatro execuções de uma janela de manutenção

Execute o seguinte comando:

Linux & macOS

```
aws ssm describe-maintenance-window-schedule \
    --window-id "mw-0c50858d01EXAMPLE" \
    --max-results "4"
```

Windows

```
aws ssm describe-maintenance-window-schedule ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --max-results "4"
```

O sistema retorna informações semelhantes às seguintes.

```
{
    "WindowSchedule": [
        {
            "ScheduledWindowExecutions": [
                {
                    "ExecutionTime": "2019-10-04T10:10:10Z",
                    "Name": "My-First-Maintenance-Window",
                    "WindowId": "mw-0c50858d01EXAMPLE"
                },
                {
```

```
                    "ExecutionTime": "2019-10-11T10:10:10Z",
                    "Name": "My-First-Maintenance-Window",
                    "WindowId": "mw-0c50858d01EXAMPLE"
                },
                {
                    "ExecutionTime": "2019-10-18T10:10:10Z",
                    "Name": "My-First-Maintenance-Window",
                    "WindowId": "mw-0c50858d01EXAMPLE"
                },
                {
                    "ExecutionTime": "2019-10-25T10:10:10Z",
                    "Name": "My-First-Maintenance-Window",
                    "WindowId": "mw-0c50858d01EXAMPLE"
                }
            ]
        }
    ]
}
```

Tutorial: Visualizar informações sobre tarefas e execuções de tarefas usando a AWS CLI

Este tutorial demonstra como usar a AWS Command Line Interface (AWS CLI) para visualizar detalhes sobre as tarefas concluídas da janela de manutenção.

Se você estiver continuando diretamente do Tutorial: Criar e configurar uma janela de manutenção usando a AWS CLI, reserve tempo suficiente para que a sua janela de manutenção seja executada pelo menos uma vez, para ver os resultados dessa execução.

Ao seguir as etapas neste tutorial, substitua os valores em texto itálico *vermelho* por suas próprias opções e IDs. Por exemplo, substitua o ID da janela de manutenção *mw-0c50858d01EXAMPLE* e o ID da instância*i-02573cafcfEXAMPLE* pelos IDs de recursos criados.

Para visualiar informações sobre tarefas e execuções de tarefas usando a AWS CLI

1. Execute o comando a seguir para visualizar uma lista de execuções de tarefas para uma janela de manutenção específica.

   Linux & macOS

   ```
   aws ssm describe-maintenance-window-executions \
       --window-id "mw-0c50858d01EXAMPLE"
   ```

Windows

```
aws ssm describe-maintenance-window-executions ^
    --window-id "mw-0c50858d01EXAMPLE"
```

O sistema retorna informações semelhantes às seguintes.

```
{
    "WindowExecutions": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowExecutionId": "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE",
            "Status": "SUCCESS",
            "StartTime": 1557593793.483,
            "EndTime": 1557593798.978
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowExecutionId": "791b72e0-f0da-4021-8b35-f95dfEXAMPLE",
            "Status": "SUCCESS",
            "StartTime": 1557593493.096,
            "EndTime": 1557593498.611
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowExecutionId": "ecec60fa-6bb0-4d26-98c7-140308EXAMPLE",
            "Status": "SUCCESS",
            "StatusDetails": "No tasks to execute.",
            "StartTime": 1557593193.309,
            "EndTime": 1557593193.334
        }
    ]
}
```

2. Execute o comando a seguir para obter informações sobre uma execução de tarefa da janela de manutenção.

   Linux & macOS

   ```
   aws ssm get-maintenance-window-execution \
   ```

```
    --window-execution-id "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE"
```

Windows

```
aws ssm get-maintenance-window-execution ^
    --window-execution-id "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE"
```

O sistema retorna informações semelhantes às seguintes.

```
{
    "WindowExecutionId": "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE",
    "TaskIds": [
        "c9b05aba-197f-4d8d-be34-e73fbEXAMPLE"
    ],
    "Status": "SUCCESS",
    "StartTime": 1557593493.096,
    "EndTime": 1557593498.611
}
```

3. Execute o comando a seguir para listar as tarefas executadas como parte de uma execução da janela de manutenção.

   Linux & macOS

   ```
   aws ssm describe-maintenance-window-execution-tasks \
       --window-execution-id "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE"
   ```

   Windows

   ```
   aws ssm describe-maintenance-window-execution-tasks ^
       --window-execution-id "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE"
   ```

   O sistema retorna informações semelhantes às seguintes.

   ```
   {
       "WindowExecutionTaskIdentities": [
           {
               "WindowExecutionId": "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE",
               "TaskExecutionId": "c9b05aba-197f-4d8d-be34-e73fbEXAMPLE",
   ```

```
            "Status": "SUCCESS",
            "StartTime": 1557593493.162,
            "EndTime": 1557593498.57,
            "TaskArn": "AWS-RunShellScript",
            "TaskType": "RUN_COMMAND"
        }
    ]
}
```

4. Execute o seguinte comando para obter os detalhes de uma execução de tarefa.

   Linux & macOS

   ```
   aws ssm get-maintenance-window-execution-task \
       --window-execution-id "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE" \
       --task-id "c9b05aba-197f-4d8d-be34-e73fbEXAMPLE"
   ```

   Windows

   ```
   aws ssm get-maintenance-window-execution-task ^
       --window-execution-id "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE" ^
       --task-id "c9b05aba-197f-4d8d-be34-e73fbEXAMPLE"
   ```

   O sistema retorna informações semelhantes às seguintes.

   ```
   {
       "WindowExecutionId": "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE",
       "TaskExecutionId": "c9b05aba-197f-4d8d-be34-e73fbEXAMPLE",
       "TaskArn": "AWS-RunShellScript",
       "ServiceRole": "arn:aws:iam::111122223333:role/MyMaintenanceWindowServiceRole",
       "Type": "RUN_COMMAND",
       "TaskParameters": [
           {
               "aws:InstanceId": {
                   "Values": [
                       "i-02573cafcfEXAMPLE"
                   ]
               },
               "commands": {
                   "Values": [
                       "df"
   ```

```
                ]
            }
        }
    ],
    "Priority": 10,
    "MaxConcurrency": "1",
    "MaxErrors": "1",
    "Status": "SUCCESS",
    "StartTime": 1557593493.162,
    "EndTime": 1557593498.57
}
```

5. Execute o seguinte comando para obter as invocações de tarefas específicas realizadas para uma execução de tarefa.

   Linux & macOS

   ```
   aws ssm describe-maintenance-window-execution-task-invocations \
       --window-execution-id "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE" \
       --task-id "c9b05aba-197f-4d8d-be34-e73fbEXAMPLE"
   ```

   Windows

   ```
   aws ssm describe-maintenance-window-execution-task-invocations ^
       --window-execution-id "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE" ^
       --task-id "c9b05aba-197f-4d8d-be34-e73fbEXAMPLE"
   ```

   O sistema retorna informações semelhantes às seguintes.

```
{
    "WindowExecutionTaskInvocationIdentities": [
        {
            "WindowExecutionId": "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE",
            "TaskExecutionId": "c9b05aba-197f-4d8d-be34-e73fbEXAMPLE",
            "InvocationId": "c336d2ab-09de-44ba-8f6a-6136cEXAMPLE",
            "ExecutionId": "76a5a04f-caf6-490c-b448-92c02EXAMPLE",
            "TaskType": "RUN_COMMAND",
            "Parameters": "{\"documentName\":\"AWS-RunShellScript\",\"instanceIds
\":[\"i-02573cafcfEXAMPLE\"],\"maxConcurrency\":\"1\",\"maxErrors\":\"1\",
\"parameters\":{\"commands\":[\"df\"]}}",
            "Status": "SUCCESS",
```

```
            "StatusDetails": "Success",
            "StartTime": 1557593493.222,
            "EndTime": 1557593498.466
        }
    ]
}
```

Tutorial: Atualizar uma janela de manutenção usando a AWS CLI

Este tutorial demonstra como usar a AWS Command Line Interface (AWS CLI) para atualizar uma janela de manutenção. Ele também mostra como atualizar diferentes tipos de tarefas, incluindo aquelas do AWS Systems Manager Run Command e Automation, do AWS Lambda e do AWS Step Functions.

Os exemplos nesta seção usam as seguintes ações do Systems Manager para atualizar uma janela de manutenção.

- UpdateMaintenanceWindow
- UpdateMaintenanceWindowTarget
- UpdateMaintenanceWindowTask
- DeregisterTargetFromMaintenanceWindow

Para obter informações sobre como usar o console do Systems Manager para atualizar uma janela de manutenção, consulte Atualize ou exclua recursos da janela de manutenção usando o console.

Ao seguir as etapas neste tutorial, substitua os valores em texto itálico *vermelho* por suas próprias opções e IDs. Por exemplo, substitua o ID da janela de manutenção *mw-0c50858d01EXAMPLE* e o ID da instância*i-02573cafcfEXAMPLE* pelos IDs de recursos criados.

Para atualizar uma janela de manutenção usando a AWS CLI

1. Abra o AWS CLI e execute o seguinte comando para atualizar um destino de forma a incluir um nome e uma descrição.

   Linux & macOS

   ```
   aws ssm update-maintenance-window-target \
       --window-id "mw-0c50858d01EXAMPLE" \
       --window-target-id "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE" \
   ```

```
    --name "My-Maintenance-Window-Target" \
    --description "Description for my maintenance window target"
```

Windows

```
aws ssm update-maintenance-window-target ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --window-target-id "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE" ^
    --name "My-Maintenance-Window-Target" ^
    --description "Description for my maintenance window target"
```

O sistema retorna informações semelhantes às seguintes.

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
    "WindowTargetId": "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE",
    "Targets": [
        {
            "Key": "InstanceIds",
            "Values": [
                "i-02573cafcfEXAMPLE"
            ]
        }
    ],
    "Name": "My-Maintenance-Window-Target",
    "Description": "Description for my maintenance window target"
}
```

2. Execute o seguinte comando para usar a opção `replace` a fim de remover o campo de descrição e adicionar outro destino. O campo de descrição é removido, pois a atualização não inclui o campo (um valor nulo). Especifique um nó adicional que tenha sido configurado para uso com o Systems Manager:

   Linux & macOS

   ```
   aws ssm update-maintenance-window-target \
       --window-id "mw-0c50858d01EXAMPLE" \
       --window-target-id "d208dedf-3f6b-41ff-ace8-8e751EXAMPLE" \
       --targets "Key=InstanceIds,Values=i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE" \
       --name "My-Maintenance-Window-Target" \
   ```

```
    --replace
```

Windows

```
aws ssm update-maintenance-window-target ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --window-target-id "d208dedf-3f6b-41ff-ace8-8e751EXAMPLE" ^
    --targets "Key=InstanceIds,Values=i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE" ^
    --name "My-Maintenance-Window-Target" ^
    --replace
```

O sistema retorna informações semelhantes às seguintes.

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
    "WindowTargetId": "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE",
    "Targets": [
        {
            "Key": "InstanceIds",
            "Values": [
                "i-02573cafcfEXAMPLE",
                "i-0471e04240EXAMPLE"
            ]
        }
    ],
    "Name": "My-Maintenance-Window-Target"
}
```

3. A opção `start-date` permite atrasar a ativação de uma janela de manutenção até uma data futura determinada. A opção `end-date` permite que você defina uma data e hora no futuro após a qual a janela de manutenção não será mais executada. Especifique as opções no formato estendido ISO-8601.

   Execute o comando a seguir para especificar um intervalo de data e hora para execuções de janela de manutenção programadas regularmente:

   Linux & macOS

```
aws ssm update-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --start-date "2020-10-01T10:10:10Z" \
```

```
    --end-date "2020-11-01T10:10:10Z"
```

Windows

```
aws ssm update-maintenance-window ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --start-date "2020-10-01T10:10:10Z" ^
    --end-date "2020-11-01T10:10:10Z"
```

4. Execute o comando a seguir para atualizar uma tarefa do Run Command.

> **Tip**
>
> Se seu destino for uma instância do Amazon Elastic Compute Cloud (Amazon EC2) para o Windows Server, altere `df` para `ipconfig`, e `AWS-RunShellScript` para `AWS-RunPowerShellScript` no comando a seguir.

Linux & macOS

```
aws ssm update-maintenance-window-task \
    --window-id "mw-0c50858d01EXAMPLE" \
    --window-task-id "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE" \
    --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
 \
    --task-arn "AWS-RunShellScript" \
    --service-role-arn "arn:aws:iam::account-id:role/MaintenanceWindowsRole" \
    --task-invocation-parameters "RunCommand={Comment=Revising my Run Command
 task,Parameters={commands=df}}" \
    --priority 1 --max-concurrency 10 --max-errors 4 \
    --name "My-Task-Name" --description "A description for my Run Command task"
```

Windows

```
aws ssm update-maintenance-window-task ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --window-task-id "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE" ^
    --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
 ^
    --task-arn "AWS-RunShellScript" ^
    --service-role-arn "arn:aws:iam::account-id:role/MaintenanceWindowsRole" ^
```

```
    --task-invocation-parameters "RunCommand={Comment=Revising my Run Command
task,Parameters={commands=df}}" ^
    --priority 1 --max-concurrency 10 --max-errors 4 ^
    --name "My-Task-Name" --description "A description for my Run Command task"
```

O sistema retorna informações semelhantes às seguintes.

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
    "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
    "Targets": [
        {
            "Key": "WindowTargetIds",
            "Values": [
                "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
            ]
        }
    ],
    "TaskArn": "AWS-RunShellScript",
    "ServiceRoleArn": "arn:aws:iam::111122223333:role/MaintenanceWindowsRole",
    "TaskParameters": {},
    "TaskInvocationParameters": {
        "RunCommand": {
            "Comment": "Revising my Run Command task",
            "Parameters": {
                "commands": [
                    "df"
                ]
            }
        }
    },
    "Priority": 1,
    "MaxConcurrency": "10",
    "MaxErrors": "4",
    "Name": "My-Task-Name",
    "Description": "A description for my Run Command task"
}
```

5. Adapte e execute o comando a seguir para atualizar uma tarefa do Lambda.

Linux & macOS

```
aws ssm update-maintenance-window-task \
    --window-id mw-0c50858d01EXAMPLE \
    --window-task-id 4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE \
    --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
 \
    --task-arn "arn:aws:lambda:region:111122223333:function:SSMTestLambda" \
    --service-role-arn "arn:aws:iam:account-id:role/MaintenanceWindowsRole" \
    --task-invocation-parameters '{"Lambda":{"Payload":"{\"InstanceId\":
\"{{RESOURCE_ID}}\",\"targetType\":\"{{TARGET_TYPE}}\"}"}}' \
    --priority 1 --max-concurrency 10 --max-errors 5 \
    --name "New-Lambda-Task-Name" \
    --description "A description for my Lambda task"
```

Windows

```
aws ssm update-maintenance-window-task ^
    --window-id mw-0c50858d01EXAMPLE ^
    --window-task-id 4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE ^
    --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
 ^
    --task-arn --task-arn
 "arn:aws:lambda:region:111122223333:function:SSMTestLambda" ^
    --service-role-arn "arn:aws:iam:account-id:role/MaintenanceWindowsRole" ^
    --task-invocation-parameters '{"Lambda":{"Payload":"{\"InstanceId\":
\"{{RESOURCE_ID}}\",\"targetType\":\"{{TARGET_TYPE}}\"}"}}' ^
    --priority 1 --max-concurrency 10 --max-errors 5 ^
    --name "New-Lambda-Task-Name" ^
    --description "A description for my Lambda task"
```

O sistema retorna informações semelhantes às seguintes.

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
    "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
    "Targets": [
        {
            "Key": "WindowTargetIds",
            "Values": "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
```

```
        }
    ],
    "TaskArn": "arn:aws:lambda:us-east-2:111122223333:function:SSMTestLambda",
    "ServiceRoleArn": "arn:aws:iam::111122223333:role/MaintenanceWindowsRole",
    "TaskParameters": {},
    "TaskInvocationParameters": {
        "Lambda": {
            "Payload": "e30="
        }
    },
    "Priority": 1,
    "MaxConcurrency": "10",
    "MaxErrors": "5",
    "Name": "New-Lambda-Task-Name",
    "Description": "A description for my Lambda task"
}
```

6. Se você estiver atualizando uma tarefa do Step Functions, adapte e execute o seguinte comando para atualizar os task-invocation-parameters:

   Linux & macOS

   ```
   aws ssm update-maintenance-window-task \
       --window-id "mw-0c50858d01EXAMPLE" \
       --window-task-id "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE" \
       --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
    \
       --task-arn "arn:aws:states:region:execution:SSMStepFunctionTest" \
       --service-role-arn "arn:aws:iam:account-id:role/MaintenanceWindowsRole" \
       --task-invocation-parameters '{"StepFunctions":{"Input":"{\"InstanceId\":
   \"{{RESOURCE_ID}}\"}"}}' \
       --priority 0 --max-concurrency 10 --max-errors 5 \
       --name "My-Step-Functions-Task" \
       --description "A description for my Step Functions task"
   ```

   Windows

   ```
   aws ssm update-maintenance-window-task ^
       --window-id "mw-0c50858d01EXAMPLE" ^
       --window-task-id "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE" ^
       --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
    ^
       --task-arn "arn:aws:states:region:execution:SSMStepFunctionTest" ^
   ```

```
    --service-role-arn "arn:aws:iam:account-id:role/MaintenanceWindowsRole" ^
    --task-invocation-parameters '{"StepFunctions":{"Input":"{\"InstanceId\":
\"{{RESOURCE_ID}}\"}"}}' ^
    --priority 0 --max-concurrency 10 --max-errors 5 ^
    --name "My-Step-Functions-Task" ^
    --description "A description for my Step Functions task"
```

O sistema retorna informações semelhantes às seguintes.

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
    "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
    "Targets": [
        {
            "Key": "WindowTargetIds",
            "Values": [
                "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
            ]
        }
    ],
    "TaskArn": "arn:aws:states:us-
east-2:111122223333:execution:SSMStepFunctionTest",
    "ServiceRoleArn": "arn:aws:iam::111122223333:role/MaintenanceWindowsRole",
    "TaskParameters": {},
    "TaskInvocationParameters": {
        "StepFunctions": {
            "Input": "{\"instanceId\":\"{{RESOURCE_ID}}\"}"
        }
    },
    "Priority": 0,
    "MaxConcurrency": "10",
    "MaxErrors": "5",
    "Name": "My-Step-Functions-Task",
    "Description": "A description for my Step Functions task"
}
```

7. Execute o comando a seguir para cancelar o registro de um destino de uma janela de manutenção. Este exemplo usa o parâmetro `safe` para determinar se o destino é referenciado por qualquer tarefa e, portanto, seguro para ter o registro cancelado.

Linux & macOS

```
aws ssm deregister-target-from-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --window-target-id "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE" \
    --safe
```

Windows

```
aws ssm deregister-target-from-maintenance-window ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --window-target-id "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE" ^
    --safe
```

O sistema retorna informações semelhantes às seguintes.

```
An error occurred (TargetInUseException) when calling the
 DeregisterTargetFromMaintenanceWindow operation:
This Target cannot be deregistered because it is still referenced in Task:
 4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE
```

8. Execute o comando a seguir para cancelar o registro de um destino de uma janela de manutenção, mesmo que o destino seja referenciado por uma tarefa. Você pode forçar a operação de cancelamento de registro usando o parâmetro `no-safe`.

   Linux & macOS

   ```
   aws ssm deregister-target-from-maintenance-window \
       --window-id "mw-0c50858d01EXAMPLE" \
       --window-target-id "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE" \
       --no-safe
   ```

   Windows

   ```
   aws ssm deregister-target-from-maintenance-window ^
       --window-id "mw-0c50858d01EXAMPLE" ^
       --window-target-id "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE" ^
       --no-safe
   ```

O sistema retorna informações semelhantes às seguintes.

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
    "WindowTargetId": "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
}
```

9. Execute o comando a seguir para atualizar uma tarefa do Run Command. Este exemplo usa um parâmetro Parameter Store do Systems Manager chamado `UpdateLevel`, que é formatado da seguinte maneira: '`{{ssm:UpdateLevel}}`'

   Linux & macOS

   ```
   aws ssm update-maintenance-window-task \
       --window-id "mw-0c50858d01EXAMPLE" \
       --window-task-id "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE" \
       --targets "Key=InstanceIds,Values=i-02573cafcfEXAMPLE"  \
       --task-invocation-parameters "RunCommand={Comment=A comment for my task
    update,Parameters={UpdateLevel='{{ssm:UpdateLevel}}'}}"
   ```

   Windows

   ```
   aws ssm update-maintenance-window-task ^
       --window-id "mw-0c50858d01EXAMPLE" ^
       --window-task-id "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE" ^
       --targets "Key=InstanceIds,Values=i-02573cafcfEXAMPLE"  ^
       --task-invocation-parameters "RunCommand={Comment=A comment for my task
    update,Parameters={UpdateLevel='{{ssm:UpdateLevel}}'}}"
   ```

   O sistema retorna informações semelhantes às seguintes.

   ```
   {
       "WindowId": "mw-0c50858d01EXAMPLE",
       "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
       "Targets": [
           {
               "Key": "InstanceIds",
               "Values": [
                   "i-02573cafcfEXAMPLE"
   ```

```
            ]
        }
    ],
    "TaskArn": "AWS-RunShellScript",
    "ServiceRoleArn": "arn:aws:iam::111122223333:role/
MyMaintenanceWindowServiceRole",
    "TaskParameters": {},
    "TaskInvocationParameters": {
        "RunCommand": {
            "Comment": "A comment for my task update",
            "Parameters": {
                "UpdateLevel": [
                    "{{ssm:UpdateLevel}}"
                ]
            }
        }
    },
    "Priority": 10,
    "MaxConcurrency": "1",
    "MaxErrors": "1"
}
```

10. Execute o seguinte comando para atualizar uma tarefa do Automation para especificar os parâmetros `WINDOW_ID` e `WINDOW_TASK_ID` para o parâmetro `task-invocation-parameters`:

    Linux & macOS

    ```
    aws ssm update-maintenance-window-task \
        --window-id "mw-0c50858d01EXAMPLE" \
        --window-task-id "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE" \
        --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE \
        --task-arn "AutoTestDoc" \
        --service-role-arn "arn:aws:iam:account-id:role/
    MyMaintenanceWindowServiceRole \
        --task-invocation-parameters
     "Automation={Parameters={InstanceId='{{RESOURCE_ID}}',initiator='{{WINDOW_ID}}.Task-
    {{WINDOW_TASK_ID}}'}}" \
        --priority 3 --max-concurrency 10 --max-errors 5
    ```

    Windows

    ```
    aws ssm update-maintenance-window-task ^
    ```

```
    --window-id "mw-0c50858d01EXAMPLE" ^
    --window-task-id "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE" ^
    --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE ^
    --task-arn "AutoTestDoc" ^
    --service-role-arn "arn:aws:iam:account-id:role/
MyMaintenanceWindowServiceRole ^
    --task-invocation-parameters
 "Automation={Parameters={InstanceId='{{RESOURCE_ID}}',initiator='{{WINDOW_ID}}.Task-
{{WINDOW_TASK_ID}}'}}" ^
    --priority 3 --max-concurrency 10 --max-errors 5
```

O sistema retorna informações semelhantes às seguintes.

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
    "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
    "Targets": [
        {
            "Key": "WindowTargetIds",
            "Values": [
                "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
            ]
        }
    ],
    "TaskArn": "AutoTestDoc",
    "ServiceRoleArn": "arn:aws:iam::111122223333:role/
MyMaintenanceWindowServiceRole",
    "TaskParameters": {},
    "TaskInvocationParameters": {
        "Automation": {
            "Parameters": {
                "multi": [
                    "{{WINDOW_TASK_ID}}"
                ],
                "single": [
                    "{{WINDOW_ID}}"
                ]
            }
        }
    },
    "Priority": 0,
    "MaxConcurrency": "10",
```

```
    "MaxErrors": "5",
    "Name": "My-Automation-Task",
    "Description": "A description for my Automation task"
}
```

Tutorial: Excluir uma janela de manutenção usando a AWS CLI

Para excluir uma janela de manutenção que você criou nesses tutoriais, execute o seguinte comando:

```
aws ssm delete-maintenance-window --window-id "mw-0c50858d01EXAMPLE"
```

O sistema retorna informações semelhantes às seguintes.

```
{
   "WindowId":"mw-0c50858d01EXAMPLE"
}
```

Tutorial: Create a maintenance window for patching using the console

> **Important**
>
> Você pode continuar a usar esse tópico legado para criar uma janela de manutenção para aplicar patch. No entanto, recomendamos usar uma política de patch em vez disso. Para obter mais informações, consulte Configurações de políticas de patches em Quick Setup e Configurar a aplicação de patches para instâncias em uma organização usando uma política de patch do Quick Setup.

Para minimizar o impacto na disponibilidade do seu servidor, recomendamos que você configure uma janela de manutenção para executar a aplicação de patch em horários que não interromperão suas operações de negócios.

É necessário configurar perfis e permissões para o Maintenance Windows, uma ferramenta do AWS Systems Manager, antes de começar este procedimento. Para obter mais informações, consulte Configurar o Maintenance Windows.

Para criar uma janela de manutenção para aplicação de patch

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Maintenance Windows.
3. Escolha Create maintenance window (Criar janela de manutenção).
4. No campo Name (Nome), insira um nome que designe isso como uma janela de manutenção para aplicar patch a atualizações críticas e importantes.
5. (Opcional) Em Description (Descrição), insira uma descrição.
6. Escolha Allow unregistered targets (Permitir destinos não registrados) se quiser permitir que uma tarefa da janela de manutenção seja executada em nós gerenciados, mesmo que você não tenha registrado esses nós como destinos.

   Se você escolher essa opção, poderá escolher os nós não registrados (por ID do nó) quando registrar uma tarefa na janela de manutenção.

   Se você não escolher essa opção, deverá escolher destinos anteriormente registrados quando registrar uma tarefa na janela de manutenção.
7. Na parte superior da seção Schedule (Programar) especifique uma programação para a janela de manutenção usando uma das três opções de agendamento.

   Para obter mais informações sobre criar expressões cron/rate, consulte Referência: Expressões cron e rate para o Systems Manager.
8. Em Duration (Duração), insira o número de horas que a janela de manutenção será executada. O valor especificado determina a hora de término específica para a janela de manutenção com base no horário em que ela começa. Nenhuma tarefa da janela de manutenção tem permissão para iniciar após a hora de término resultante menos o número de horas especificado para Stop initiating tasks (Parar de iniciar tarefas) na próxima etapa.

   Por exemplo, se a janela de manutenção começar às 15h, a duração for de três horas e o valor Stop initiating tasks (Parar de iniciar tarefas) for uma hora, nenhuma tarefa da janela de manutenção poderá ser iniciada depois das 17h.
9. Em Stop initiating tasks (Para de iniciar tarefas), insira o número de horas antes do final da janela de manutenção que o sistema deve parar de agendar novas tarefas para execução.
10. (Opcional) Em Window start date (Data de início da janela), especifique uma data e hora no formato ISO-8601 estendido para quando você deseja que a janela de manutenção se torne

ativa. Isso permite que você atrase a ativação da janela de manutenção até a data futura especificada.

11. (Opcional) Em Window end date (Data de término da janela), especifique uma data e hora no formato ISO-8601 estendido para quando você deseja que a janela de manutenção se torne inativa. Isso permite que você defina uma data e hora no futuro após a qual a janela de manutenção não será mais executada.
12. (Opcional) Em Fuso horário do agendamento, especifique o fuso horário no qual as execuções da janela de manutenção devem se basear, no formato da IANA (Internet Assigned Numbers Authority). Por exemplo: "America/Los_Angeles", "etc/UTC" ou "Ásia/Seul".

    Para obter mais informações sobre os formatos válidos, consulte o Banco de dados de fusos horários no site da IANA.
13. (Opcional) Na área Manage tags (Gerenciar tags), aplique um ou mais pares de nome/valor de chave de tag à janela de manutenção.

    Tags são metadados opcionais que você atribui a um recurso. As tags permitem categorizar um recurso de diferentes formas, como por finalidade, proprietário ou ambiente. Por exemplo, você poderá marcar essa janela de manutenção para identificar o tipo de tarefa a ser executada. Nesse caso, você pode especificar o seguinte par de nome/valor:

    - `Key=TaskType,Value=Patching`
14. Escolha Create maintenance window (Criar janela de manutenção).
15. Na lista da janela de manutenção, escolha a janela de manutenção que você acabou de criar e selecione Actions (Ações), Register targets (Registrar destinos).
16. (Opcional) Na seção Maintenance window target details, forneça um nome, uma descrição e informações sobre o proprietário (seu nome ou alias) para esse destino.
17. Em Seleção de destinos, escolha Especificação de tags de instância.
18. Em Especificar tags de instância, insira uma chave de tag e um valor de tag para identificar os nós a serem registrados na janela de manutenção, e escolha Adicionar.
19. Escolha Register target. O sistema cria um destino de janela de manutenção.
20. Na página de detalhes da janela de manutenção que você criou, selecione Actions (Ações), Register run command task (Registrar tarefa do Run Command).
21. (Opcional) Em Maintenance window task details (Detalhes da janela de manutenção), forneça um nome e uma descrição para essa tarefa.
22. Para Command document (Documento de comando), escolha `AWS-RunPatchBaseline`.

23. Em Task priority (Prioridade de tarefa), escolha uma prioridade. Zero (`0`) é a prioridade mais alta.
24. Em Targets (Destinos), em Target by (Destino por), escolha o destino da janela de manutenção que você criou anteriormente neste procedimento.
25. Para Rate control (Controle de taxa):

    - Em Concurrency (Concorrência), especifique um número ou uma porcentagem de nós gerenciados nos quais executar o comando ao mesmo tempo.

      > **Note**
      >
      > Se você selecionou destinos especificando tags aplicadas a instâncias a nós gerenciados ou especificando grupos de recursos da AWS, e não tiver certeza de quantas instâncias são direcionadas, restrinja o número de instâncias que poderão executar o documento ao mesmo tempo, especificando uma porcentagem.

    - Em Error threshold (Limite de erro), especifique quando parar de executar o comando em outros nós depois de falhar em alguns ou em uma porcentagem de nós. Por exemplo, se você especificar três erros, o Systems Manager deixará de enviar o comando quando o 4° erro for recebido. Os nós gerenciados que continuam processando o comando também podem enviar erros.

26. (Opcional) Em Perfil de serviço do IAM, escolha um perfil para fornecer permissões ao Systems Manager para assumir quando executar uma tarefa da janela de manutenção.

    Se você não especificar um ARN de perfil de serviço, o Systems Manager usará um perfil vinculado ao serviço em sua conta. Se nenhum perfil vinculado ao serviço apropriado para Systems Manager existir em sua conta, ele será criado quando a tarefa for registrada com êxito.

    > **Note**
    >
    > Para melhorar a postura de segurança, é altamente recomendável criar uma política personalizada e um perfil de serviço personalizado para executar as tarefas da janela de manutenção. A política pode ser criada para fornecer somente as permissões necessárias para as tarefas da sua janela de manutenção específica. Para obter mais informações, consulte Configurar o Maintenance Windows.

27. (Opcional) Em Opções de saída, para salvar a saída de comando em um arquivo, selecione a caixa Habilitar a gravação da saída no S3. Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

> **Note**
>
> As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil da instância atribuído ao nó gerenciado, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço IAM associada ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

Para fazer streaming da saída para um grupo de logs do Amazon CloudWatch Logs, selecione a caixa de resultado do CloudWatch. Insira o nome do grupo de logs na caixa.

28. Na seção SNS notifications (Notificações do SNS), se quiser enviar notificações sobre o status da execução do comando, marque a caixa de seleção Enable SNS notifications (Habilitar notificações do SNS).

    Para obter mais informações sobre a configuração de notificações do Amazon SNS para o Run Command, consulte Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS.

29. Em Parameters (Parameters):

    - Em Operation (Operação), escolha Scan (Verificar) para verificar se há patches ausentes ou escolha Install (Instalar) para verificar e instalar patches ausentes.
    - Não é necessário inserir nada no campo Snapshot Id (ID do snapshot). Esse sistema gera e fornece esse parâmetro automaticamente.
    - Não é necessário inserir nada no campo Install Override List (Instalar a lista de substituição), a menos que você queira que o Patch Manager use um conjunto de patches diferente do especificado na lista de referência do patch. Para mais informações, consulte Nome do parâmetro: `InstallOverrideList`.

- Na opção RebootOption, especifique se deseja que os nós gerenciados sejam reinicializados se os patches forem instalados durante a operação `Install` ou se o Patch Manager detectar outros patches que foram instalados desde a última reinicialização do nó. Para mais informações, consulte Nome do parâmetro: `RebootOption`.
- (Opcional) Em Comment (Comentário), insira uma nota de acompanhamento ou lembrete sobre esse comando.
- Em Timeout (segundos) (Tempo limite em segundos), insira o número de segundos que o sistema deve aguardar a conclusão da operação para que ela seja considerada malsucedida.

30. Selecione Register run command task (Registrar tarefa executar comando).

Após a conclusão da tarefa da janela de manutenção, você pode visualizar os detalhes de conformidade de patches no console do Systems Manager, na página da ferramenta Fleet Manager.

Também é possível visualizar as informações de conformidade na guia Relatórios de conformidade da ferramenta Patch Manager.

Você também pode usar as APIs DescribePatchGroupState e DescribeInstancePatchStatesForPatchGroup para visualizar detalhes de conformidade. Para obter informações sobre dados de conformidade dos patches, consulte Sobre a conformidade de patches.

Agendamentos de aplicação de patches usando janelas de manutenção

Depois de configurar uma lista de referência de patches (e opcionalmente um grupo de patches), você poderá aplicar patches ao nó usando uma janela de manutenção. Uma janela de manutenção pode reduzir o impacto na disponibilidade do servidor, permitindo especificar um tempo para executar o processo de aplicação de patch que não interrompa as operações de negócios. Uma janela de manutenção funciona assim:

1. Crie uma janela de manutenção com uma programação para suas operações de aplicação de patch.
2. Escolha os destinos para a janela de manutenção, especificando a tag `Patch Group` ou `PatchGroup` para o nome da tag e qualquer valor para o qual você tenha definido tags do Amazon Elastic Compute Cloud (Amazon EC2), p. ex., “web servers” ou “US-EAST-PROD”. (Você deve usar `PatchGroup`, sem espaço, se tiver permissão para tags nos metadados da instância do EC2.)
3. Crie uma tarefa de janela de manutenção e especifique o documento `AWS-RunPatchBaseline`.

Ao configurar a tarefa, você pode optar por verificar os nós ou verificar e instalar patches nesses nós. Se você optar por verificar os nós, o Patch Manager, uma ferramenta do AWS Systems Manager, verificará cada um e gerará uma lista de patches ausentes para você examinar.

Se você optar por verificar e instalar patches, o Patch Manager verificará cada nó e comparará a lista de patches instalados com a lista de patches aprovados na lista de referência. O Patch Manager identifica patches ausentes e, em seguida, baixa e instala todos os patches ausentes e aprovados.

Se você quiser executar uma verificação ou instalação única para corrigir um problema, use o Run Command para chamar o documento `AWS-RunPatchBaseline` diretamente.

> **Important**
>
> Depois de instalar os patches, o Systems Manager reinicia cada nó. A reinicialização é necessária para garantir que os patches sejam instalados corretamente e que o sistema não tenha deixado o nó em um estado potencialmente ruim. (Exceção: Se o parâmetro `RebootOption` estiver definido como `NoReboot` no documento `AWS-RunPatchBaseline`, o nó gerenciado não será reinicializado depois que o Patch Manager for executado. Para obter mais informações, consulte Nome do parâmetro: `RebootOption`).

Usar pseudoparâmetros ao registrar tarefas da janela de manutenção

Ao registrar uma tarefa no Maintenance Windows, uma ferramenta do AWS Systems Manager, é necessário especificar os parâmetros que são exclusivos para cada um dos quatro tipos de tarefa. (Nos comandos da CLI, eles são fornecidos usando a opção `--task-invocation-parameters`.)

Você também pode fazer referência a determinados valores usando sintaxe de pseudoparâmetro, como `{{RESOURCE_ID}}`, `{{TARGET_TYPE}}` e `{{WINDOW_TARGET_ID}}`. Quando a tarefa de janela de manutenção é executada, ela envia os valores corretos em vez dos espaços reservados do pseudoparâmetro. A lista completa de pseudoparâmetros que podem ser usados é mostrada mais adiante neste tópico em Pseudoparâmetros compatíveis.

> **Important**
>
> Para o tipo de destino `RESOURCE_GROUP`, dependendo do formato de ID necessário para a tarefa, é possível escolher entre usar `{{TARGET_ID}}` e `{{RESOURCE_ID}}` para fazer referência ao recurso quando a tarefa for executada. `{{TARGET_ID}}` retorna o ARN

completo do recurso. {{RESOURCE_ID}} retorna somente um nome mais curto ou o ID do recurso, conforme mostrado nestes exemplos.

- Formato {{TARGET_ID}}: arn:aws:ec2:us-east-1:123456789012:instance/i-02573cafcfEXAMPLE
- Formato {{RESOURCE_ID}}: i-02573cafcfEXAMPLE

Para o tipo de destino INSTANCE, os parâmetros {{TARGET_ID}} e {{RESOURCE_ID}} produzem somente o ID da instância. Para ter mais informações, consulte Pseudoparâmetros compatíveis.
{{TARGET_ID}} e {{RESOURCE_ID}} podem ser usados para passar os IDs dos recursos da AWS somente para o Automation, para o Lambda e para o Step Functions. Esses dois pseudoparâmetros não podem ser usados com as tarefas do Run Command.

Exemplos de pseudoparâmetros

Suponha que sua carga para uma tarefa do AWS Lambda precise fazer referência a uma instância por seu ID.

Se você estiver usando como destino uma janela de manutenção INSTANCE ou RESOURCE_GROUP, isso pode ser feito usando o pseudoparâmetro {{RESOURCE_ID}}. Por exemplo:

```
"TaskArn": "arn:aws:lambda:us-east-2:111122223333:function:SSMTestFunction",
    "TaskType": "LAMBDA",
    "TaskInvocationParameters": {
        "Lambda": {
            "ClientContext": "ew0KICAi--truncated--0KIEXAMPLE",
            "Payload": "{ \"instanceId\": \"{{RESOURCE_ID}}\" }",
            "Qualifier": "$LATEST"
        }
    }
```

Se a sua tarefa do Lambda tiver como objetivo ser executada em outro tipo de destino compatível, além de instâncias do Amazon Elastic Compute Cloud (Amazon EC2), como uma tabela do Amazon DynamoDB, a mesma sintaxe poderá ser usada e {{RESOURCE_ID}} produzirá somente o nome da tabela. No entanto, se você precisar do ARN completo da tabela, use {{TARGET_ID}}, conforme mostrado no exemplo a seguir.

```
"TaskArn": "arn:aws:lambda:us-east-2:111122223333:function:SSMTestFunction",
    "TaskType": "LAMBDA",
    "TaskInvocationParameters": {
        "Lambda": {
            "ClientContext": "ew0KICAi--truncated--0KIEXAMPLE",
            "Payload": "{ \"tableArn\": \"{{TARGET_ID}}\" }",
            "Qualifier": "$LATEST"
        }
    }
```

A mesma sintaxe funciona para instâncias de destino ou outros tipos de recursos. Quando vários tipos de recursos forem adicionados a um grupo de recursos, a tarefa será executada em cada um dos recursos apropriados.

> **Important**
>
> Nem todos os tipos de recursos que podem ser incluídos em um grupo de recursos geram um valor para o parâmetro `{{RESOURCE_ID}}`. Para obter uma lista de tipos de recursos com suporte, consulte Pseudoparâmetros compatíveis.

Como outro exemplo, para executar uma tarefa do Automation que interrompa suas instâncias do EC2, especifique o documento do Systems Manager `AWS-StopEC2Instance` (documento SSM) como o valor `TaskArn` e use o pseudoparâmetro `{{RESOURCE_ID}}`:

```
"TaskArn": "AWS-StopEC2Instance",
    "TaskType": "AUTOMATION"
    "TaskInvocationParameters": {
        "Automation": {
            "DocumentVersion": "1",
            "Parameters": {
                "instanceId": [
                    "{{RESOURCE_ID}}"
                ]
            }
        }
    }
```

Para executar uma tarefa do Automation que copia um snapshot de um volume do Amazon Elastic Block Store (Amazon EBS), especifique o documento do SSM `AWS-CopySnapshot` como o valor de `TaskArn` e use o pseudoparâmetro `{{RESOURCE_ID}}`.

```
"TaskArn": "AWS-CopySnapshot",
    "TaskType": "AUTOMATION"
    "TaskInvocationParameters": {
        "Automation": {
            "DocumentVersion": "1",
            "Parameters": {
                "SourceRegion": "us-east-2",
                "targetType":"RESOURCE_GROUP",
                "SnapshotId": [
                    "{{RESOURCE_ID}}"
                ]
            }
        }
    }
```

Pseudoparâmetros compatíveis

A lista a seguir descreve os pseudoparâmetros que podem ser especificados usando a sintaxe `{{`*PSEUDO_PARAMETER*`}}` na opção `--task-invocation-parameters`.

- **`WINDOW_ID`**: o ID da janela de manutenção do destino.
- **`WINDOW_TASK_ID`**: o ID da tarefa da janela que está em execução.
- **`WINDOW_TARGET_ID`**: o ID do destino de janela que inclui o destino (ID de destino).
- **`WINDOW_EXECUTION_ID`**: o ID da execução de janela atual.
- **`TASK_EXECUTION_ID`**: o ID da execução de tarefa atual.
- **`INVOCATION_ID`**: o ID da invocação atual.
- **`TARGET_TYPE`**: o tipo de destino. Os tipos com suporte incluem `RESOURCE_GROUP` e `INSTANCE`.
- **`TARGET_ID`**:

  Se o tipo de destino especificado for `INSTANCE`, o pseudoparâmetro `TARGET_ID` será substituído pelo ID da instância. Por exemplo, `i-078a280217EXAMPLE`.

  Se o tipo de destino especificado for `RESOURCE_GROUP`, o valor referenciado para a execução da tarefa será o ARN completo do recurso. Por exemplo: `arn:aws:ec2:us-`

east-1:123456789012:instance/i-078a280217EXAMPLE. A tabela a seguir fornece valores de exemplo de TARGET_ID para tipos de recursos específicos em um grupo de recursos.

> **Note**
>
> TARGET_ID não tem suporte para tarefas do Run Command.

Tipo de recurso	Exemplo de TARGET_ID
AWS::CloudWatch::Alarm	arn:aws:cloudwatch:us-east-1:123456789012:alarm:MyCloudWatchAlarm i-078a280217EXAMPLE
AWS::DynamoDB::Table	arn:aws:dynamodb:us-east-1:123456789012:table/MyTable
AWS::EC2::Instance	arn:aws:ec2:us-east-1:123456789012:instance/ i-078a280217EXAMPLE
AWS::EC2::Image	arn:aws:ec2:us-east-1:123456789012:image/ami-02250b3732EXAMPLE
AWS::EC2::SecurityGroup	arn:aws:ec2:us-east-1:123456789012:security-group/sg-cEXAMPLE

Tipo de recurso	Exemplo de TARGET_ID
AWS::EC2::Snapshot	arn:aws:ec2:us-east-1:123456789012:snapshot/snap-03866bf003EXAMPLE
AWS::EC2::Volume	arn:aws:ec2:us-east-1:123456789012:volume/vol-0912e04d78EXAMPLE
AWS::ECS::Service	arn:aws:ecs:us-east-1:123456789012:service/my-ecs-service
AWS::RDS::DBCluster	arn:aws:rds:us-east-2:123456789012:cluster:My-Cluster
AWS::RDS::DBInstance	arn:aws:rds:us-east-1:123456789012:db:My-SQL-Instance
AWS::S3::Bucket	arn:aws:s3:::amzn-s3-demo-bucket
AWS::SSM::ManagedInstance	arn:aws:ssm:us-east-1:123456789012:managed-instance/mi-0feadcf2d9EXAMPLE

- **`RESOURCE_ID`**: o ID curto de um tipo de recurso contido em um grupo de recursos. A tabela a seguir fornece valores de exemplo de `RESOURCE_ID` para tipos de recursos específicos em um grupo de recursos.

> **Note**
>
> RESOURCE_ID não tem suporte para tarefas do Run Command.

Tipo de recurso	Exemplo de RESOURCE_ID	
AWS::CloudWatch::Alarm	MyCloudWatchAlarm	
AWS::DynamoDB::Table	MyTable	
AWS::EC2::Instance	i-078a280217EXAMPLE	
AWS::EC2::Image	ami-02250b3732EXAMPLE	
AWS::EC2::SecurityGroup	sg-cEXAMPLE	
AWS::EC2::Snapshot	snap-03866bf003EXAMPLE	
AWS::EC2::Volume	vol-0912e04d78EXAMPLE	
AWS::ECS::Service	my-ecs-service	
AWS::RDS::DBCluster	My-Cluster	
AWS::RDS::DBInstance	My-SQL-Instance	
AWS::S3::Bucket	amzn-s3-demo-bucket	
AWS::SSM::ManagedInstance	mi-0feadcf2d9EXAMPLE	

> **Note**
>
> Se o grupo de recursos da AWS especificado incluir tipos de recursos que não produzem um valor de `RESOURCE_ID` e que não estiverem listados na tabela acima, o parâmetro `RESOURCE_ID` não será preenchido. Uma invocação de execução ainda ocorrerá para esse recurso. Nesses casos, use o pseudoparâmetro `TARGET_ID` no lugar, que será substituído pelo ARN completo do recurso.

Opções de programação da janela de manutenção e do período ativo

Ao criar uma janela de manutenção, você deve especificar com qual frequência a janela de manutenção é executada usando uma Expressão cron ou rate. Como opção, você pode especificar um intervalo de datas durante o qual a janela de manutenção pode ser executada em sua programação regular, bem como um fuso horário que servirá de base para essa programação regular.

Observe, no entanto, que a opção de fuso horário e as opções de data de início/término não se influenciam. Qualquer horário de data de início e de término que você especificar (com ou sem um deslocamento para o seu fuso horário) determina apenas o período válido durante o qual a janela de manutenção pode ser executada em sua programação. Uma opção de fuso horário determina o fuso horário internacional em que a programação da janela de manutenção se baseia durante seu período válido.

> **Note**
>
> As datas de início e de término devem ser especificadas no formato de carimbo de data e hora da ISO-8601. Por exemplo: `2021-04-07T14:29:00-08:00`
> Os fusos horários devem ser especificados no formato IANA (Internet Assigned Numbers Authority). Por exemplo: `America/Chicago`, `Europe/Berlin` ou `Asia/Tokyo`

Exemplos

- Exemplo 1: especifique uma data de início da janela de manutenção
- Exemplo 2: especifique a data de início e a data de término de uma janela de manutenção
- Exemplo 3: criar uma janela de manutenção que é executada somente uma vez

- Exemplo 4: especificar o número de dias de deslocamento de programação para uma janela de manutenção

Exemplo 1: especifique uma data de início da janela de manutenção

Digamos que você use a AWS Command Line Interface (AWS CLI) para criar uma janela de manutenção com as seguintes opções:

- `--start-date 2021-01-01T00:00:00-08:00`
- `--schedule-timezone "America/Los_Angeles"`
- `--schedule "cron(0 09 ? * WED *)"`

Por exemplo:

Linux & macOS

```
aws ssm create-maintenance-window \
    --name "My-LAX-Maintenance-Window" \
    --allow-unassociated-targets \
    --duration 3 \
    --cutoff 1 \
    --start-date 2021-01-01T00:00:00-08:00 \
    --schedule-timezone "America/Los_Angeles" \
    --schedule "cron(0 09 ? * WED *)"
```

Windows

```
aws ssm create-maintenance-window ^
    --name "My-LAX-Maintenance-Window" ^
    --allow-unassociated-targets ^
    --duration 3 ^
    --cutoff 1 ^
    --start-date 2021-01-01T00:00:00-08:00 ^
    --schedule-timezone "America/Los_Angeles" ^
    --schedule "cron(0 09 ? * WED *)"
```

Isso significa que a primeira execução da janela de manutenção não ocorrerá até depois da data e hora de início especificados, ou seja, até a meia-noite do fuso horário do Pacífico nos Estados Unidos, na sexta-feira, 1º de janeiro de 2021. (Esse fuso horário está cinco horas atrás do horário

UTC.) Nesse caso, a data e a hora de início do período da janela não representam quando as janelas de manutenção são executadas pela primeira vez. Em conjunto, os valores `--schedule-timezone` e `--schedule` significam que a janela de manutenção será executada toda quarta-feira, às 9h, no fuso horário do Pacífico nos Estados Unidos (representado por "America/Los Angeles" no formato IANA). A primeira execução no período ativado será na quarta-feira, 4 de janeiro de 2021, às 9h no horário Pacífico dos EUA.

Exemplo 2: especifique a data de início e a data de término de uma janela de manutenção

Suponha que, depois disso, você crie uma janela de manutenção com as seguintes opções:

- `--start-date 2019-01-01T00:03:15+09:00`
- `--end-date 2019-06-30T00:06:15+09:00`
- `--schedule-timezone "Asia/Tokyo"`
- `--schedule "rate(7 days)"`

Por exemplo:

Linux & macOS

```
aws ssm create-maintenance-window \
    --name "My-NRT-Maintenance-Window" \
    --allow-unassociated-targets \
    --duration 3 \
    --cutoff 1 \
    --start-date 2019-01-01T00:03:15+09:00 \
    --end-date 2019-06-30T00:06:15+09:00 \
    --schedule-timezone "Asia/Tokyo" \
    --schedule "rate(7 days)"
```

Windows

```
aws ssm create-maintenance-window ^
    --name "My-NRT-Maintenance-Window" ^
    --allow-unassociated-targets ^
    --duration 3 ^
    --cutoff 1 ^
    --start-date 2019-01-01T00:03:15+09:00 ^
    --end-date 2019-06-30T00:06:15+09:00 ^
```

```
    --schedule-timezone "Asia/Tokyo" ^
    --schedule "rate(7 days)"
```

O período permitido para esta janela de manutenção começa às 3:15h no horário padrão do Japão em 1º de janeiro de 2019. O período válido para esta janela de manutenção termina às 6:15 AM no horário padrão do Japão no domingo, 30 de junho de 2019. (Este fuso horário está nove horas à frente do horário UTC.) Em conjunto, os valores `--schedule-timezone` e `--schedule` significam que a janela de manutenção será executada às 3:15 AM, todas as terças-feiras, no fuso horário padrão do Japão (representado por "Asia/Tokyo" no formato IANA). Isso ocorre porque a janela de manutenção é executada a cada sete dias, sendo ativada às 3:15 AM de terça-feira, 1º de janeiro. A última execução será às 3:15 AM, horário padrão do Japão, na terça-feira, 25 de junho de 2019. Essa é a última terça-feira antes que o período da janela de manutenção permitida termine cinco dias depois.

Exemplo 3: criar uma janela de manutenção que é executada somente uma vez

Agora você cria uma janela de manutenção com esta opção:

- `--schedule "at(2020-07-07T15:55:00)"`

Por exemplo:

Linux & macOS

```
aws ssm create-maintenance-window \
    --name "My-One-Time-Maintenance-Window" \
    --schedule "at(2020-07-07T15:55:00)" \
    --duration 5 \
    --cutoff 2 \
    --allow-unassociated-targets
```

Windows

```
aws ssm create-maintenance-window ^
    --name "My-One-Time-Maintenance-Window" ^
    --schedule "at(2020-07-07T15:55:00)" ^
    --duration 5 ^
    --cutoff 2 ^
    --allow-unassociated-targets
```

Esta janela de manutenção é executada somente uma vez, às 15h55 no horário UTC em 7 de julho de 2020. A janela de manutenção tem permissão para ser executada por até cinco horas, conforme necessário, mas novas tarefas são impedidas de serem iniciadas duas horas antes do término do período da janela de manutenção.

Exemplo 4: especificar o número de dias de deslocamento de programação para uma janela de manutenção

Agora você cria uma janela de manutenção com esta opção:

```
--schedule-offset 2
```

Por exemplo:

Linux & macOS

```
aws ssm create-maintenance-window \
    --name "My-Cron-Offset-Maintenance-Window" \
    --schedule "cron(0 30 23 ? * TUE#3 *)" \
    --duration 4 \
    --cutoff 1 \
    --schedule-offset 2 \
    --allow-unassociated-targets
```

Windows

```
aws ssm create-maintenance-window ^
    --name "My-Cron-Offset-Maintenance-Window" ^
    --schedule "cron(0 30 23 ? * TUE#3 *)" ^
    --duration 4 ^
    --cutoff 1 ^
    --schedule-offset 2 ^
    --allow-unassociated-targets
```

Um deslocamento de programação é o número de dias de espera após a data e a hora especificadas por uma expressão CRON antes de executar a janela de manutenção.

No exemplo acima, a expressão CRON agenda a execução de uma janela de manutenção na terceira terça-feira de cada mês às 23h30:

```
--schedule "cron(0 30 23 ? * TUE#3 *)
```

No entanto, incluir `--schedule-offset 2` significa que a janela de manutenção só será executada dois dias após a terceira terça-feira de cada mês às 23h30.

Os deslocamentos de programação são compatíveis apenas com as expressões CRON.

Mais informações

- Referência: Expressões cron e rate para o Systems Manager
- Crie uma janela de manutenção usando o console
- Tutorial: Criar e configurar uma janela de manutenção usando a AWS CLI
- CreateMaintenanceWindow, na Referência de APIs do AWS Systems Manager
- create-maintenance-window na seção AWS Systems Manager da AWS CLI Command Reference
- Banco de dados de fuso horário no site da IANA

Registrar tarefas da janela de manutenção sem destinos

Para cada janela de manutenção criada, você pode especificar uma ou mais tarefas a serem executadas quando a janela de manutenção for executada. Na maioria dos casos, você deve especificar os recursos, ou destinos, nos quais a tarefa deve ser executada. No entanto, em alguns casos, você não precisa especificar destinos explicitamente na tarefa.

Um ou mais destinos devem ser especificados para as tarefas da janela de manutenção do Systems Manager do tipo Run Command. Dependendo da natureza da tarefa, os destinos serão opcionais para outros tipos de tarefas da janela de manutenção (Systems Manager Automation, AWS Lambda, e AWS Step Functions).

Para tipos de tarefa Lambda e Step Functions, a necessidade de um destino dependerá do conteúdo da função ou da máquina de estado que você criou.

> **Note**
>
> Quando uma tarefa tem destinos registrados, o Automation, AWS Lambda e as tarefas do AWS Step Functions resolvem os destinos a partir de grupos de recursos e tags e enviam uma invocação por recurso resolvido, o que resulta em várias invocações de tarefas. Mas

digamos, por exemplo, que você queira apenas uma invocação para uma tarefa do Lambda registrada em um grupo de recursos contendo mais de uma instância. Nesse caso, se você estiver trabalhando no AWS Management Console, escolha a opção Destino da tarefa não obrigatório na página Registrar tarefa do Lambda ou Editar tarefa do Lambda. Se você estiver usando o comando da AWS CLI, não especifique destinos usando o parâmetro `--targets` ao executar o comando register-task-with-maintenance-window ou o update-maintenance-window-task.

Em muitos casos, você não precisa especificar explicitamente um destino para uma tarefa de automação. Por exemplo, digamos que você esteja criando uma tarefa do tipo Automation para atualizar uma Amazon Machine Image (AMI) para Linux, usando o runbook `AWS-UpdateLinuxAmi`. Quando a tarefa for executada, a AMI será atualizada com os pacotes de distribuição Linux e o software da Amazon mais recentes disponível. As novas instâncias criadas na AMI já têm essas atualizações instaladas. Como o ID da AMI a ser atualizado é especificado nos parâmetros de entrada para o runbook, não há necessidade de especificar um destino novamente na tarefa da janela de manutenção.

Da mesma forma, suponha que você esteja usando o AWS Command Line Interface (AWS CLI) para registrar uma tarefa do Automation da janela de manutenção que usa o runbook `AWS-RestartEC2Instance`. Como o nó a ser reiniciado é especificado no argumento `--task-invocation-parameters`, você não precisa especificar também uma opção de `--targets`.

Note

Para tarefas de janela de manutenção sem um destino especificado, você não pode fornecer valores para `--max-errors` e `--max-concurrency`. Em vez disso, o sistema insere um valor de espaço reservado de 1, que pode ser relatado na resposta a comandos como describe-maintenance-window-tasks e get-maintenance-window-task. Esses valores não afetam a execução da tarefa e podem ser ignorados.

O exemplo a seguir demonstra a omissão das opções de `--targets`, `--max-errors` e `--max-concurrency` para uma tarefa da janela de manutenção sem destino.

Linux & macOS

```
aws ssm register-task-with-maintenance-window \
```

```
    --window-id "mw-ab12cd34eEXAMPLE" \
    --service-role-arn "arn:aws:iam::123456789012:role/
MaintenanceWindowAndAutomationRole" \
    --task-type "AUTOMATION" \
    --name "RestartInstanceWithoutTarget" \
    --task-arn "AWS-RestartEC2Instance" \
    --task-invocation-parameters "{\"Automation\":{\"Parameters\":{\"InstanceId\":
[\"i-02573cafcfEXAMPLE\"]}}}" \
    --priority 10
```

Windows

```
aws ssm register-task-with-maintenance-window ^
    --window-id "mw-ab12cd34eEXAMPLE" ^
    --service-role-arn "arn:aws:iam::123456789012:role/
MaintenanceWindowAndAutomationRole" ^
    --task-type "AUTOMATION" ^
    --name "RestartInstanceWithoutTarget" ^
    --task-arn "AWS-RestartEC2Instance" ^
    --task-invocation-parameters "{\"Automation\":{\"Parameters\":{\"InstanceId\":
[\"i-02573cafcfEXAMPLE\"]}}}" ^
    --priority 10
```

> **Note**
>
> Para tarefas da janela de manutenção registradas antes de 23 de dezembro de 2020: se você especificou destinos para a tarefa e eles não forem mais necessários, você poderá atualizar essa tarefa para remover os destinos usando o console do Systems Manager ou o comando update-maintenance-window-task da AWS CLI.

Mais informações

- Mensagens de erro: “Tarefas da janela de manutenção sem destinos não suportam valores MaxConcurrency” e “Tarefas da janela de manutenção sem destinos não suportam valores MaxErrors”

Solução de problemas das janelas

Use as informações a seguir para ajudar a solucionar problemas com as janelas de manutenção.

Tópicos

- Erro na edição da tarefa: na página de edição de uma tarefa da janela de manutenção, a lista de funções do IAM retorna uma mensagem de erro: "Não foi possível encontrar a função da janela de manutenção do IAM especificada para esta tarefa. Ela pode ter sido excluída, ou pode não ter sido criada ainda."
- Nem todos os destinos da janela de manutenção são atualizados
- A tarefa falha com o status de invocação de tarefa: “O perfil fornecido não contém as permissões corretas do SSM”.
- Tarefa falha com mensagem de erro: “Falha na etapa quando ela estiver validando e resolvendo as entradas da etapa”
- Mensagens de erro: “Tarefas da janela de manutenção sem destinos não suportam valores MaxConcurrency” e “Tarefas da janela de manutenção sem destinos não suportam valores MaxErrors”

Erro na edição da tarefa: na página de edição de uma tarefa da janela de manutenção, a lista de funções do IAM retorna uma mensagem de erro: "Não foi possível encontrar a função da janela de manutenção do IAM especificada para esta tarefa. Ela pode ter sido excluída, ou pode não ter sido criada ainda."

Problema 1: a função da janela de manutenção do AWS Identity and Access Management (IAM) que você especificou originalmente especificada foi excluída depois que você criou a tarefa.

Possible fix (Correção possível): 1) selecione outro perfil de janela de manutenção do IAM, se houver algum em sua conta, ou crie um novo e selecione-o para a tarefa.

Problema 2: se a tarefa foi criada usando a AWS Command Line Interface (AWS CLI), o AWS Tools for Windows PowerShell ou um AWS SDK, um nome de função do IAM não existente pode não ter sido especificado. Por exemplo, a função da janela de manutenção do IAM pode ter sido excluída antes de você criar a tarefa, ou o nome da função pode ter sido digitado incorretamente, como **`myrole`** em vez de **`my-role`**.

Possible fix (Correção possível): selecione o nome correto do perfil de janela de manutenção do IAM que deseja usar ou crie um novo específico para a tarefa.

Nem todos os destinos da janela de manutenção são atualizados

Problema: você percebe que as tarefas da janela de manutenção não foram executadas em todos os recursos determinados como destino pela janela de manutenção. Por exemplo, nos resultados da

execução da janela de manutenção, a tarefa desse recurso é marcada como falha ou com o tempo limite expirado.

Solução: Os motivos mais comuns para uma tarefa de janela de manutenção que não está sendo executada em um recurso de destino envolvem conectividade e disponibilidade. Por exemplo:

- O Systems Manager perdeu a conexão com o recurso antes ou durante a operação da janela de manutenção.
- O recurso estava offline ou parado durante a operação da janela de manutenção.

Você pode aguardar a próxima janela de manutenção agendada para executar tarefas nos recursos. Você pode executar manualmente as tarefas da janela de manutenção nos recursos que não estavam disponíveis ou estavam offline.

A tarefa falha com o status de invocação de tarefa: “O perfil fornecido não contém as permissões corretas do SSM”.

Problema: você especificou um perfil de serviço de janela de manutenção para uma tarefa, mas a tarefa não é executada com êxito, e o status de invocação da tarefa informa “O perfil fornecido não contém as permissões corretas do SSM”.

- Solução: no Tarefa 1: crie uma política personalizada para seu perfil de serviço de janela de manutenção usando o console, fornecemos uma política básica que você pode anexar ao seu perfil de serviço de janela de manutenção personalizada. Essa política inclui as permissões necessárias para diversos cenários de tarefas. Porém, dada a grande diversidade de tarefas que podem ser executadas, talvez seja necessário fornecer permissões adicionais na política para a sua função de janela de manutenção.

  Por exemplo, algumas ações da Automação trabalham com pilhas do AWS CloudFormation. Por isso, pode ser necessário adicionar as permissões `cloudformation:CreateStack`, `cloudformation:DescribeStacks` e `cloudformation:DeleteStack` extras à política para seu perfil de serviço de janela de manutenção.

  Outro exemplo: o runbook `AWS-CopySnapshot` do Automation requer permissão para criar um snapshot do Amazon Elastic Block Store (Amazon EBS). Por isso, pode ser necessário adicionar a permissão `ec2:CreateSnapshot`.

Para obter informações sobre as permissões de perfil necessárias para um runbook de automação gerenciado pela AWS, consulte as descrições de runbooks na Referência de runbooks do AWS Systems Manager Automation.

Para informações sobre as permissões de função necessárias para um documento do SSM gerenciado pela AWS, revise o conteúdo do documento na seção Documents (Documentos) do console do Systems Manager.

Para informações sobre as permissões de função necessárias para tarefas do Step Functions, tarefas do Lambda e runbooks personalizados do Automation e documentos do SSM, verifique os requisitos de permissão com o autor desses recursos.

Tarefa falha com mensagem de erro: “Falha na etapa quando ela estiver validando e resolvendo as entradas da etapa”

Problema: um runbook do Automation ou um documento de comando do Systems Manager que você estiver usando em uma tarefa requer que você especifique entradas como `InstanceId` ou `SnapshotId`, mas um valor não é fornecido ou não é fornecido corretamente.

- Solução 1: se sua tarefa estiver determinando um único recurso como destino, por exemplo um único nó ou um único snapshot, insira seu ID nos parâmetros de entrada para a tarefa.
- Solução 2: se sua tarefa estiver determinando vários recursos como destino, como criar imagens de vários nós ao usar o runbook `AWS-CreateImage`, você pode usar um dos pseudoparâmetros suportados para tarefas de janela de manutenção nos parâmetros de entrada para representar IDs dos nós em comandos.

  Os comandos a seguir registram uma tarefa do Systems Manager Automation com uma janela de manutenção usando a:AWS CLI. O `--targets` indica um ID de destino para a janela de manutenção. Além disso, mesmo que o parâmetro `--targets` especifique um ID de destino de janela, os parâmetros do runbook do Automation exigem que um ID do nó gerenciado seja fornecido. Nesse caso, o comando usa o pseudoparâmetro `{{RESOURCE_ID}}` como o valor `InstanceId`.

  Comando da AWS CLI:

Linux & macOS

O comando a seguir reinicia as instâncias do Amazon Elastic Compute Cloud (Amazon EC2) que pertencem ao grupo de destino da janela de manutenção com o ID e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE.

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --targets Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE \
    --task-arn "AWS-RestartEC2Instance" \
    --service-role-arn arn:aws:iam::123456789012:role/
MyMaintenanceWindowServiceRole \
    --task-type AUTOMATION \
    --task-invocation-parameters
 "Automation={DocumentVersion=5,Parameters={InstanceId='{{RESOURCE_ID}}'}}" \
    --priority 0 --max-concurrency 10 --max-errors 5 --name "My-Restart-EC2-
Instances-Automation-Task" \
    --description "Automation task to restart EC2 instances"
```

Windows

```
aws ssm register-task-with-maintenance-window ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --targets Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE ^
    --task-arn "AWS-RestartEC2Instance" ^
    --service-role-arn arn:aws:iam::123456789012:role/
MyMaintenanceWindowServiceRole ^
    --task-type AUTOMATION ^
    --task-invocation-parameters
 "Automation={DocumentVersion=5,Parameters={InstanceId='{{RESOURCE_ID}}'}}" ^
    --priority 0 --max-concurrency 10 --max-errors 5 --name "My-Restart-EC2-
Instances-Automation-Task" ^
    --description "Automation task to restart EC2 instances"
```

Para obter mais informações sobre como trabalhar com pseudoparâmetros para tarefas da janela de manutenção, consulte Usar pseudoparâmetros ao registrar tarefas da janela de manutenção e Exemplos de registro de tarefas.

Mensagens de erro: “Tarefas da janela de manutenção sem destinos não suportam valores MaxConcurrency” e “Tarefas da janela de manutenção sem destinos não suportam valores MaxErrors”

Problema: quando você registrar uma tarefa do tipo Run Command, você deverá especificar pelo menos um destino no qual executar a tarefa. Para outros tipos de tarefas, (Automation, AWS Lambda e AWS Step Functions) dependendo da natureza da tarefa, os destinos serão opcionais. As opções `MaxConcurrency` (o número de recursos para executar uma tarefa ao mesmo tempo) e `MaxErrors` (o número de falhas para executar a tarefa nos recursos de destino antes que a tarefa falhe) não são necessárias ou suportadas para tarefas de janela de manutenção que não especificam destinos. O sistema gera essas mensagens de erro se os valores forem especificados para qualquer uma dessas opções quando nenhum destino para a tarefa for especificado.

Solução: se você receber um desses erros, remova os valores de simultaneidade e limite de erro antes de continuar a registrar ou atualizar a tarefa de janela de manutenção.

Para obter mais informações sobre como executar tarefas que não especificam destinos, consulte Registrar tarefas da janela de manutenção sem destinos no Manual do usuário do AWS Systems Manager.

AWS Systems Manager Quick Setup

Use o Quick Setup, uma ferramenta do AWS Systems Manager, para configurar rapidamente os serviços e recursos da Amazon Web Services usados com frequência de acordo com as melhores práticas recomendadas. A Quick Setup simplifica a configuração de serviços, incluindo o Systems Manager, automatizando tarefas comuns ou recomendadas. Essas tarefas incluem, por exemplo, a criação de AWS Identity and Access Management funções de perfil de instância (IAM) e configuração de práticas recomendadas operacionais, como verificações periódicas de patches e coleta de inventário. Não há custo para usar o Quick Setup. No entanto, é possível haver cobrança de custos com base no tipo de serviço que você configurou e nos limites de uso, sem taxas pelos serviços usados para configurar seu serviço. Para começar a usar o Quick Setup, abra o Systems Manager console (Console do gerenciador de sistemas). No painel de navegação, escolha Quick Setup.

> **Note**
>
> Se você tiver sido direcionado para Quick Setup a fim de obter ajuda na configuração de suas instâncias para gerenciamento pelo Systems Manager, conclua o procedimento em Configurar o gerenciamento de host do Amazon EC2 usando a Quick Setup.

Quais são os benefícios do Quick Setup?

Os benefícios do Quick Setup incluem o seguinte:

- Simplifique a configuração do serviço e

  O Quick Setup orienta você na configuração das práticas recomendadas operacionais e implanta automaticamente essas configurações. O Quick Setup exibe uma exibição em tempo real do status de implantação da configuração.
- Implante configurações automaticamente em várias contas

  Você pode usar o Quick Setup em uma Conta da AWS individual ou por meio de várias Contas da AWS e Regiões da AWS ao integrar com o AWS Organizations. O uso do Quick Setup em várias contas ajuda a garantir que sua organização mantenha configurações consistentes.
- Elimine o desvio da configuração

  O desvio de configuração ocorre sempre que um usuário faz qualquer alteração em um serviço ou recurso que entra em conflito com as seleções feitas por meio do Quick Setup. O Quick Setup verifica periodicamente se há desvio de configuração e tenta corrigi-lo.

Quem deve usar o Quick Setup?

O Quick Setup será mais benéfico para os clientes que já tiverem alguma experiência com os serviços e recursos que estiverem configurando, e quiserem simplificar o processo de configuração. Se você não estiver familiarizado com o AWS service (Serviço da AWS) que está configurando com a Quick Setup, recomendamos que você saiba mais sobre o serviço. Revise o conteúdo no Guia do Usuário relevante antes de criar uma configuração com o Quick Setup.

Disponibilidade do Quick Setup nas Regiões da AWS

No seguinte Regiões da AWS, você pode usar todos os tipos de Quick Setup configuração para uma organização inteira, conforme configurado em AWS Organizations, ou somente para as contas organizacionais e regiões que você escolher. Você também pode usar Quick Setup com apenas uma única conta nessas regiões.

- Leste dos EUA (Ohio)
- Leste dos EUA (Norte da Virgínia)
- Oeste dos EUA (Norte da Califórnia)

- Oeste dos EUA (Oregon)
- Ásia-Pacífico (Mumbai)
- Ásia-Pacífico (Seul)
- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Tóquio)
- Canadá (Central)
- Europa (Frankfurt)
- Europa (Estocolmo)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Paris)
- América do Sul (São Paulo)

Nas seguintes regiões, somente o tipo de configuração do Host Management está disponível para contas individuais:

- Europa (Milão)
- Ásia-Pacífico (Hong Kong)
- Oriente Médio (Bahrein)
- China (Pequim)
- China (Ningxia)
- AWS GovCloud (Leste dos EUA)
- AWS GovCloud (Oeste dos EUA)

Para ver uma lista de regiões com suporte, consulte a coluna Region (Região) em Systems Manager service endpoints (Endpoints de serviçso do Systems Manager) no Referência geral da Amazon Web Services.

Conceitos básicos do Quick Setup

Utilize as informações deste tópico para ajudar você a se preparar para o uso da Quick Setup.

Tópicos

- Perfis e permissões do IAM para integração com a Quick Setup
- Integração manual para trabalhar com a API do Quick Setup de forma programática

Perfis e permissões do IAM para integração com a Quick Setup

O Quick Setup lançou uma nova experiência de console e uma nova API. Agora você pode interagir com essa API usando o console, a AWS CLI, o AWS CloudFormation e os SDKs. Se você optar pela nova experiência, suas configurações existentes serão recriadas usando a nova API. Dependendo do número de configurações existentes na sua conta, esse processo pode levar vários minutos.

Para usar o novo console da Quick Setup, é necessário ter permissões para as seguintes ações:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm-quicksetup:*",
                "cloudformation:DescribeStackSetOperation",
                "cloudformation:ListStacks",
                "cloudformation:DescribeStacks",
                "cloudformation:DescribeStackResources",
                "cloudformation:ListStackSetOperations",
                "cloudformation:ListStackInstances",
                "cloudformation:DescribeStackSet",
                "cloudformation:ListStackSets",
                "cloudformation:DescribeStackInstance",
                "cloudformation:DescribeOrganizationsAccess",
                "cloudformation:ActivateOrganizationsAccess",
                "cloudformation:GetTemplate",
                "cloudformation:ListStackSetOperationResults",
                "cloudformation:DescribeStackEvents",
                "cloudformation:UntagResource",
                "ec2:DescribeInstances",
                "ssm:DescribeAutomationExecutions",
                "ssm:GetAutomationExecution",
                "ssm:ListAssociations",
                "ssm:DescribeAssociation",
```

```
                "ssm:GetDocument",
                "ssm:ListDocuments",
                "ssm:DescribeDocument",
                "ssm:ListResourceDataSync",
                "ssm:DescribePatchBaselines",
                "ssm:GetPatchBaseline",
                "ssm:DescribeMaintenanceWindows",
                "ssm:DescribeMaintenanceWindowTasks",
                "ssm:GetOpsSummary",
                "organizations:DeregisterDelegatedAdministrator",
                "organizations:DescribeAccount",
                "organizations:DescribeOrganization",
                "organizations:ListDelegatedAdministrators",
                "organizations:ListRoots",
                "organizations:ListParents",
                "organizations:ListOrganizationalUnitsForParent",
                "organizations:DescribeOrganizationalUnit",
                "organizations:ListAWSServiceAccessForOrganization",
                "s3:GetBucketLocation",
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "resource-groups:ListGroups",
                "iam:ListRoles",
                "iam:ListRolePolicies",
                "iam:GetRole",
                "iam:CreatePolicy",
                "organizations:RegisterDelegatedAdministrator",
                "organizations:EnableAWSServiceAccess",
                "cloudformation:TagResource"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudformation:RollbackStack",
                "cloudformation:CreateStack",
                "cloudformation:UpdateStack",
                "cloudformation:DeleteStack"
            ],
            "Resource": [
                "arn:aws:cloudformation:*:*:stack/StackSet-AWS-QuickSetup-*",
                "arn:aws:cloudformation:*:*:stack/AWS-QuickSetup-*",
                "arn:aws:cloudformation:*:*:type/resource/*",
```

```
                "arn:aws:cloudformation:*:*:stack/StackSet-SSMQuickSetup"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudformation:CreateStackSet",
                "cloudformation:UpdateStackSet",
                "cloudformation:DeleteStackSet",
                "cloudformation:DeleteStackInstances",
                "cloudformation:CreateStackInstances",
                "cloudformation:StopStackSetOperation"
            ],
            "Resource": [
                "arn:aws:cloudformation:*:*:stackset/AWS-QuickSetup-*",
                "arn:aws:cloudformation:*:*:stackset/SSMQuickSetup",
                "arn:aws:cloudformation:*:*:type/resource/*",
                "arn:aws:cloudformation:*:*:stackset-target/AWS-QuickSetup-*:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:AttachRolePolicy",
                "iam:DetachRolePolicy",
                "iam:GetRolePolicy",
                "iam:PutRolePolicy"
            ],
            "Resource": [
                "arn:aws:iam::*:role/AWS-QuickSetup-*",
                "arn:aws:iam::*:role/service-role/AWS-QuickSetup-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/AWS-QuickSetup-*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ssm-quicksetup.amazonaws.com"
                }
            }
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:DeleteAssociation",
                "ssm:CreateAssociation",
                "ssm:StartAssociationsOnce"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ssm:StartAutomationExecution",
            "Resource": "arn:aws:ssm:*:*:automation-definition/AWS-
EnableExplorer:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetOpsSummary",
                "ssm:CreateResourceDataSync",
                "ssm:UpdateResourceDataSync"
            ],
            "Resource": "arn:aws:ssm:*:*:resource-data-sync/AWS-QuickSetup-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "accountdiscovery.ssm.amazonaws.com",
                        "ssm.amazonaws.com",
                        "ssm-quicksetup.amazonaws.com",
                        "stacksets.cloudformation.amazonaws.com"
                    ]
                }
            },
            "Resource": "*"
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::*:role/aws-
service-role/stacksets.cloudformation.amazonaws.com/
AWSServiceRoleForCloudFormationStackSetsOrgAdmin"
        }
    ]
}
```

Para restringir os usuários às permissões de somente leitura, permita somente as operações `ssm-quicksetup:List*` e `ssm-quicksetup:Get*` para a API Quick Setup.

Durante a integração, o Quick Setup cria as seguintes regras do AWS Identity and Access Management (IAM) em seu nome:

- `AWS-QuickSetup-LocalExecutionRole`: concede permissões do AWS CloudFormation para usar qualquer modelo, excluindo o modelo de política de patch, e cria os recursos necessários.
- `AWS-QuickSetup-LocalAdministrationRole`: concede permissões para o AWS CloudFormation assumir a `AWS-QuickSetup-LocalExecutionRole`.
- `AWS-QuickSetup-PatchPolicy-LocalExecutionRole`: concede permissões do AWS CloudFormation para usar o modelo de política de patch, e cria os recursos necessários.
- `AWS-QuickSetup-PatchPolicy-LocalAdministrationRole`: concede permissões para o AWS CloudFormation assumir a `AWS-QuickSetup-PatchPolicy-LocalExecutionRole`.

Se você estiver integrando uma conta de gerenciamento (a conta que você usa para criar uma organização no AWS Organizations), a Quick Setup também criará os seguintes perfis em seu nome:

- `AWS-QuickSetup-SSM-RoleForEnablingExplorer`: concede permissões para a execução do `AWS-EnableExplorer` runbook de automação. O runbook `AWS-EnableExplorer` configura o Explorer, uma ferramenta do Systems Manager, para exibir informações de várias Contas da AWS e Regiões da AWS.
- `AWSServiceRoleForAmazonSSM`: uma função vinculada ao serviço que concede acesso do ao recursos da AWS gerenciados e usados pelo Systems Manager.
- `AWSServiceRoleForAmazonSSM_AccountDiscovery`: uma função vinculada ao serviço que concede permissões ao Systems Manager para chamar Serviços da AWS, a fim de descobrir

informações da Conta da AWS ao sincronizar os dados. Para obter mais informações, consulte Usar perfis para coletar informações da Conta da AWS para o OpsCenter e o Explorer.

Ao integrar uma conta de gerenciamento, o Quick Setup habilita o acesso confiável entre o AWS Organizations e o CloudFormation para implantar as configurações do Quick Setup em toda a sua organização. Para habilitar o acesso confiável, sua conta de gerenciamento deve ter permissões de administrador. Após a integração, você não precisa mais de permissões de administrador. Para obter mais informações, consulte Habilitar o acesso confiável com o Organizations.

Para obter mais informações sobre tipos de conta do AWS Organizations, consulte Terminologia e conceitos do AWS Organizations no Guia do usuário do AWS Organizations.

> **Note**
>
> O Quick Setup usa StackSets do AWS CloudFormation para implantar suas configurações entre Contas da AWS e regiões. Se o número de contas de destino multiplicado pelo número de regiões exceder dez mil, a implantação da configuração falhará. É recomendável analisar seu caso de uso e criar configurações que usem menos destinos para comportar o crescimento de sua organização. As instâncias de pilhas não são implantadas na conta de gerenciamento de sua organização. Para obter mais informações, consulte Considerations when creating a stack set with service-managed permissions.

Integração manual para trabalhar com a API do Quick Setup de forma programática

Se você usa o console para trabalhar com o Quick Setup, o serviço gerencia as etapas de integração para você. Se você planeja usar SDKs ou AWS CLI trabalhar com a API do Quick Setup, ainda pode usar o console para concluir as etapas de integração, sem precisar executá-las manualmente. No entanto, alguns clientes precisam concluir as etapas de integração do Quick Setup forma programática sem interagir com o console. Se esse método for adequado ao seu caso de uso, é necessário executar as etapas a seguir. Todas essas etapas devem ser concluídas em sua conta de gerenciamento do AWS Organizations.

Para concluir a integração manual do Quick Setup

1. Ative o acesso confiável para o AWS CloudFormation com o Organizations. Isso fornece à conta de gerenciamento as permissões para criar e gerenciar StackSets para sua organização. Você pode usar a ação da API `ActivateOrganizationsAccess` do AWS CloudFormation para

concluir essa etapa. Para obter mais informações, consulte ActivateOrganizationsAccess na AWS CloudFormation API Reference.

2. Permita a integração do Systems Manager com o Organizations. Assim, o Systems Manager pode criar uma função vinculada ao serviço em todas as contas de sua organização. Isso também permite que o Systems Manager realize as operações em seu nome em sua organização e em suas contas. Você pode usar a ação da API `EnableAWSServiceAccess` do AWS Organizations para concluir essa etapa. A entidade principal de serviço do Systems Manager é `ssm.amazonaws.com` .Para obter mais informações, consulte EnableAWSServiceAccess na AWS Organizations API Reference.
3. Crie o perfil do IAM necessário para o Explorer. Isso permite que o Quick Setup crie painéis para suas configurações para que você possa visualizar os status de implantação e associação. Crie um perfil do IAM de cluster e anexe a política gerenciada do `AWSSystemsManagerEnableExplorerExecutionPolicy`. Modifique a política de confiança de uma função para que corresponda ao seguinte. Substitua cada *`account ID`* por suas próprias informações.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "account ID"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:*:ssm:*:111122223333:automation-
execution/*"
                }
            }
        }
    ]
}
```

4. Atualize a configuração do serviço do Quick Setup para o Explorer. Você pode usar a ação da API `UpdateServiceSettings` do Quick Setup para concluir essa etapa. Revise o ARN do perfil do IAM que você criou na etapa anterior para o parâmetro de solicitação `ExplorerEnablingRoleArn`. Para obter mais informações, consulte UpdateServiceSettings na Quick Setup API Reference.
5. Crie os perfis do IAM necessários para que os StackSets do AWS CloudFormation os usem. Você deve criar um perfil de execução e um perfil de administração.

   a. Crie a função de execução. O perfil de execução deve ter pelo menos uma das políticas gerenciadas `AWSQuickSetupDeploymentRolePolicy` ou `AWSQuickSetupPatchPolicyDeploymentRolePolicy` anexadas. Se você estiver criando apenas configurações de políticas de patch, poderá usar a política gerenciada `AWSQuickSetupPatchPolicyDeploymentRolePolicy`. Todas as outras configurações usam a política `AWSQuickSetupDeploymentRolePolicy`. Modifique a política de confiança de uma função para que corresponda ao seguinte. Substitua cada *`ID da conta`* e *`nome do perfil administrativo`* por suas informações.

      JSON

      ```
      {
          "Version": "2012-10-17",
          "Statement": [
              {
                  "Effect": "Allow",
                  "Principal": {
                      "AWS": "arn:aws:iam::111122223333:role/administration
       role name"
                  },
                  "Action": "sts:AssumeRole"
              }
          ]
      }
      ```

   b. Crie o perfil administrativo. A política de permissões deve corresponder ao seguinte. Substitua cada *`ID da conta`* e *`nome do perfil de execução`* por suas informações.

      JSON

      ```
      {
      ```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sts:AssumeRole"
            ],
            "Resource": "arn:*:iam::111122223333:role/execution role
 name",
            "Effect": "Allow"
        }
    ]
}
```

Modifique a política de confiança de uma função para que corresponda ao seguinte. Substitua cada *account ID* por suas próprias informações.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudformation.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn":
 "arn:aws:cloudformation:*:111122223333:stackset/AWS-QuickSetup-*"
                }
            }
        }
    ]
}
```

Usar um administrador delegado para a Quick Setup

Depois que você registra uma conta de administrador delegado para o Quick Setup, os usuários com as devidas permissões naquela conta são capazes de criar, atualizar, visualizar e excluir gerenciadores de configuração cujo alvo são as unidades organizacionais na sua estrutura do AWS Organizations. Essa conta de administrador delegado também pode gerenciar gerenciadores de configuração criados anteriormente pela conta gerencial da sua organização.

A conta gerencial no Organizations pode designar uma conta dentro da sua organização como administrador delegado. Ao registrar uma conta como administrador delegado para o Quick Setup, esta se torna automaticamente um administrador delegado de StackSets do AWS CloudFormation e para o Systems Manager Explorer, uma vez que esses serviços são necessários para implantar e monitorar configurações de Quick Setup.

> **Note**
>
> No momento, o tipo de configuração da política de patch não é aceito pelo Quick Setup administrador delegado do . As configurações de políticas de patch para uma organização precisam ser criadas e mantidas na conta gerencial de uma organização. Para obter mais informações, consulte the section called “Criação de uma política de patch”.

Os tópicos a seguir descrevem como registrar e cancelar o registro de um administrador delegado para a Quick Setup.

Tópicos

- Registrar um administrador delegado para a Quick Setup
- Cancelar o registro de um administrador delegado para a Quick Setup

Registrar um administrador delegado para a Quick Setup

Use o procedimento a seguir para registrar um administrador delegado da Quick Setup.

Para registrar um administrador delegado da Quick Setup

1. Faça login na conta de gerenciamento do AWS Organizations.
2. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
3. No painel de navegação, escolha Quick Setup.

4. Escolha Settings.
5. Na seção Administrador delegado para o Quick Setup, verifique se você configurou a função vinculada ao serviço e as opções de acesso ao serviço necessárias. Se for necessário, selecione os botões Create role (Criar regra) e Enable access (Ativar acesso) para configurar essas opções.
6. Em Account ID (ID da conta), insira o ID da Conta da AWS. Essa conta deve ser uma conta-membro no AWS Organizations.
7. Selecione Registrar administrador delegado.

Cancelar o registro de um administrador delegado para a Quick Setup

Use o procedimento a seguir para cancelar o registro de um administrador delegado da Quick Setup.

Para cancelar o registro de um administrador delegado da Quick Setup

1. Faça login na conta de gerenciamento do AWS Organizations.
2. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
3. No painel de navegação, escolha Quick Setup.
4. Escolha Settings.
5. Na seção Administrador delegado da Quick Setup, selecione Cancelar registro na lista suspensa Ações.
6. Selecione Confirmar.

Aprender a terminologia e os detalhes do Quick Setup

A Quick Setup, uma ferramenta do AWS Systems Manager, exibe os resultados de todos os gerenciadores de configuração que você criou em todas as Regiões da AWS na tabela Gerenciadores de configuração na página inicial do Quick Setup. Nessa página, você pode View details (Exibir detalhes) de cada configuração, excluir configurações do menu suspenso Actions (Ações), ou Create (Criar) configurações. A tabela Gerenciadores de configuração contém as seguintes informações:

- Nome: o nome do gerenciador de configuração, se fornecido quando criado.
- Configuration type (Tipo de configuração): o tipo de configuração escolhido ao criar a configuração.
- Versão: a versão do tipo de configuração atualmente implantada.

- Organizational units (Unidades organizacionais): exibe as unidades organizacionais (UOs) nas quais a configuração é implantada se você escolher um conjunto de destinos Custom (Personalizado). Unidades organizacionais e metas personalizadas só estão disponíveis para a conta de gerenciamento da sua organização. A conta de gerenciamento é a conta que você usa para criar uma organização no AWS Organizations.
- Tipo de implantação — Indica se a implantação se aplica a toda a organização (`Organizational`) ou somente à sua conta (`Local`).
- Regions (Regiões): as regiões nas quais a configuração é implantada se você escolheu um conjunto de destinos Custom (Personalizado) ou destinos dentro da sua Current account (Conta atual).
- Deployment status (Status da implantação): o status da implantação indica se o AWS CloudFormation implantou com sucesso a instância de destino ou pilha. As instâncias de destino e pilha contêm as opções de configuração escolhidas durante a criação da configuração.
- Status da associação: o status da associação é o estado de todas as associações criadas pela configuração que você criou. As associações de todos os destinos devem ser executadas com êxito, caso contrário, o status será Failed (Com falha).

  O Quick Setup cria e executa uma associação do State Manager para cada destino de configuração. O State Manager é uma ferramenta do AWS Systems Manager.

Para ver as configurações implantadas na região em que você está navegando no momento, selecione a guia Configurações.

Detalhes da configuração

A página Configuration details (Detalhes da configuração) exibe informações sobre a implantação da configuração e suas associações relacionadas. A partir dessa página, você pode editar opções de configuração, atualizar destinos ou excluir a configuração. Você também pode exibir os detalhes de cada implantação de configuração para obter mais informações sobre as associações.

Conforme o tipo de configuração, um ou mais dos seguintes grafos de status são exibidos:

Status da implantação da configuração

Exibe o número de implantações que foram bem-sucedidas, falharam ou estão em execução ou pendentes. As implantações ocorrem nas contas e regiões de destino especificadas que contêm nós afetados pela configuração.

Status da associação de configuração

Exibe o número de associações do State Manager que foram bem-sucedidas, falharam ou estão pendentes. O Quick Setup cria uma associação em cada implantação para as opções de configuração selecionadas.

Setup status (Status da configuração)

Exibe o número de ações executadas pelo tipo de configuração e seus status atuais.

Resource compliance (Conformidade de recursos)

Exibe o número de recursos que estão em conformidade com a política especificada da configuração.

A tabela Configuration details (Detalhes da configuração) exibe informações sobre a implantação da sua configuração. Você pode ver mais detalhes sobre cada implantação selecionando a implantação e, em seguida, escolhendoView details (Visualizar os detalhes). A página de detalhes de cada implantação exibe as associações implantadas nos nós nessa implantação.

Editar e excluir a configuração

Você pode editar as opções de configuração de uma configuração a partir da página Configuration details (Detalhes da configuração) escolhendo Actions (Ações) e, em seguida, Edit configuration options (Editar opções de configuração). Quando você adiciona novas opções à configuração, o Quick Setup executa suas implantações e cria novas associações. Quando você remove opções de uma configuração, o Quick Setup executa suas implantações e remove todas as associações relacionadas.

> **Note**
>
> Você pode editar as configurações do Quick Setup para sua conta a qualquer momento. Para editar uma configuração Organization (Organização), o Configuration status (Status da configuração) deve ser Success (Êxito) ou Failed (Com falha).

Você também pode atualizar os destinos incluídos nas configurações escolhendo Actions (Ações) e Add OUs (Adicionar UOs), Add Regions (Adicionar regiões), Remove OUs (Remover UOs) ou Remove Regions (Remover regiões). Se sua conta não estiver configurada como a conta de gerenciamento ou se você criou a configuração apenas para a conta atual, não será possível

atualizar as unidades organizacionais (UOs) de destino. A remoção de uma região ou UO remove as associações dessas regiões ou UOs.

Periodicamente, o Quick Setup lança novas versões das configurações. Você pode selecionar a opção Upgrade configuration para atualizar sua configuração para a versão mais recente.

Você pode excluir uma configuração do Quick Setup escolhendo a configuração e, em seguida Actions (Ações), e, depois Delete configuration (Excluir configuração). Ou então, você pode excluir a configuração da página Configuration details (Detalhes da configuração), no menu suspenso Actions (Ações) e, depois Delete configuration (Excluir configuração). O Quick Setup solicita que você Remove all OUs and Regions (Remover todas as UOs e regiões), o que pode levar algum tempo. A exclusão de uma configuração também exclui todas as associações relacionadas. Esse processo de exclusão em duas etapas remove todos os recursos implantados de todas as contas e regiões e, em seguida, exclui a configuração.

Conformidade de configuração

É possível verificar se suas instâncias estão em conformidade com as associações criadas por suas configurações no Explorer ou no Compliance, ambos ferramentas do AWS Systems Manager. Para saber mais sobre conformidade, consulte Conheça mais sobre a conformidade. Para saber mais sobre a exibição de conformidade no Explorer, consulte AWS Systems Manager Explorer.

Usar a API da Quick Setup para gerenciar configurações e implantações

Você pode usar a API fornecida pelo Quick Setup para criar e gerenciar configurações e implantações usando a AWS CLI ou seu SDK preferido. Você também pode usar o AWS CloudFormation para criar um recurso do gerenciador de configurações que implanta configurações. Usando a API, você cria gerenciadores de configuração que implantam definições de configuração. As definições de configuração contêm todas as informações necessárias para implantar um tipo de configuração específico. Para obter mais informações sobre como usar a API do Quick Setup, consulte a Quick Setup API Reference.

Os exemplos a seguir demonstram como criar gerenciadores de configuração usando a AWS CLI e o AWS CloudFormation.

AWS CLI

```
aws ssm-quicksetup create-configuration-manager \
--name configuration manager name \
--description Description of your configuration manager
```

```
--configuration-definitions JSON string containing configuration defintion
```

Veja a seguir um exemplo de string JSON contendo uma definição de configuração para a política de patch.

```
'{"Type":"AWSQuickSetupType-
PatchPolicy","LocalDeploymentAdministrationRoleArn":"arn:aws:iam::123456789012:role/
AWS-QuickSetup-StackSet-Local-
AdministrationRole","LocalDeploymentExecutionRoleName":"AWS-
QuickSetup-StackSet-Local-ExecutionRole","Parameters":
{"ConfigurationOptionsInstallNextInterval":"true","ConfigurationOptionsInstallValue":"cron(0
 2 ? * SAT#1
 *)","ConfigurationOptionsPatchOperation":"ScanAndInstall","ConfigurationOptionsScanNextInte
 1 * * ?
 *)","HasDeletedBaseline":"false","IsPolicyAttachAllowed":"true","OutputBucketRegion":"","Ou
east-1","PatchBaselineUseDefault":"custom","PatchPolicyName":"dev-patch-
policy","RateControlConcurrency":"5","RateControlErrorThreshold":"0%","RebootOption":"Reboot
\":{\"value\":\"arn:aws:ssm:us-east-1:123456789012:patchbaseline/
pb-0cb0c4966f86b059b\",\"label\":\"AWS-AlmaLinuxDefaultPatchBaseline\",\"description
\":\"Default Patch Baseline for Alma Linux Provided by AWS.\",\"disabled\":false},
\"AMAZON_LINUX\":{\"value\":\"arn:aws:ssm:us-east-1:123456789012:patchbaseline/
pb-0c10e657807c7a700\",\"label\":\"AWS-AmazonLinuxDefaultPatchBaseline\",
\"description\":\"Default Patch Baseline for Amazon Linux Provided by AWS.
\",\"disabled\":false},\"AMAZON_LINUX_2\":{\"value\":\"arn:aws:ssm:us-
east-1:123456789012:patchbaseline/pb-0be8c61cde3be63f3\",\"label\":\"AWS-
AmazonLinux2DefaultPatchBaseline\",\"description\":\"Baseline containing
 all Security and Bugfix updates approved for Amazon Linux 2 instances
\",\"disabled\":false},\"AMAZON_LINUX_2022\":{\"value\":\"arn:aws:ssm:us-
east-1:123456789012:patchbaseline/pb-0028ca011460d5eaf\",\"label\":\"AWS-
AmazonLinux2022DefaultPatchBaseline\",\"description\":\"Default Patch
 Baseline for Amazon Linux 2022 Provided by AWS.\",\"disabled\":false},
\"AMAZON_LINUX_2023\":{\"value\":\"arn:aws:ssm:us-east-1:123456789012:patchbaseline/
pb-05c9c9bf778d4c4d0\",\"label\":\"AWS-AmazonLinux2023DefaultPatchBaseline
\",\"description\":\"Default Patch Baseline for Amazon Linux 2023 Provided
 by AWS.\",\"disabled\":false},\"CENTOS\":{\"value\":\"arn:aws:ssm:us-
east-1:123456789012:patchbaseline/pb-03e3f588eec25344c\",\"label\":\"AWS-
CentOSDefaultPatchBaseline\",\"description\":\"Default Patch Baseline for CentOS
 Provided by AWS.\",\"disabled\":false},\"DEBIAN\":{\"value\":\"arn:aws:ssm:us-
east-1:123456789012:patchbaseline/pb-09a5f8eb62bde80b1\",\"label\":\"AWS-
DebianDefaultPatchBaseline\",\"description\":\"Default Patch Baseline for Debian
 Provided by AWS.\",\"disabled\":false},\"MACOS\":{\"value\":\"arn:aws:ssm:us-
east-1:123456789012:patchbaseline/pb-0ee4f94581368c0d4\",\"label\":\"AWS-
MacOSDefaultPatchBaseline\",\"description\":\"Default Patch Baseline for
```

```
 MacOS Provided by AWS.\",\"disabled\":false},\"ORACLE_LINUX\":{\"value\":
\"arn:aws:ssm:us-east-1:123456789012:patchbaseline/pb-06bff38e95fe85c02\",
\"label\":\"AWS-OracleLinuxDefaultPatchBaseline\",\"description\":\"Default
 Patch Baseline for Oracle Linux Server Provided by AWS.\",\"disabled\":false},
\"RASPBIAN\":{\"value\":\"arn:aws:ssm:us-east-1:123456789012:patchbaseline/
pb-0ec16280999c5c75e\",\"label\":\"AWS-RaspbianDefaultPatchBaseline\",
\"description\":\"Default Patch Baseline for Raspbian Provided by AWS.\",
\"disabled\":false},\"REDHAT_ENTERPRISE_LINUX\":{\"value\":\"arn:aws:ssm:us-
east-1:123456789012:patchbaseline/pb-0cbb3a633de00f07c\",\"label\":\"AWS-
RedHatDefaultPatchBaseline\",\"description\":\"Default Patch Baseline for Redhat
 Enterprise Linux Provided by AWS.\",\"disabled\":false},\"ROCKY_LINUX\":{\"value
\":\"arn:aws:ssm:us-east-1:123456789012:patchbaseline/pb-03ec98bc512aa3ac0\",\"label
\":\"AWS-RockyLinuxDefaultPatchBaseline\",\"description\":\"Default Patch Baseline
 for Rocky Linux Provided by AWS.\",\"disabled\":false},\"SUSE\":{\"value\":
\"arn:aws:ssm:us-east-1:123456789012:patchbaseline/pb-07d8884178197b66b\",\"label\":
\"AWS-SuseDefaultPatchBaseline\",\"description\":\"Default Patch Baseline for Suse
 Provided by AWS.\",\"disabled\":false},\"UBUNTU\":{\"value\":\"pb-06e3563bd35503f2b
\",\"label\":\"custom-UbuntuServer-Blog-Baseline\",\"description\":\"Default Patch
 Baseline for Ubuntu Provided by AWS.\",\"disabled\":false},\"WINDOWS\":{\"value
\":\"pb-016889927b2bb8542\",\"label\":\"custom-WindowsServer-Blog-Baseline\",
\"disabled\":false}}","TargetInstances":"","TargetOrganizationalUnits":"ou-9utf-
example","TargetRegions":"us-east-1,us-
east-2","TargetTagKey":"Patch","TargetTagValue":"true","TargetType":"Tags"}}' \
```

AWS CloudFormation

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
SSMQuickSetupTestConfigurationManager:
Type: "AWS::SSMQuickSetup::ConfigurationManager"
Properties:
    Name: "MyQuickSetup"
    Description: "Test configuration manager"
    ConfigurationDefinitions:
    - Type: "AWSQuickSetupType-CFGRecording"
      Parameters:
        TargetAccounts:
            Ref: AWS::AccountId
        TargetRegions:
            Ref: AWS::Region
        LocalDeploymentAdministrationRoleArn: !Sub "arn:aws:iam::
${AWS::AccountId}:role/AWS-QuickSetup-StackSet-ContractTest-AdministrationRole"
```

```
        LocalDeploymentExecutionRoleName: "AWS-QuickSetup-StackSet-ContractTest-
ExecutionRole"
    Tags:
        foo1: "bar1"
```

Tipos de configuração da Quick Setup compatíveis

Tipos de configuração compatíveis

O Quick Setup orienta você na configuração das práticas recomendadas operacionais para diversos Systems Managers e outros Serviços da AWS e implanta automaticamente essas configurações. O Quick Setup exibe uma exibição em tempo real do status de implantação da configuração.

É possível usar o Quick Setup em uma Conta da AWS individual ou em várias regiões e Contas da AWS via integração com o AWS Organizations. O uso do Quick Setup em várias contas ajuda a garantir que sua organização mantenha configurações consistentes.

A Quick Setup fornece é compatível com os tipos de configuração a seguir.

- Configurar o gerenciamento de host do Amazon EC2 usando a Quick Setup
- Definir a opção Configuração de gerenciamento de host padrão para uma organização usando a Quick Setup
- Criar um gravador de configuração do AWS Config usando o Quick Setup
- Implantar um pacote de conformidade do AWS Config usando o Quick Setup
- Configurar a aplicação de patches para instâncias em uma organização usando uma política de patch do Quick Setup
- Configuração da organização do Change Manager
- Configurar o DevOps Guru usando a Quick Setup
- Implantar pacotes do Distributor usando o Quick Setup
- Parar e iniciar instâncias do EC2 automaticamente de acordo com uma programação usando a Quick Setup
- Configuração da organização do OpsCenter
- Configurar o Explorador de recursos da AWS usando a Quick Setup

Configurar o gerenciamento de host do Amazon EC2 usando a Quick Setup

Use a Quick Setup, uma ferramenta do AWS Systems Manager, para configurar rapidamente os perfis de segurança necessários e as ferramentas do Systems Manager comumente usadas nas instâncias do Amazon Elastic Compute Cloud (Amazon EC2). Você pode usar o Quick Setup em uma conta individual ou em várias contas e Regiões da AWS ao integrar com o AWS Organizations. Essas ferramentas ajudam você a gerenciar e monitorar a integridade de suas instâncias e, ao mesmo tempo, fornecer as permissões mínimas necessárias para começar.

Se você não estiver familiarizado com os serviços e recursos do Systems Manager, recomendamos que consulte o Guia do usuário do AWS Systems Manager antes de criar uma configuração com o Quick Setup. Para obter mais informações sobre o Systems Manager, consulte O que é o AWS Systems Manager?.

Important

Talvez o Quick Setup não seja a ferramenta certa a ser usada para o gerenciamento do EC2 se uma das seguintes opções se aplicar a você:

- Você está tentando criar uma instância do EC2 pela primeira vez para testar capacidades da AWS.
- Você ainda é novo no gerenciamento de instâncias do EC2.

Em vez disso, recomendamos explorar o seguinte conteúdo:

- Conceitos básicos do Amazon EC2
- Iniciar uma instância usando o novo assistente de inicialização de instâncias no Guia do usuário do Amazon EC2
- Tutorial: conceitos básicos das instâncias do Linux do Amazon EC2 no Guia do usuário do Amazon EC2

Se você já estiver familiarizado com o gerenciamento de instâncias do EC2 e quiser simplificar a configuração e o gerenciamento de várias instâncias do EC2, use o Quick Setup. Se sua organização tiver dezenas, milhares ou milhões de instâncias do EC2, use o seguinte procedimento do Quick Setup para configurar várias opções para elas de uma só vez.

> **Note**
>
> Esse tipo de configuração permite que você defina várias opções para uma organização inteira definida em AWS Organizations, apenas para algumas contas organizacionais e regiões, ou para uma única conta. Uma dessas opções é verificar e aplicar atualizações a SSM Agent cada duas semanas. Se você for administrador da organização, também poderá optar por atualizar todas as instâncias do EC2 em sua organização com atualizações do atendente a cada duas semanas usando o tipo de configuração padrão de gerenciamento de host. Para ter mais informações, consulte Definir a opção Configuração de gerenciamento de host padrão para uma organização usando a Quick Setup.

Configurar opções de gerenciamento de host para instâncias do EC2

Para configurar o gerenciamento do host, realize as seguintes tarefas no AWS Systems Manager Quick Setup console do .

Para abrir a página de configuração de gerenciamento de hosts

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Quick Setup.
3. No cartão Gerenciamento de host, escolha Criar.

> **Tip**
>
> Se você já tem uma ou mais configurações na conta, primeiro escolha a guia Biblioteca ou o botão Criar na seção Configurações para ver os cartões.

Para configurar as opções de gerenciamento de hosts do Systems Manager

- Para configurar a funcionalidade do Systems Manager, na seção Opções de configuração, escolha as opções no grupo Systems Manager que você deseja habilitar para sua configuração:

Atualizar o agente do Systems Manager (SSM) a cada duas semanas

Permite que o Systems Manager verifique, a cada duas semanas, se há uma nova versão do agente. Se houver uma nova versão, o Systems Manager atualizará automaticamente o agente em seu nó gerenciado para a última versão lançada. A Quick Setup não instala o agente em instâncias em que ainda não esteja presente. Para obter informações sobre quais AMIs estão SSM Agent pré-instaladas, consulte Encontrar AMIs com o SSM Agent pré-instalado.

Recomendamos escolher essa opção para garantir que os nós estejam sempre executando a versão mais atualizada do SSM Agent. Para obter mais informações sobre o SSM Agent, incluindo informações sobre como instalar o agente manualmente, consulte Trabalhar com o SSM Agent.

Coletar o inventário das instâncias a cada 30 minutos

Permite que o Quick Setup configure a coleta dos seguintes tipos de metadados:

- Componentes da AWS: driver do EC2, agentes, versões e muito mais.
- Aplicações: nomes de aplicações, editores, versões e muito mais.
- Detalhes de nós: nome do sistema, nome do sistema operacional (SO), versão do SO, última inicialização, DNS, domínio, grupo de trabalho, arquitetura do SO e muito mais.
- Configuração de rede:: endereço IP, endereço MAC, DNS, gateway, máscara de sub-rede e muito mais.
- Serviços: nome, nome de exibição, status, serviços dependentes, tipo de serviço, tipo de início e muito mais (somente instâncias do Windows Server).
- Funções do Windows: nome, nome de exibição, caminho, tipo de recurso, estado instalado e muito mais (somente nós do Windows Server).
- Atualizações do Windows: ID do hotfix, autor da instalação, data da instalação e muito mais (somente nós do Windows Server).

Para obter mais informações sobre o Inventory, uma ferramenta do AWS Systems Manager, consulte AWS Systems Manager Inventory.

> **Note**
>
> A opção Inventory collection (Coleta de inventário) pode demorar até 10 minutos para ser concluída, mesmo que você tenha selecionado apenas alguns nós.

Verificar patches ausentes nas instâncias diariamente

Habilita o Patch Manager, uma ferramenta do Systems Manager, que permite verificar seus nós diretamente e gerar um relatório na página Conformidade. O relatório mostra quantos nós são compatíveis com os patches de acordo com a lista padrão de referência de patches. O relatório inclui uma lista de cada nó e seu status de conformidade.

Para obter mais informações sobre operações e listas de referência de patches, consulte AWS Systems Manager Patch Manager.

Para obter informações sobre conformidade de patches, consulte a página Compliance (Conformidade) do Systems Manager.

Para obter informações sobre como aplicar patches em nós gerenciados em várias contas e regiões em uma configuração, consulte Configurações de políticas de patches em Quick Setup e Configurar a aplicação de patches para instâncias em uma organização usando uma política de patch do Quick Setup.

> **Important**
>
> O Systems Manager oferece suporte a vários métodos de verificação de nós gerenciados para conformidade de patches. Se você implementar mais de um desses métodos ao mesmo tempo, as informações de conformidade de patch que você vir serão sempre o resultado da verificação mais recente. Os resultados de verificações anteriores serão sobrescritos. Se os métodos de verificação usarem listas de referência de patches diferentes, com regras de aprovação diferentes, as informações de conformidade do patch poderão mudar inesperadamente. Para ter mais informações, consulte Prevenção de substituições não intencionais de dados de conformidade de patches.

Para configurar as opções de gerenciamento de host do Amazon CloudWatch

- Para configurar a funcionalidade do CloudWatch, na seção Opções de configuração, escolha as opções no grupo Amazon CloudWatch que você deseja habilitar para sua configuração:

  Baixar e configurar o atendente do CloudWatch

  Instala a configuração básica do agente unificado do CloudWatch em suas instâncias do Amazon EC2. O agente do coleta métricas e arquivos de log de suas instâncias do Amazon CloudWatch. Essas informações são consolidadas para que você possa determinar rapidamente a integridade de suas instâncias. Para obter mais informações sobre a configuração básica do agente do CloudWatch, consulte Conjuntos de métricas predefinidas do agente CloudWatch. Pode haver custo adicional. Para obter mais informações, consulte Preço do Amazon CloudWatch.

  Atualizar o agente do CloudWatch uma vez a cada 30 dias

  Permite que o Systems Manager verifique a cada 30 dias se há uma nova versão do agente do CloudWatch. Se houver uma nova versão, o Systems Manager atualizará o agente na sua instância. Recomendamos escolher essa opção para garantir que as instâncias estejam sempre executando a versão mais atualizada do agente do CloudWatch.

Para configurar as opções de gerenciamento de host do Amazon EC2 Launch Agent

- Para configurar a funcionalidade do Amazon EC2 Launch Agent, na seção Opções de configuração, escolha as opções no grupo Amazon EC2 Launch Agent que você deseja habilitar para sua configuração:

  Atualizar o agente de execução do EC2 uma vez a cada 30 dias

  Permite que o Systems Manager verifique a cada 30 dias se há uma nova versão do agente de execução instalada na instância. Se uma nova versão estiver disponível, o Systems Manager atualizará o agente na instância. Recomendamos escolher essa opção para garantir que as instâncias estejam sempre executando a versão mais atualizada do agente de execução aplicável. Em instâncias do Windows do Amazon EC2, essa opção é compatível com EC2Launch, EC2Launch v2 e EC2Config. Em instâncias do Linux do Amazon EC2, essa

opção é compatível com `cloud-init`. Em instâncias Mac do Amazon EC2, essa opção é compatível com `ec2-macos-init`. A Quick Setup não oferece suporte à atualização de agentes de execução instalados em sistemas operacionais não compatíveis com o agente de execução ou no AL2023.

Para obter mais informações sobre esses agentes de inicialização, consulte os seguintes tópicos:

- Configurar uma instância do Windows usando o EC2Launch v2
- Configurar uma instância do Windows usando o EC2Launch
- Configurar uma instância do Windows usando o serviço EC2Config
- Documentação do cloud-init
- ec2-macos-init

Para selecionar as instâncias do EC2 a serem atualizadas pela configuração de gerenciamento de host

- Na seção Destinos escolha o método para determinar as contas e regiões nas quais a configuração será implantada:

> **Note**
>
> Não é possível criar várias configurações de gerenciamento de host do Quick Setup que tenham como destino a mesma Região da AWS.

Entire organization

Sua configuração é implantada em todas as unidades organizacionais (OUs) e nas Regiões da AWS em sua organização.

> **Note**
>
> A opção Organização inteira só estará disponível se você estiver configurando o gerenciamento de hosts a partir da conta de gerenciamento da organização.

Custom

1. Na seção UOs de destino, selecione as UOs nas quais você deseja implantar essa configuração de gerenciamento de host.
2. Na seção Regiões de destino, selecione as regiões nas quais você deseja implantar essa configuração de gerenciamento de host.

Current account

Escolha uma das opções de região e siga as etapas aplicáveis a essa opção.

Região atual

Escolha como definir instâncias como destino somente na região atual:

- Todas as instâncias: a configuração de gerenciamento de host define automaticamente como alvo cada EC2 na região atual.
- Tag: escolha Adicionar e insira a chave e o valor opcional que são adicionados às instâncias a serem segmentadas.
- Grupo de recursos: em Grupo de recursos, selecione um grupo de recursos existente que contenha as instâncias do EC2 que serão definidas como destino.
- Manual: na seção Instâncias, marque a caixa de seleção de cada instância do EC2 que será definida como destino.

Escolha Regiões

Escolha como definir como destino instâncias na região especificada escolhendo uma das seguintes opções:

- Todas as instâncias: todas as instâncias nas regiões que você especificar são definidas como destino.
- Tag: escolha Adicionar e insira a chave e o valor opcional que foram adicionados às instâncias a serem segmentadas.

Na seção Regiões de destino, selecione as regiões nas quais você deseja implantar essa configuração de gerenciamento de host.

Para especificar uma opção de perfil de instância

- Somente destinos Toda a organização e Personalizados.

  Na seção Opções de perfil da instância, escolha se deseja adicionar as políticas obrigatórias do IAM aos perfis existentes anexados às suas instâncias ou permitir que o Quick Setup crie as políticas do IAM e os perfis de instância com as permissões necessárias para a configuração escolhida.

Após especificar todas as opções de configuração, escolha Criar.

Definir a opção Configuração de gerenciamento de host padrão para uma organização usando a Quick Setup

Com a Quick Setup, uma ferramenta do AWS Systems Manager, é possível ativar a Configuração de gerenciamento de host padrão para todas as contas e regiões que foram adicionadas à sua organização no AWS Organizations. Isso garante que você SSM Agent esteja atualizado em todas as instâncias do Amazon Elastic Compute Cloud (EC2) na organização e que elas possam se conectar ao Systems Manager.

Antes de começar

Verifique se os seguintes requisitos foram atendidos antes de ativar essa configuração.

- A versão mais recente do já SSM Agent está instalada em todas as instâncias do EC2 a serem gerenciadas em sua organização.
- As instâncias do EC2 que você deseja gerenciar estão usando o Serviço de Metadados de Instância Versão 2 (IMDSv2).
- Você está conectado à conta de gerenciamento da sua organização, conforme especificado em AWS Organizations, usando uma identidade AWS Identity and Access Management (IAM) (usuário, função ou grupo) com permissões de administrador.

Usando a função padrão de gerenciamento de instâncias do EC2

A configuração padrão de gerenciamento de host usa a configuração `default-ec2-instance-management-role` de serviço do Systems Manager. Essa é uma função com permissões que você deseja disponibilizar para todas as contas em sua organização para permitir a comunicação entre SSM Agent a instância e o serviço Systems Manager na nuvem.

Se você já definiu essa função usando o comando CLI update-service-setting, a Configuração Padrão de Gerenciamento de Host usa essa função. Se você ainda não definiu essa função, Quick Setup criará e aplicará a função para você.

Para verificar se essa função já foi especificada para sua organização, use o comando get-service-setting.

Ative atualizações automáticas SSM Agent a cada duas semanas

Use o procedimento a seguir para ativar a opção Configuração de gerenciamento de host padrão para toda a sua AWS Organizations organização.

Para ativar atualizações automáticas de SSM Agent a cada duas semanas

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Quick Setup.
3. No cartão Configuração de gerenciamento de host padrão, escolha Criar.

> **Tip**
>
> Se você já tem uma ou mais configurações na conta, primeiro escolha a guia Biblioteca ou o botão Criar na seção Configurações para ver os cartões.

4. Na seção Opções de configuração, selecione Ativar atualizações automáticas de SSM Agent a cada duas semanas.
5. Selecione Criar

Criar um gravador de configuração do AWS Config usando o Quick Setup

Com Quick Setup, uma ferramenta do AWS Systems Manager, é possível criar rapidamente um gravador de configuração com o AWS Config. Usa o gravador de configurações para detectar alterações nas configurações do seu recurso e capturar as alterações como itens de configuração. Se você não estiver familiarizado com AWS Config, recomendamos saber mais sobre o serviço revisando o conteúdo no AWS Config Guia do desenvolvedor antes de criar uma configuração com o Quick Setup. Para obter mais informações sobre o AWS Config, consulte O que é o AWS Config? no Guia do desenvolvedor do AWS Config.

Por padrão, o gravador de configurações registra todos os recursos com suporte na Região da AWS em que o AWS Config estiver em execução. Você pode personalizar a configuração para que

apenas os tipos de recursos especificados sejam registrados. Para obter mais informações, consulte Selecionar quais recursos a AWS Config grava no AWS Config Guia do desenvolvedor.

Você será cobrado pelo uso do serviço quando o AWS Config começar a registrar as configurações. Para obter informações sobre preços, consulte AWS Config preços.

> **Note**
>
> Se você já criou um gravador de configuração, a Quick Setup não interrompe a gravação nem faz alterações nos tipos de recursos que já estão sendo gravados. Se você optar por registrar outros tipos de recursos usando a Quick Setup, o serviço os anexará aos grupos de gravadores atuais. A exclusão da configuração Config recording (Gravação do Config) do Quick Setup não interrompe o gravador de configurações. As alterações continuam a ser registradas e as taxas de uso do serviço são aplicadas até que você interrompa o gravador de configurações. Para saber mais sobre como gerenciar o gravador de configurações, consulte Gerenciar o gravador de configurações no Guia do desenvolvedor do AWS Config.

Para configurar a AWS Config, realize as seguintes tarefas no console do AWS Systems Manager.

Para configurar a gravação do AWS Config com o Quick Setup

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Quick Setup.
3. No cartão Gravação de configuração, escolha Criar.

   > **Tip**
   >
   > Se você já tem uma ou mais configurações na conta, primeiro escolha a guia Biblioteca ou o botão Criar na seção Configurações para ver os cartões.

4. Na seção Opções de configuração, faça o seguinte:

   a. Em Escolher os tipos de recursos da AWS a serem registrados, especifique se deseja registrar todos os recursos suportados ou somente os tipos de recursos que você escolher.

   b. Para Configurações de entrega, especifique se deseja criar um novo bucket do Amazon Simple Storage Service (Amazon S3) ou escolha um bucket existente para o qual deseja enviar snapshots da configuração.

c. Para Opções de notificação, selecione a opção de notificação desejada. O AWS Config usa o Amazon Simple Notification Service (Amazon SNS) para notificar você sobre importantes eventos do AWS Config relacionados aos recursos. Se escolher a opção Usar tópicos do SNS existentes, você deverá fornecer o ID da Conta da AWS e o nome do tópico existente do Amazon SNS na conta que você deseja usar. No Opções de configuração, selecione a Regiões da AWS que você deseja que o registre e se você deseja incluir recursos globais.

5. Na seção Schedule (Programação), escolha a frequência com que o Quick Setup deverá corrigir as alterações feitas nos recursos que diferem da sua configuração. A opção Default (Padrão) é executada uma vez. Se você não quiser que o Quick Setup corrija as alterações feitas em recursos que diferem da sua configuração, escolha Desabilitar a correção em Personalizado.
6. Na seção Destinos, escolha uma das opções a seguir para identificar as contas e regiões para gravação.

> **Note**
>
> Se você estiver trabalhando em uma única conta, as opções para trabalhar com organizações e unidades organizacionais (UOs) não estão disponíveis. É possível escolher se deseja aplicar essa configuração a todas as Regiões da AWS em sua conta ou somente às regiões que você escolher.

- Entire organization (Organização inteira): todas as contas e regiões da sua organização.
- Custom (Personalizado): somente as UOs e regiões que você especificar.
  - Na seção UOs de destino, selecione as UOs nas quais você deseja usar a gravação.
  - Na seção Regiões de destino, selecione as regiões nas quais você deseja usar a gravação.
- Current account (Conta atual): somente as regiões especificadas na conta em que você está conectado atualmente são visadas. Escolha uma das seguintes opções:
  - Current Region (Região atual): somente os nós gerenciados na região selecionada no console são visados.
  - Escolher regiões: escolha as regiões individuais nas quais aplicar a configuração de gravação.

7. Escolha Criar.

Implantar um pacote de conformidade do AWS Config usando o Quick Setup

Um pacote de conformidade é uma coleção de regras e ações de correção da AWS Config. Com Quick Setup, é possível implantar um pacote de conformidade como uma única entidade em uma conta e um Região da AWS ou através de uma organização no AWS Organizations. Isso ajuda você a gerenciar a conformidade com a configuração de seu recursos da AWS em grande escala, desde a definição de políticas até a auditoria e a geração de relatórios agregados, utilizando um framework comum e um modelo de empacotamento.

Para implantar pacotes de conformidade, execute as tarefas a seguir no console do Quick Setup do AWS Systems Manager.

> **Note**
>
> Você deve habilitar a gravação do AWS Config antes de implantar essa configuração. Para obter mais informações, consulte Pacotes de conformidade no Manual do desenvolvedor do AWS Config.

Para implantar pacotes de conformidade com o Quick Setup

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Quick Setup.
3. No cartão Pacotes de conformidade, escolha Criar.

   > **Tip**
   >
   > Se você já tem uma ou mais configurações na conta, primeiro escolha a guia Biblioteca ou o botão Criar na seção Configurações para ver os cartões.

4. Na seção Escolher pacotes de conformidade, escolha os pacotes de conformidade que deseja implantar.
5. Na seção Schedule (Programação), escolha a frequência com que o Quick Setup deverá corrigir as alterações feitas nos recursos que diferem da sua configuração. A opção Default (Padrão) é executada uma vez. Se não quiser que o Quick Setup corrija as alterações feitas nos recursos diferentes de sua configuração, escolha Disabled (Desabilitado) em Custom (Personalizar).
6. No Destinos, escolha se deseja implantar pacotes de conformidade em toda a sua organização, alguns Regiões da AWS ou na conta na qual você está conectado no momento.

Se escolher Entire organization (Toda a organização), continue na etapa 8.

Se escolher Custom (Personalizado), continue na etapa 7.

7. No Regiões de destino, marque as caixas de seleção das Regiões nas quais você deseja implantar pacotes de conformidade.
8. Escolha Criar.

Configurar a aplicação de patches para instâncias em uma organização usando uma política de patch do Quick Setup

Com o Quick Setup, uma ferramenta do AWS Systems Manager, é possível criar políticas de patch baseadas no Patch Manager. Uma política de patch define a programação e a lista de referência a serem usadas ao aplicar patches automaticamente nas suas instâncias do Amazon Elastic Compute Cloud (Amazon EC2) e em outros nós gerenciados. Usando uma única configuração de política de patch, é possível definir a aplicação de patches para todas as contas em várias Regiões da AWS da sua organização, somente para as contas e regiões que você escolher ou para um único par de conta-região. Para obter mais informações sobre políticas de patch, consulte Configurações de políticas de patches em Quick Setup.

Pré-requisito

Para definir uma política de patch para um nó usando o Quick Setup, o nó deve ser um nó gerenciado. Para obter mais informações sobre o gerenciamento de seus nós, consulte Configurar o console unificado do Systems Manager para uma organização.

> **Important**
>
> Métodos de verificação da conformidade de patches: o Systems Manager oferece suporte a vários métodos de verificação de nós gerenciados para conformidade de patches. Se você implementar mais de um desses métodos ao mesmo tempo, as informações de conformidade de patch que você vir serão sempre o resultado da verificação mais recente. Os resultados de verificações anteriores serão sobrescritos. Se os métodos de verificação usarem listas de referência de patches diferentes, com regras de aprovação diferentes, as informações de conformidade do patch poderão mudar inesperadamente. Para obter mais informações, consulte Prevenção de substituições não intencionais de dados de conformidade de patches.

Status de conformidade da associação e políticas de patch: o status de patch de um nó gerenciado que está sob uma política de patch de Quick Setup corresponde ao status de execução da associação do State Manager para esse nó. Se o status de execução da associação for `Compliant`, o status de patch do nó gerenciado também será marcado como `Compliant`. Se o status de execução da associação for `Non-Compliant`, o status de patch do nó gerenciado também será marcado como `Non-Compliant`.

Regiões com suporte para configurações de políticas de patch

No momento, as configurações de política de patch do Quick Setup são compatíveis nas seguintes regiões:

- Leste dos EUA (Ohio) (us-east-2)
- Leste dos EUA (Norte da Virgínia) (us-east-1)
- Oeste dos EUA (Norte da Califórnia) (us-west-1)
- Oeste dos EUA (Oregon) (us-west-2)
- Ásia-Pacífico (Mumbai) (ap-south-1)
- Ásia-Pacífico (Seul) (ap-northeast-2)
- Ásia-Pacífico (Singapura) (ap-southeast-1)
- Ásia-Pacífico (Sydney) (ap-southeast-2)
- Ásia Pacific (Tóquio) (ap-northeast-1)
- Canadá (Central) (ca-central-1)
- Europa (Frankfurt) (eu-central-1)
- Europa (Irlanda) (eu-west-1)
- Europa (Londres) (eu-west-2)
- Europa (Paris) (eu-west-3)
- UE (Estocolmo) (eu-north-1)
- América do Sul (São Paulo) (sa-east-1)

Permissões para o bucket do S3 da política de patch

Quando você cria uma política de patch, a Quick Setup cria um bucket do Amazon S3 que contém um arquivo chamado `baseline_overrides.json`. Esse arquivo armazena informações sobre as listas de referência de patches que você especificou para a política de patch.

O nome do bucket do S3 está no formato `aws-quicksetup-patchpolicy-`*`account-id`*`-`*`quick-setup-configuration-id`*.

Por exemplo: `aws-quicksetup-patchpolicy-123456789012-abcde`

Se você estiver criando uma política de patch para uma organização, o bucket será criado na conta de gerenciamento da organização.

Há dois casos de uso em que é necessário fornecer permissão a outros recursos da AWS para acessar esse bucket do S3 usando políticas do AWS Identity and Access Management (IAM):

- Caso 1: use seu próprio perfil de instância ou perfil de serviço com seus nós gerenciados em vez de um fornecido pela Quick Setup
- Caso 2: use endpoints da VPC para se conectar ao Systems Manager

A política de permissões necessária em ambos os casos está localizada na seção abaixo, Permissões de política para buckets do S3 da Quick Setup.

Caso 1: use seu próprio perfil de instância ou perfil de serviço com seus nós gerenciados em vez de um fornecido pela Quick Setup

As configurações de políticas de patch incluem a opção Adicionar as políticas do IAM necessárias aos perfis de instância existentes anexados às suas instâncias.

Se você não escolher essa opção, mas quiser que a Quick Setup aplique patches em seus nós gerenciados usando essa política de patch, é necessário garantir que o seguinte seja implementado:

- A política gerenciada do IAM `AmazonSSMManagedInstanceCore` deve ser anexada ao perfil de instância do IAM ou ao perfil de serviço do IAM que é usado para fornecer permissões do Systems Manager aos nós gerenciados.
- É necessário adicionar permissões para acessar seu bucket de política de patch como uma política em linha no perfil de instância do IAM ou perfil de serviço do IAM. Você pode fornecer acesso curinga a todos os buckets `aws-quicksetup-patchpolicy` ou somente ao bucket específico criado para sua organização ou conta, conforme mostrado nos exemplos de código anteriores.

- É necessário marcar seu perfil de instância do IAM ou perfil de serviço do IAM com o par de chave-valor a seguir.

  `Key: QSConfigId-`*`quick-setup-configuration-id`*`, Value: `*`quick-setup-configuration-id`*

  `quick-setup-configuration-id` representa o valor do parâmetro aplicado à pilha do AWS CloudFormation que é usada ao criar a configuração da política de patch. Para recuperar o ID, faça o seguinte:

  1. Abra o console do AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
  2. Selecione o nome da pilha usada para criar a política de patch. O nome está em um formato como `StackSet-AWS-QuickSetup-PatchPolicy-LA-q4bkg-52cd2f06-d0f9-499e-9818-d887cEXAMPLE`.
  3. Selecione a guia Parâmetros.
  4. Na lista Parâmetros, na coluna Chave, localize a chave QSConfigurationId. Na coluna Valor da respectiva linha, localize o ID de configuração, como `abcde`.

     Neste exemplo, para que a etiqueta seja aplicada ao perfil de instância ou perfil de serviço, a chave é `QSConfigId-abcde`, e o valor é `abcde`.

Para obter informações sobre como adicionar etiquetas a um perfil do IAM, consulte Etiquetar perfis do IAM e Gerenciar tags em perfis de instância (AWS CLI ou AWS API) no Guia do usuário do IAM.

Caso 2: use endpoints da VPC para se conectar ao Systems Manager

Se você usa endpoints da VPC para se conectar ao Systems Manager, sua política de endpoint da VPC para o S3 deve permitir acesso ao bucket do S3 de sua política de patch da Quick Setup.

Para obter informações sobre a adição de permissões a uma política de endpoint da VPC para o S3, consulte Controlar o acesso a partir de endpoints da VPC com políticas de bucket no Guia do usuário do Amazon S3.

Permissões de política para buckets do S3 da Quick Setup

Você pode fornecer acesso curinga a todos os buckets `aws-quicksetup-patchpolicy` ou somente ao bucket específico criado para sua organização ou conta. Para fornecer as permissões necessárias nos dois casos descritos abaixo, use qualquer um dos formatos.

All patch policy buckets

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AccessToAllPatchPolicyRelatedBuckets",
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::aws-quicksetup-patchpolicy-*"
    }
  ]
}
```

Specific patch policy bucket

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AccessToMyPatchPolicyRelatedBucket",
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::aws-quicksetup-patchpolicy-111122223333-quick-
setup-configuration-id"
    }
  ]
}
```

[1]Depois que a configuração da política de patch for criada, você localizará o nome completo do bucket no console do S3. Por exemplo: `aws-quicksetup-patchpolicy-123456789012-abcde`

IDs aleatórios da lista de referência de patches em operações de política de patches

As operações de aplicações de patch para políticas de patch utilizam o parâmetro `BaselineOverride` no documento do SSM Command `AWS-RunPatchBaseline`.

Ao usar `AWS-RunPatchBaseline` para aplicar patches fora de uma política de patch, você pode usar `BaselineOverride` para especificar uma lista de referência de patches a ser usada durante a operação que são diferentes dos padrões especificados. Você cria essa lista em um arquivo chamado `baseline_overrides.json` e a adiciona manualmente a um bucket do Amazon S3 de sua propriedade, conforme explicado em Usar o parâmetro BaselineOverride.

No entanto, para operações de aplicações de patch com base em políticas de patch, o Systems Manager cria um bucket do S3 automaticamente e adiciona um arquivo `baseline_overrides.json` a ele. Então, toda vez que a Quick Setup executa uma operação de aplicação de patches (usando a ferramenta Run Command), o sistema gera um ID aleatório para cada lista de referência de patches. Esse ID é diferente para cada operação de aplicação de patches da política de patch, e a lista de referência de patches que ela representa não está armazenada nem acessível a você em sua conta.

Como resultado, você não verá o ID da lista de referência de patches selecionada em sua configuração nos logs de patches. Isso se aplica tanto às listas de referência de patches gerenciadas pela AWS como às listas de referência de patches personalizadas que você possa ter selecionado. Em vez disso, o ID da lista de referência relatado no log é aquele que foi gerado para a operação de aplicação de patches específica.

Além disso, se você tentar visualizar no Patch Manager detalhes sobre uma lista de referência de patches que foi gerada com um ID aleatório, o sistema informará que a lista de referência de patches não existe. Esse comportamento é esperado e pode ser ignorado.

Criação de uma política de patch

Para criar uma política de patch, execute as tarefas a seguir no console do Systems Manager.

Para criar uma política de patch com o Quick Setup

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.

   Se estiver configurando a aplicação de patches para uma organização, certifique-se de estar conectado à conta de gerenciamento da organização. Você não pode configurar a política usando a conta de administrador delegado ou uma conta de membro.

2. No painel de navegação, escolha Quick Setup.
3. No cartão do Patch Manager, escolha Create (Criar).

> **Tip**
>
> Se você já tem uma ou mais configurações na conta, primeiro escolha a guia Biblioteca ou o botão Criar na seção Configurações para ver os cartões.

4. Em Configuration name (Nome da configuração), insira um nome para ajudar a identificar a política de patch.
5. Na seção Scanning and installation (Verificação e instalação), em Patch operation (Operação de patch), escolha se a política de patch fará Scan (Verificação) nos destinos especificados ou se fará Scan and install (Verificação e instalação) de patches em destinos especificados.
6. Em Scanning schedule (Programação de verificação), escolha Use recommended defaults (Usar padrões recomendados) ou Custom scan schedule (Programação de verificação personalizada). A programação de verificação padrão examinará seus destinos diariamente à 1h UTC.

   - Se você escolher Custom scan schedule (Programação de verificação personalizada), selecione a Scanning frequency (Frequência da verificação).
   - Se você escolher Daily (Diariamente), insira a hora, em UTC, em que deseja verificar seus destinos.
   - Se você escolher Custom CRON expression (Expressão do CRON personalizada), insira a programação como uma CRON expression (Expressão do CRON). Para obter mais informações sobre a formatação das expressões do CRON para o Systems Manager, consulte Referência: Expressões cron e rate para o Systems Manager.

     Além disso, selecione Wait to scan targets until first CRON interval (Aguardar para verificar os destinos até o primeiro intervalo do CRON). Por padrão, o Patch Manager varre imediatamente os nós à medida que eles se tornam destinos.
7. Se você escolher Scan and install (Verificar e instalar), escolha a Installation schedule (Programação de instalação) a ser usada ao instalar patches nos destinos especificados. Se você escolher Use recommended defaults (Usar padrões recomendados), o Patch Manager instalará os patches semanais às 2:00 UTC de domingo.

   - Se você escolher Custom install schedule (Programação de instalação personalizada), selecione a Installation frequency (Frequência da instalação).

- Se você escolher Daily (Diariamente), insira a hora, em UTC, em que deseja instalar as atualizações nos seus destinos.
- Se você escolher Custom CRON expression (Expressão do CRON personalizada), insira a programação como uma CRON expression (Expressão do CRON). Para obter mais informações sobre a formatação das expressões do CRON para o Systems Manager, consulte Referência: Expressões cron e rate para o Systems Manager.

  Além disso, desmarque Wait to install updates until first CRON interval (Esperar para instalar as atualizações até o primeiro intervalo do CRON) para instalar imediatamente as atualizações nos nós quando eles se tornarem destinos. Por padrão, o Patch Manager aguarda até o primeiro intervalo do CRON para instalar as atualizações.
- Escolha Reboot if needed (Reinicializar, se necessário), para reinicializar os nós após a instalação dos patches. A reinicialização após a instalação é recomendada, mas pode causar problemas de disponibilidade.

8. Na seção lista de referência de patches, escolha as lista de referência de patches a serem usadas ao verificar e atualizar seus destinos.

   Por padrão, o Patch Manager usa as listas de referência de patches predefinidas. Para obter mais informações, consulte Linhas de base predefinidas.

   Se você escolher Custom patch baseline (Lista de referência de patches personalizada), altere a lista de referência de patches para sistemas operacionais predefinida da AWS que você não deseja usar.

   > **Note**
   >
   > Se você usa endpoints da VPC para se conectar ao Systems Manager, verifique se a política de endpoint da VPC para o S3 permite acesso ao bucket do S3. Para obter mais informações, consulte Permissões para o bucket do S3 da política de patch.

   > **Important**
   >
   > Se você estiver usando uma configuração de política de patch em Quick Setup, as atualizações feitas nas listas de referência de patches personalizadas serão sincronizadas com Quick Setup uma vez por hora.

Se uma lista de referência de patches personalizada que foi referenciada em uma política de patch for excluída, um banner será exibido na página Configuration details (Detalhes da configuração) do Quick Setup da sua política de patch. O banner informa que a política de patch faz referência a uma lista de referência de patches que não existe mais e que as operações de aplicação de patches subsequentes falharão. Nesse caso, retorne à página Configurations (Configurações) do Quick Setup, selecione a configuração Patch Manager e escolha Actions (Ações), Edit configuration (Editar configuração). O nome da lista de referência de patches excluída será destacado, e você deverá selecionar uma nova lista de referência de patches para o sistema operacional afetado.

9. (Opcional) Na seção Patching log storage (Armazenamento de logs de patches), selecione Write output to S3 bucket (Gravar saída no bucket do S3) para armazenar os logs da operação de aplicação de patches um bucket do Amazon S3.

> **Note**
>
> Se você estiver configurando uma política de patch para uma organização, a conta de gerenciamento da sua organização deverá ter pelo menos permissões somente de leitura para esse bucket. Todas as unidades organizacionais incluídas na política devem ter acesso de gravação ao bucket. Para obter informações sobre como conceder acesso a diferentes contas, consulte o Exemplo 2: Concessão de permissões de bucket entre contas pelo proprietário do bucket no Guia do usuário do Amazon Simple Storage Service.

10. Escolha Procurar no S3 para selecionar o bucket no qual você deseja armazenar a saída de log dos patches. A conta de gerenciamento deve ter acesso de leitura para esse bucket. Todas as contas e destinos não gerenciais configurados na seção Targets (Destinos) devem ter acesso de gravação ao bucket S3 fornecido para os logs.
11. Na seção Targets (Destinos), escolha uma das opções a seguir para identificar as contas e regiões dessa operação de política de patch.

> **Note**
>
> Se você estiver trabalhando em uma única conta, as opções para trabalhar com organizações e unidades organizacionais (UOs) não estão disponíveis. É possível

escolher se deseja aplicar essa configuração a todas as Regiões da AWS em sua conta ou somente às regiões que você escolher.
Se você especificou anteriormente uma região de origem para sua conta e não se integrou à nova experiência do console de Quick Setup, não poderá excluir essa região da configuração de Destinos.

- Entire organization (Organização inteira): todas as contas e regiões da sua organização.
- Custom (Personalizado): somente as UOs e regiões que você especificar.
  - Na seção Target OUs (UOs de destino), selecione as UOs nas quais você deseja configurar a política de patch.
  - Na seção Target Regions (Regiões de destino), selecione as regiões nas quais você deseja aplicar a política de patch.
- Current account (Conta atual): somente as regiões especificadas na conta em que você está conectado atualmente são visadas. Escolha uma das seguintes opções:
  - Current Region (Região atual): somente os nós gerenciados na região selecionada no console são visados.
  - Choose Regions (Escolher regiões): escolha as regiões individuais nas quais aplicar a política de patch.

12. Em Choose how you want to target instances (Escolher como deseja visar as instâncias), escolha uma das opções a seguir para identificar os nós nos quais aplicar os patches:

- All managed nodes (Todos os nós gerenciados): todos os nós gerenciados nas UOs e regiões selecionadas.
- Specify the resource group (Especificar o grupo de recursos): escolha o nome de um grupo de recursos na lista para visar seus recursos associados.

> **Note**
>
> Atualmente, há suporte para a seleção de grupos de recursos somente em configurações de conta única. Para aplicar patches em recursos em várias contas, escolha uma opção de visar diferente.

- Specify a node tag (Especificar uma tag de nó): somente os nós marcados com o par de valores-chave que você especificar terão patches aplicados em todas as contas e regiões visadas.

- Manual: escolhe nós gerenciados de todas as contas e regiões especificadas manualmente em uma lista.

> **Note**
>
> No momento, há suporte apenas para instâncias do Amazon EC2 com essa opção. Você pode adicionar no máximo 25 instâncias manualmente em uma configuração de política de patch.

13. Na seção Rate control (Controle de taxa), faça o seguinte:

    - Em Concurrency (Concorrência), insira um número ou uma porcentagem de nós nos quais executar a política de patch ao mesmo tempo.
    - Em Error threshold (Limite de erro), insira o número ou a porcentagem de nós que podem apresentar um erro antes que a política de patch falhe.

14. (Opcional) Marque a caixa de seleção Adicionar políticas do IAM necessárias aos perfis de instância existentes anexados às suas instâncias.

    Essa seleção aplica as políticas do IAM criadas por essa configuração da Quick Setup aos nós que já têm um perfil de instância ou um perfil de serviço anexado (instâncias do EC2) ou um perfil de serviço anexados (nós ativados para ambientes híbridos). Recomendamos essa seleção quando seus nós gerenciados já tiverem um perfil de instância ou um perfil de serviço anexado, mas ele não contiver todas as permissões necessárias para trabalhar com o Systems Manager.

    Sua seleção aqui é aplicada aos nós gerenciados criados posteriormente nas contas e regiões às quais essa configuração de política de patch se aplica.

> **Important**
>
> Se você não marcou essa opção, mas quiser que a Quick Setup aplique patches em seus nós gerenciados usando essa política de patch, é necessário fazer o seguinte: Adicione permissões ao perfil de instância do IAM ou perfil de serviço do IAM para acessar o bucket do S3 criado para sua política de patch
> Marque seu perfil de instância do IAM ou perfil de serviço do IAM com um par de chave-valor específico.

Para mais informações, consulte Caso 1: use seu próprio perfil de instância ou perfil de serviço com seus nós gerenciados em vez de um fornecido pela Quick Setup.

15. Escolha Criar.

    Para revisar o status da aplicação de patches após a criação da política de patch, é possível acessar a configuração na página Quick Setup.

Configurar o DevOps Guru usando a Quick Setup

Você pode configurar rapidamente as opções do DevOps Guru usando Quick Setup. O Amazon DevOps Guru é um serviço alimentado por machine Learning (ML) que facilita o aprimoramento da performance operacional e da disponibilidade de uma aplicação. O DevOps Guru detecta comportamentos diferentes dos padrões operacionais normais para que você possa identificar problemas operacionais bem antes deles afetarem seus clientes. O DevOps Guru ingere automaticamente os dados operacionais das aplicações da AWS e fornece um único painel para visualizar problemas nos dados operacionais. Você pode começar a usar o DevOps Guru para melhorar a disponibilidade e a confiabilidade das aplicações, sem experiência em configuração manual ou machine learning.

A configuração do DevOps Guru com o Quick Setup está disponível nas seguintes Regiões da AWS:

- Leste dos EUA (Norte da Virgínia)
- Leste dos EUA (Ohio)
- Oeste dos EUA (Oregon)
- Europa (Frankfurt)
- Europa (Irlanda)
- Europa (Estocolmo)
- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Tóquio)

Para obter informações sobre preços, consulte Preços do Amazon DevOps Guru.

Para configurar o DevOps Guru, realize as tarefas a seguir no console do Quick Setup do AWS Systems Manager.

Para configurar o DevOps Guru com Quick Setup

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Quick Setup.
3. No cartão DevOps Guru, escolha Criar.

> **Tip**
>
> Se você já tem uma ou mais configurações na conta, primeiro escolha a guia Biblioteca ou o botão Criar na seção Configurações para ver os cartões.

4. Na seção Configuration options (Opções de configuração), selecione os tipos de recursos da AWS que você deseja analisar e suas preferências de notificação.

   Se você não selecionar a opção Analyze all AWS resources in all the accounts in my organization (Analisar todos os recursos da em todas as contas da minha organização), você poderá escolher os recursos da AWS para analisar posteriormente no console do DevOps Guru. O DevOps Guru analisa diferentes tipos de recursos da AWS (como buckets do Amazon Simple Storage Service (Amazon S3) e instâncias do Amazon Elastic Compute Cloud (Amazon EC2), que são categorizados em dois grupos de preços. Você paga por horas de recursos da AWS analisadas, para cada recurso ativo. Um recurso só estará ativo se produzir métricas, eventos ou entradas de log em uma hora. A tarifa cobrada por um tipo de recurso da AWS específico depende do grupo de preços.

   Se você selecionar a opção Enable SNS notifications (Habilitar notificações do SNS), um tópico do Amazon Simple Notification Service (Amazon SNS) é criado em cada Conta da AWS nas unidades organizacionais (UOs) que você determina como destino em sua configuração. O DevOps Guru usa o tópico para notificar você sobre eventos importantes do DevOps Guru, como a criação de um novo insight. Se você não habilitar essa opção, poderá adicionar um tópico posteriormente no console DevOps Guru.

   Se você selecionar a opção Enable AWS Systems Manager OpsItems (Habilitar Opsitems do Systems Manager), os items de trabalho operacionais (OpsItems) serão criados para eventos do Amazon EventBridge relacionados e para alarmes do Amazon CloudWatch.
5. Na seção Schedule (Programação), escolha a frequência com que o Quick Setup deverá corrigir as alterações feitas nos recursos que diferem da sua configuração. A opção Default (Padrão) é executada uma vez. Se não quiser que o Quick Setup corrija as alterações feitas nos recursos diferentes de sua configuração, escolha Disabled (Desabilitado) em Custom (Personalizar).

6. Na seção Targets (Destinos_, escolha se deseja permitir que o DevOps Guru analise recursos em algumas de suas unidades organizacionais (UOs) ou na conta na qual você está conectado.

   Se escolher Custom (Personalizado), continue na etapa 8.

   Se escolher Current account (Conta corrente), continue na etapa 9.
7. Nas seções Target OUs (UOs de destno) e Target Regions (Regiões de destino), marque as caixas de seleção das UOs e Regiões nas quais você deseja usar o DevOps Guru.
8. Escolha as Regiões nas quais você quer usar o DevOps Guru na conta atual.
9. Escolha Criar.

Implantar pacotes do Distributor usando o Quick Setup

O Distributor é uma ferramenta do AWS Systems Manager. O pacote do Distributor é uma coleção de softwares instaláveis ou ativos que podem ser implantados como uma única entidade. Com o Quick Setup, você pode implantar um pacote do Distributor em uma Conta da AWS e uma Região da AWS ou por meio de uma organização no AWS Organizations. No momento, só é possível implantar o agente do EC2Launch v2, o pacote de utilitários do Amazon Elastic File System (Amazon EFS) e o agente do Amazon CloudWatch com a Quick Setup. Para obter mais informações sobre o Distributor, consulte AWS Systems Manager Distributor.

Para implantar o Distributor, realize as seguintes tarefas no AWS Systems Manager Quick Setup console do.

Para implantar pacotes do Distributor com o Quick Setup

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Quick Setup.
3. No cartão Distributor, escolha Criar.

   > **Tip**
   >
   > Se você já tem uma ou mais configurações na conta, primeiro escolha a guia Biblioteca ou o botão Criar na seção Configurações para ver os cartões.
4. NoOpções de configuração, escolha o pacote que você deseja implantar.
5. Na seção Targets (Destinos), escolha se deseja implantar o pacote em toda a sua organização, em algumas de suas unidades organizacionais (UOs) ou na conta na qual você está conectado.

Se escolher Entire organization (Toda a organização), continue na etapa 8.

Se escolher Custom (Personalizado), continue na etapa 7.

6. NoUOs de destino, marque as caixas de seleção das UOs e Regiões nas quais você deseja implantar o pacote.
7. Escolha Criar.

Parar e iniciar instâncias do EC2 automaticamente de acordo com uma programação usando a Quick Setup

Com o Quick Setup, uma ferramenta do AWS Systems Manager, é possível configurar o Agendador de Recursos para automatizar o início e o encerramento das instâncias do Amazon Elastic Compute Cloud (Amazon EC2).

Essa configuração do Quick Setup ajuda a reduzir os custos operacionais iniciando e interrompendo instâncias de acordo com a programação que você especificar. Essa ferramenta ajuda a evitar custos desnecessários com a execução de instâncias quando elas não são necessárias.

Por exemplo, atualmente pode ser que você mantenha suas instâncias em execução constante, mesmo que elas sejam usadas apenas 10 horas por dia, 5 dias por semana. Em vez disso, é possível programar suas instâncias para serem interrompidas todos os dias após o horário comercial. Como resultado, há uma economia de 70% nessas instâncias, pois o runtime é reduzido de 168 horas para 50 horas. Não há custo para usar o Quick Setup. No entanto, é possível haver cobrança de custos pelos recursos que você configurou e limites de uso sem custo pelos serviços usados para definir sua configuração.

Com o Agendador de Recursos, é possível optar por interromper e iniciar automaticamente as instâncias em várias Regiões da AWS e Contas da AWS de acordo com uma programação definida por você. A configuração do Quick Setup visa as instâncias do Amazon EC2 usando a chave e o valor da tag que você especificar. Somente as instâncias com uma tag correspondente ao valor especificado em sua configuração são interrompidas ou iniciadas pelo Programador de recursos. Observe que, se os volumes do Amazon EBS anexados à instância forem criptografados, você deverá adicionar as permissões necessárias para a chave do AWS KMS para o perfil do IAM para que o Agendador de Recursos inicie a instância.

Máximo de instâncias por configuração

Uma configuração individual oferece suporte ao agendamento de até 5.000 instâncias por região. Se seu caso exigir que mais de 5.000 instâncias sejam programadas em uma determinada região, você deverá criar várias configurações. Aplique tags em suas instâncias adequadamente para que cada configuração gerencie até 5.000 instâncias. Ao criar várias configurações do Quick Setup do Programador de recursos, você deve especificar valores de chave de tag diferentes. Por exemplo, uma configuração pode usar a chave de tag `Environment` com o valor `Production`, enquanto outra usa `Environment` e `Development`.

Comportamentos de agendamento

Os pontos a seguir descrevem alguns comportamentos das configurações de agendamento:

- O Programador de recursos iniciará as instâncias marcadas somente se elas estiverem no estado `Stopped`. Da mesma forma, as instâncias só são interrompidas se estiverem no estado `running`. O Programador de recursos opera em um modelo orientado por eventos e só inicia ou interrompe instâncias nos horários que você especificar. Por exemplo, você cria uma programação que inicia as instâncias às 9h. O Programador de recursos iniciará todas as instâncias associadas à tag especificada que estejam no estado `Stopped` às 9h. Se as instâncias forem interrompidas manualmente mais tarde, o Programador de recursos não as iniciará novamente para manter o estado `Running`. Da mesma forma, se uma instância for iniciada manualmente depois de ser interrompida de acordo com sua programação, o Programador de recursos não interromperá a instância novamente.
- Se você criar um agendamento com horário de início posterior ao horário de interrupção em um dia de 24 horas, o Agendador de Recursos presumirá que suas instâncias devem ser executadas durante a noite. Por exemplo, você cria uma programação que inicia as instâncias às 21h e as interrompe às 7h da manhã. O Programador de recursos iniciará todas as instâncias associadas à tag especificada que estejam no estado `Stopped` às 9h, e as interromperá às 7h do dia seguinte. Para programações noturnas, o horário de início se aplica aos dias que você selecionar para sua programação. No entanto, o horário de interrupção se aplica ao dia seguinte em sua programação.
- Quando você cria uma configuração de agendamento, o estado atual de suas instâncias pode ser alterado para corresponder aos requisitos do agendamento.

  Por exemplo, digamos que hoje seja uma quarta-feira e você especifique um agendamento para que suas instâncias gerenciadas iniciem às 9h e parem às 17h somente às terças e quintas-feiras. Como seu horário atual está fora do horário prescrito para a execução das instâncias, elas serão interrompidas após a criação da configuração. As instâncias não serão executadas novamente até o próximo horário prescrito, às 9h de quinta-feira.

Se suas instâncias estiverem atualmente em um estado `Stopped` e você especificar um agendamento no qual elas seriam executadas no instante atual, o Agendador de Recursos as iniciará após a criação da configuração.

Se você excluir sua configuração, as instâncias não serão mais interrompidas e iniciadas de acordo com a programação definida anteriormente. Em casos raros, as instâncias podem não ser interrompidas ou iniciadas com êxito devido a falhas na operação da API.

Para configurar a programação das instâncias do Amazon EC2, realize as tarefas a seguir no console do AWS Systems Manager Quick Setup.

Para configurar a programação de instâncias com o Quick Setup

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Quick Setup.
3. No cartão Agendador de Recursos, escolha Criar.

> **Tip**
>
> Se você já tem uma ou mais configurações na conta, primeiro escolha a guia Biblioteca ou o botão Criar na seção Configurações para ver os cartões.

4. Na seção Instance tag (Tag da instância), especifique a chave e o valor da tag aplicados às instâncias que você deseja associar à sua programação.
5. Na seção Schedule options (Opções de agendamento), especifique o fuso horário, os dias e os horários em que você deseja iniciar e interromper suas instâncias.
6. Na seção Targets (Destinos), escolha se deseja configurar a programação para um grupo Custom (Personalizado) de unidades organizacionais (UOs) ou para a Current account (Conta atual) à qual você está conectado:

   - Custom (Personalizado): na seção Target OUs (UOs de destino), selecione as UOs nas quais você deseja configurar a programação. Em seguida, na seção Target Regions (Regiões de destino), selecione as regiões nas quais você deseja configurar a programação.
   - Conta corrente: selecione Current Region (Região atual) ou Choose Regions (Selecionar regiões). Se você selecionou Choose Regions (Escolher regiões), escolha as Target Regions (Regiões de destino) nas quais deseja configurar a programação.

7. Verifique as informações da programação na seção Summary (Resumo).
8. Escolha Criar.

Configurar o Explorador de recursos da AWS usando a Quick Setup

Com o Quick Setup, uma ferramenta do AWS Systems Manager, é possível configurar rapidamente o Explorador de recursos da AWS para pesquisar e descobrir recursos em sua Conta da AWS ou em uma organização da AWS inteira. É possível pesquisar seus recursos usando metadados, como nomes, tags e IDs. O Explorador de recursos da AWS fornece respostas rápidas às suas consultas de pesquisa usando índices. O Explorador de Recursos cria e mantém índices usando várias fontes de dados para coletar informações sobre os recursos na sua Conta da AWS.

O Quick Setup para Explorador de Recursos automatiza o processo de configuração do índice. Para obter mais informações sobre o Explorador de recursos da AWS, consulte O que é Explorador de recursos da AWS? no Guia do Usuário do Explorador de recursos da AWS.

Durante a Quick Setup, o Explorador de Recursos fará o seguinte:

- Criará um índice em toda Região da AWS na sua Conta da AWS.
- Atualizará o índice na região que você especificar para ser o índice agregador para a conta.
- Criará uma visualização padrão na região do índice agregador. Essa visualização não terá filtros, portanto, retornará todos os recursos encontrados no índice.

Permissões mínimas

Para realizar as etapas do procedimento a seguir, você deve ter as seguintes permissões:

- Ação: `resource-explorer-2:*` - Recurso: nenhum recurso específico (*)
- Ação: `iam:CreateServiceLinkedRole` - Recurso: nenhum recurso específico (*)

Para configurar o Explorador de Recursos

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Quick Setup.
3. No cartão Explorador de Recursos, escolha Criar.

4. Na seção Região do Índice do Agregador, escolha qual região você deseja que contenha o índice do agregador. Você deve selecionar a região apropriada para a localização geográfica dos usuários.
5. (Opcional) Marque a caixa de seleção Substituir índices agregadores existentes em regiões diferentes da selecionada acima.
6. Na seção Destinos, escolha a organização de destino ou especifique Unidades Organizacionais (OUs) contendo os recursos que você deseja descobrir.
7. Na seção Regiões, escolha quais Regiões serão incluídas na configuração.
8. Revise o resumo da configuração e, em seguida, escolha Criar.

Na página Explorador de Recursos, é possível monitorar o status da configuração.

Solução de problemas de resultados do Quick Setup

Use as informações a seguir para obter ajuda para solucionar problemas com o Quick Setup, uma ferramenta do AWS Systems Manager. Este tópico inclui tarefas específicas para resolver problemas com base no tipo de problema do Quick Setup.

Problema: falha na implantação

Uma implantação falhará se o conjunto de pilhas do CloudFormation falhar durante a criação. Use as etapas a seguir para investigar uma falha de implantação.

1. Navegue até o console do AWS CloudFormation.
2. Escolha a pilha criada pelo sua configuração do Quick Setup. O Stack name (Nome da pilha) inclui `QuickSetup` seguido pelo tipo de configuração que você escolheu, como `SSMHostMgmt`.

   > **Note**
   >
   > Às vezes, o CloudFormation exclui implantações de pilha com falha. Se a pilha não estiver disponível na tabela Stacks (Pilhas), escolha Deleted (Excluído) na lista de filtros.

3. Visualize o Status e Status reason (Motivo do status). Para obter mais informações sobre status de pilhas, consulte Códigos de status da pilha no Guia do usuário do AWS CloudFormation.

4. Para entender a etapa exata que falhou, veja a guia Events (Eventos) e revise cada Status de evento.
5. Revise Troubleshooting (Solução de problemas) no Guia do usuário do AWS CloudFormation.
6. Se você não conseguir resolver a falha de implantação usando as etapas de solução de problemas do CloudFormation, exclua a configuração e reconfigure.

Problema: falha na associação

A tabela Configuration details (Detalhes da configuração) na página Configuration details (Detalhes da configuração) da configuração exibirá um Configuration status (Status da configuração) de Failed (Falhou) se alguma das associações falhar durante a configuração. Use as etapas a seguir para solucionar problemas de uma associação com falha.

1. Na tabela Configuration details (Detalhes da configuração), escolha a configuração com falha e escolha Exibir detalhes.
2. Copie o Association name (Nome da associação).
3. Navegue até State Manager e cole o nome da associação no campo de pesquisa.
4. Escolha a associação e escolha a guia Execution history (Histórico da execução).
5. Em Execution ID (ID da execução), escolha a execução da associação que falhou.
6. A página Association execution targets (Destinos da execução de associação) lista todos os nós em que a associação foi executada. Escolha o botão Output (Saída) de uma execução que falhou.
7. Na página Output (Saída), escolha Step - Output (Etapa - saída) para visualizar a mensagem de erro dessa etapa na execução do comando. Cada etapa pode exibir uma mensagem de erro diferente. Analise as mensagens de erro de todas as etapas para ajudar a solucionar o problema.

Se a exibição da saída da etapa não solucionar o problema, você poderá tentar recriar a associação. Para recriar a associação, primeiro exclua a associação com falha no State Manager. Depois de excluir a associação, edite a configuração e escolha a opção que você excluiu. Em seguida, escolha Update (Atualização).

> **Note**
>
> Para investigar associações Failed (Com falha) para uma configuração de Organization (Organização), você deve fazer login na conta com a associação com falha e usar o procedimento da associação com falha a seguir, descrito anteriormente. O ID da associação não é um hiperlink para a conta de destino ao visualizar os resultados da conta de gerenciamento.

Problema: status de desvio

Ao exibir a página de detalhes de uma configuração, você pode consultar o status de desvio de cada implantação. O desvio de configuração ocorre sempre que um usuário faz qualquer alteração em um serviço ou recurso que entre em conflito com as seleções feitas por meio do Quick Setup. Se uma associação tiver sido alterada após a configuração inicial, a tabela exibirá um ícone de aviso que indica o número de itens que sofreram desvio. Você pode determinar o que causou o desvio passando o mouse sobre o ícone.

Quando uma associação é excluída em State Manager, as implantações relacionadas exibem um aviso de desvio. Para corrigir isso, edite a configuração e escolha a opção que foi removida quando a associação foi excluída. Selecione Update (Atualização) e aguarde até que a implantação seja concluída.

Ferramentas de aplicações do AWS Systems Manager

As Ferramentas de aplicações são um conjunto de recursos que ajudam você a gerenciar as aplicações em execução na AWS.

Tópicos

- AWS AppConfig
- AWS Systems Manager Application Manager
- AWS Systems Manager Parameter Store

AWS AppConfig

As informações sobre o AWS AppConfig foram movidas para guias separados. Para obter mais informações, consulte:

- Guia do usuário do AWS AppConfig
- Referência de API do AWS AppConfig

AWS Systems Manager Application Manager

O Application Manager, uma ferramenta do AWS Systems Manager, ajuda os engenheiros de DevOps a investigar e corrigir problemas com os recursos da AWS no contexto de suas aplicações e clusters. O Application Manager agrega informações de operações de vários Serviços da AWS e de ferramentas do Systems Manager em um único AWS Management Console.

No Application Manager, uma aplicação é um grupo lógico de recursos da AWS que você deseja operar como uma unidade. Esse grupo lógico pode representar diferentes versões de uma aplicação, limites de propriedade para operadores ou ambientes de desenvolvedor, entre outras. O suporte do Application Manager para clusters de contêiner inclui clusters do Amazon Elastic Kubernetes Service (Amazon EKS) e do Amazon Elastic Container Service (Amazon ECS).

Na primeira vez que você abrir o Application Manager, a página O que o Application Manager pode fazer por você será exibida. Quando você escolhe Começar, o Application Manager importa automaticamente os metadados sobre os recursos criados em outros Serviços da AWS ou em ferramentas do Systems Manager. Em seguida, o Application Manager exibe esses recursos em uma lista agrupada por categorias predefinidas.

Para Applications (Aplicações), a lista inclui o seguinte:

- Pilhas do AWS CloudFormation
- Aplicações personalizadas
- Aplicações do AWS Launch Wizard
- Aplicações do AppRegistry
- Aplicações do AWS SAP Enterprise Workload
- Clusters do Amazon ECS
- Clusters do Amazon EKS

Depois de você definir e configurar Serviços da AWS e ferramentas do Systems Manager, o Application Manager exibirá os seguintes tipos de informações sobre os recursos:

- Informações sobre o estado atual, o status e a integridade do Amazon EC2 Auto Scaling das instâncias do Amazon Elastic Compute Cloud (Amazon EC2) na sua aplicação

- Alarmes fornecidos pelo Amazon CloudWatch
- Informações de conformidade fornecidas pela AWS Config e State Manager (um componente do Systems Manager)
- Informações de cluster do Kubernetes fornecidas pelo Amazon EKS
- Dados de log fornecidos pelo AWS CloudTrail e Amazon CloudWatch Logs
- OpsItems fornecido pelo OpsCenter do Systems Manager
- Detalhes do recurso fornecidos pelos Serviços da AWS que os hospedam.
- Informações de cluster de contêiner fornecidas pelo Amazon ECS.

Para ajudar você a corrigir problemas com componentes ou recursos, o Application Manager também fornece runbooks que você pode associar às aplicações. Para começar a usar o Application Manager, abra o Systems Manager console (Console do gerenciador de sistemas). No painel de navegação, escolha Application Manager.

Quais são os benefícios do uso do Application Manager?

O Application Manager reduz o tempo necessário para que os engenheiros de DevOps detectem e investiguem problemas com os recursos da AWS. Para fazer isso, o Application Manager exibe vários tipos de informações de operações no contexto de uma aplicação em um console. O Application Manager também reduz o tempo necessário para corrigir problemas fornecendo runbooks que executam tarefas comuns de remediação em recursos da AWS.

Quais são os recursos do Application Manager?

O Application Manager inclui os seguintes recursos:

- Importe os recursos da AWS automaticamente

  Durante a configuração inicial, você poderá optar por configurar o Application Manager para importar e exibir recursos automaticamente na Conta da AWS, que se baseiam em pilhas do CloudFormation, AWS Resource Groups, implantações Launch Wizard, aplicações do AppLogisty, além de clusters do Amazon ECS e do Amazon EKS. O sistema exibe esses recursos em categorias predefinidas da aplicação ou do cluster. Posteriormente, sempre que novos recursos desses tipos forem adicionados à Conta da AWS, o Application Manager exibirá automaticamente os novos recursos nas categorias de aplicações e cluster predefinidos.
- Crie ou edite pilhas e modelos do CloudFormation

O Application Manager ajuda você a provisionar e gerenciar recursos para suas aplicações integrando o CloudFormation. Você pode criar, editar e excluir modelos e pilhas do AWS CloudFormation no Application Manager. O Application Manager também inclui uma biblioteca de modelos na qual você pode clonar, criar e armazenar modelos. O Application Manager e o CloudFormation exibem as mesmas informações sobre o status atual de uma pilha. Modelos e atualizações de modelos são armazenados no Systems Manager até que você provisione a pilha, momento em que as alterações também são exibidas no CloudFormation.

- Visualize informações sobre suas instâncias no contexto de uma aplicação

  O Application Manager se integra ao Amazon Elastic Compute Cloud (Amazon EC2) para exibir informações sobre suas instâncias no contexto de uma aplicação. O Application Manager exibe o estado da instância, o status e a integridade do Amazon EC2 Auto Scaling para uma aplicação selecionada em um formato gráfico. A guia Instâncias também inclui uma tabela com as informações a seguir para cada instância na sua aplicação.

  - Estado da instância (pendente, sendo interrompida, em execução, interrompida)
  - Status de ping para SSM Agent
  - Status e nome do runbook mais recente do Systems Manager Automation processado na instância
  - Uma contagem de alarmes do Amazon CloudWatch Logs por estado.
    - `ALARM`: a métrica ou a expressão está fora do limite definido.
    - `OK`: a métrica ou a expressão está dentro do limite definido.
    - `INSUFFICIENT_DATA`: o alarme acabou de ser acionado, a métrica não está disponível ou não há dados suficientes para a métrica determinar o estado do alarme.
  - Integridade do grupo do Auto Scaling para os grupos de escalabilidade automática pai e individual
- Visualizar métricas operacionais e alarmes para uma aplicação ou cluster

  Application Manager O integra-se ao Amazon CloudWatch para fornecer métricas operacionais e alarmes em tempo real para uma aplicação ou cluster. Você pode se aprofundar na árvore de aplicações para exibir alarmes em cada nível de componente ou exibir alarmes para um cluster individual.
- Visualizar dados de log de uma aplicação

  O Application Manager integra-se ao Amazon CloudWatch Logs para fornecer dados de log no contexto da sua aplicação sem precisar sair do Systems Manager.

- Visualizar e gerenciar o OpsItems para uma aplicação ou cluster

  O Application Manager integra-se ao AWS Systems Manager OpsCenter para fornecer uma lista de itens de trabalho operacionais (OpsItems) para suas aplicações e clusters. A lista reflete automaticamente os OpsItems gerados e criados manualmente. Você pode visualizar os detalhes do recurso que criou um OpsItem e o status, origem e gravidade do OpsItem.

- Visualizar dados de conformidade de recursos para uma aplicação ou cluster

  O Application Manager integra-se ao AWS Config para fornecer detalhes de conformidade e histórico sobre os recursos da AWS, de acordo com as regras que você especificar. O Application Manager também se integra ao AWS Systems Manager State Manager para fornecer informações de conformidade sobre o estado que você deseja manter para suas instâncias do Amazon Elastic Compute Cloud (Amazon EC2).

- Visualizar informações sobre a infraestrutura de clusters do Amazon ECS e Amazon EKS

  O Application Manager integra-se ao Amazon ECS e ao Amazon EKS para fornecer informações sobre a integridade das infraestruturas de cluster e uma visualização de runtime de componentes dos recursos de computação, rede e armazenamento em um cluster.

  No entanto, você não pode gerenciar ou visualizar informações de operações sobre seus pods ou contêineres do Amazon EKS no Application Manager. Você só pode gerenciar e visualizar informações de operações sobre a infraestrutura que está hospedando os recursos do Amazon EKS.

- Exibir detalhes do custo do recurso de uma aplicação

  O Application Manager é integrado ao AWS Cost Explorer, um recurso do Gerenciamento de Faturamento e Custos da AWS, por meio do widget Cost (Custo). Após ativar o Cost Explorer no console do Gerenciamento de Faturamento e Custos, o widget Cost (Custo) no Application Manager mostrará os dados de custo para um aplicativo fora de um contêiner específico ou um componente de aplicativo. Você pode usar filtros no widget para exibir dados de custo de acordo com diferentes períodos de tempo, granularidades e tipos de custo em um gráfico de barras ou linhas.

- Visualizar informações detalhadas de recursos em um console

  Escolha um nome de recurso listado no Application Manager e exiba as informações contextuais e de operações sobre esse recurso, sem precisar sair do Systems Manager.

- Receba atualizações automáticas de recursos para aplicações

Se você fizer alterações em um recurso em um console de serviço e esse recurso fizer parte de uma aplicação no Application Manager, o Systems Manager exibirá essas alterações automaticamente. Por exemplo, se você atualizar uma pilha no console do AWS CloudFormation e se essa pilha fizer parte de uma aplicação do Application Manager, as atualizações da pilha serão refletidas automaticamente no Application Manager.

- Descobrir aplicações do Launch Wizard automaticamente

  O Application Manager é integrado ao AWS Launch Wizard. Se você usou o Launch Wizard para implantar recursos em uma aplicação, o Application Manager poderá importá-los e exibi-los automaticamente em uma seção do Launch Wizard.

- Monitorar recursos de aplicações no Application Manager usando o CloudWatch Application Insights

  O Application Manager integra-se ao Amazon CloudWatch Application Insights. O Application Insights identifica e configura as principais métricas, logs e alarmes na pilha de tecnologia e nos recursos da aplicação. O Application Insights monitora continuamente os logs e as métricas para detectar e correlacionar anomalias e erros. Quando erros e anomalias são detectados, o Application Insights gera CloudWatch Events que você pode usar para configurar notificações ou executar ações. Você pode habilitar e exibir o Application Insights em Overview (Visão geral) e nas guias Monitoring (Monitoramento) no Application Manager. Para obter mais informações sobre o Application Insights, consulte O que é o Amazon CloudWatch Application Insights no Manual do usuário do Amazon CloudWatch.

- Corrija problemas com runbooks

  O Application Manager inclui runbooks predefinidos do Systems Manager para corrigir problemas comuns com os recursos da AWS. É possível executar um runbook em todos os recursos aplicáveis em uma aplicação sem precisar sair do Application Manager.

Há cobrança pelo uso do Application Manager?

O Application Manager está disponível sem custo adicional.

Quais são as cotas de recursos para o Application Manager?

Visualize cotas de todas as ferramentas do Systems Manager em Cotas de serviço do Systems Manager no Referência geral da Amazon Web Services. A menos que especificado de outra forma, cada cota é específica da região.

Tópicos

- Configurando serviços relacionados
- Configurar permissões para o Systems Manager Application Manager
- Adicionar aplicações e clusters de contêiner ao Application Manager
- Como trabalhar com aplicações do

Configurando serviços relacionados

O Application Manager, uma ferramenta do AWS Systems Manager, exibe recursos e informações de outros Serviços da AWS e ferramentas do Systems Manager. Para maximizar a quantidade de informações de operações exibidas no Application Manager, recomendamos que você instale e configure esses outros serviços ou ferramentas antes de usar o Application Manager.

Tópicos

- Configure tarefas para importar recursos
- Configure tarefas para exibir informações de operações sobre recursos

Configure tarefas para importar recursos

As tarefas de configuração a seguir ajudam você a exibir recursos da AWS no Application Manager. Após a conclusão de cada uma dessas tarefas, o Systems Manager pode importar recursos automaticamente para o Application Manager. Depois que seus recursos forem importados, você poderá criar aplicações no Application Manager e mover seus recursos importados para eles. Isso ajuda você a visualizar informações de operações no contexto de uma aplicação.

(Opcional) Organizar seu recursos da AWS usando tags

> Você pode atribuir metadados aos seus recursos da AWS na forma de tags. Cada tag é um rótulo que consiste em um valor e uma chave definida pelo usuário. As tags podem ajudar você a gerenciar, identificar, organizar, pesquisar e filtrar recursos. Você pode criar tags para categorizar recursos por finalidade, proprietário, ambiente ou outros critérios.

(Opcional) Organize seu recursos da AWS usando as AWS Resource Groups

> É possível usar grupos de recursos para organizar seus recursos da AWS. Os grupos de recursos facilitam o gerenciamento, o monitoramento e a automatização de tarefas em vários recursos de uma vez.

O Application Manager importa automaticamente todos os grupos de recursos e lista-os na categoria de aplicações personalizadas.

(Opcional) Configure e implante os recursos da AWS usando o AWS CloudFormation

O AWS CloudFormation permite que você crie e provisione implantações de infraestrutura da AWS de maneira previsível e repetida. Ele ajuda você a usar Serviços da AWS como Amazon EC2, Amazon Elastic Block Store (Amazon EBS), Amazon Simple Notification Service (Amazon SNS), Elastic Load Balancing e AWS Auto Scaling. Com o CloudFormation, você pode criar aplicações confiáveis, escaláveis e econômicas na nuvem, sem se preocupar com a criação e a configuração da infraestrutura da AWS subjacente.

O Application Manager importa automaticamente todos os AWS CloudFormation e lista-os na categoria de pilhas do AWS CloudFormation. Você pode criar pilhas e modelos do CloudFormation no Application Manager. As alterações da pilha e do modelo são sincronizadas automaticamente entre o Application Manager e o CloudFormation. Você também pode criar aplicações no Application Manager e transferir pilhas para elas. Isso ajuda você a visualizar informações de operações para os recursos da pilha no contexto de uma aplicação. Para obter informações sobre a definição de preço, consulte Definição de preço do AWS CloudFormation.

(Opcional) Configure e implante as aplicações usando o AWS Launch Wizard

O Launch Wizard orienta você durante o processo de dimensionamento, configuração e implantação de recursos da AWS para aplicações de terceiros, sem a necessidade de identificar e provisionar manualmente os recursos da AWS.

O Application Manager importa automaticamente todos os recursos do Launch Wizard e lista-os na categoria Launch Wizard. Para obter mais informações sobre o AWS Launch Wizard, consulte Conceitos básicos da AWS Launch Wizard para SQL Server. O Launch Wizard está disponível sem custo adicional. Você só paga pelos recursos da AWS que você provisionar para executar sua solução.

(Opcional) Configure e implante as aplicações em contêineres usando o Amazon ECS e Amazon EKS

O Amazon Elastic Container Service (Amazon ECS) é um serviço de gerenciamento de contêineres altamente escalável e rápido que facilita a execução, a interrupção e o gerenciamento de contêineres em um cluster. Os contêineres são definidos em uma definição de tarefa que você usa para executar tarefas individuais ou tarefas em um serviço.

O Amazon EKS é um serviço gerenciado que ajuda você a executar o Kubernetes na AWS, eliminando a necessidade de instalar, operar e manter seu próprio ambiente de gerenciamento

ou nós do Kubernetes. O Kubernetes é um sistema de código aberto para automatizar a implantação, a escalabilidade e o gerenciamento de aplicações em contêineres.

O Application Manager importa automaticamente todos os recursos de infraestrutura do Amazon ECS e do Amazon EKS e os lista na guia Container clusters (Clusters de contêineres). No entanto, você não pode gerenciar ou visualizar informações de operações sobre seus pods ou contêineres do Amazon EKS no Application Manager. Você só pode gerenciar e visualizar informações de operações sobre a infraestrutura que está hospedando os recursos do Amazon EKS. Para obter mais informações, consulte Preço do Amazon ECS e Preço do Amazon EKS.

Configure tarefas para exibir informações de operações sobre recursos

As tarefas de configuração a seguir ajudam você a exibir informações de operações sobre os recursos da AWS no Application Manager.

(Recomendado) Verificar permissões do runbook

Corrija problemas com recursos da AWS no Application Manager, usando os runbooks do Systems Manager Automation. Para usar essa ferramenta de correção, é necessário configurar ou verificar as permissões. Para obter informações sobre a definição de preço, consulte Definição de preço do AWS Systems Manager.

(Opcional) Habilitar o Explorador de Custos

O AWS Cost Explorer é um atributo do AWS Cost Management que pode ser usado para visualizar seus dados de custo para análise posterior. Ao habilitar o Explorador de Custos, é possível visualizar informações de custo, histórico de custos e otimização de custos dos recursos de sua aplicação no console do Application Manager.

(Opcional) Definir e configurar os logs e alarmes do Amazon CloudWatch

O CloudWatch é um serviço de monitoramento e gerenciamento que fornece dados e insights acionáveis para a AWS, aplicações híbridas multinuvem e recursos de infraestrutura. Com o CloudWatch, você pode coletar e acessar todos os dados operacionais e de performance na forma de logs e métricas, em uma única plataforma. Para visualizar logs e alarmes do CloudWatch para os recursos no Application Manager, você deve configurar o CloudWatch. Para obter informações de preço, consulte Preço do CloudWatch.

> **Note**
>
> O suporte ao CloudWatch Logs se aplica somente a aplicações, não a clusters.

(Opcional) Definir e configurar o AWS Config

O AWS Config fornece uma visualização detalhada dos recursos associados à sua Conta da AWS, incluindo como eles são configurados, como eles se relacionam entre si e como as configurações e seus relacionamentos foram alterados ao longo do tempo. Você pode usar o AWS Config para avaliar as definições de configuração de seus recursos da AWS. Você pode fazer isso criando regras AWS Config, que representam suas definições ideais de configuração. Enquanto o AWS Config monitora continuamente as alterações de configuração que ocorrem entre seus recursos, ele verifica se essas alterações violam alguma das condições em suas regras. Se um recurso viola uma regra, o AWS Config sinaliza o recurso e a regra como incompatível. O Application Manager exibe informações de conformidade sobre as regras do AWS Config. Para visualizar esses dados no Application Manager, é necessário configurar o AWS Config. Para obter informações sobre a definição de preço, consulte Definição de preço do AWS Config.

(Opcional) Crie associations do State Manager

Você pode usar o Systems Manager State Manager para criar uma configuração que você atribui aos nós gerenciados. A configuração, chamada de associação, define o estado que você deseja manter em seus nós. Para visualizar dados de conformidade da associação no Application Manager, você deve configurar uma ou mais associações do State Manager. O State Manager é oferecido sem custo adicional.

(Opcional) Definir e configurar o OpsCenter

Você pode visualizar itens de trabalho operacionais (OpsItems) sobre seus recursos no Application Manager, usando o OpsCenter. Você pode configurar o Amazon CloudWatch e o Amazon EventBridge para enviar automaticamente o OpsItems para o OpsCenter, com base nos alarmes e eventos. Você também pode inserir OpsItems manualmente. Para obter informações sobre a definição de preço, consulte Definição de preço do AWS Systems Manager.

Configurar permissões para o Systems Manager Application Manager

Você poderá usar todos os recursos do Application Manager, uma ferramenta do AWS Systems Manager, se sua entidade do AWS Identity and Access Management (IAM) (por exemplo, usuários,

grupos ou perfis) tiver acesso às operações de API listadas neste tópico. As operações da API são separadas em duas tabelas para ajudar você a entender as diferentes funções que elas desempenham.

A tabela a seguir lista as operações de API chamadas pelo Systems Manager se você escolher um recurso no Application Manager porque quer visualizar os detalhes dele. Por exemplo, se o Application Manager listar um grupo do Amazon EC2 Auto Scaling e, se você escolher esse grupo para exibir seus detalhes, o Systems Manager chamará as operações de API do `autoscaling:DescribeAutoScalingGroups`. Se você não tiver nenhum grupo de Auto Scaling em sua conta, essa operação de API não será chamada do Application Manager.

Somente detalhes do recurso

```
acm:DescribeCertificate
acm:ListTagsForCertificate
autoscaling:DescribeAutoScalingGroups
cloudfront:GetDistribution
cloudfront:ListTagsForResource
cloudtrail:DescribeTrails
cloudtrail:ListTags
cloudtrail:LookupEvents
codebuild:BatchGetProjects
codepipeline:GetPipeline
codepipeline:ListTagsForResource
dynamodb:DescribeTable
dynamodb:ListTagsOfResource
ec2:DescribeAddresses
ec2:DescribeCustomerGateways
ec2:DescribeHosts
ec2:DescribeInternetGateways
ec2:DescribeNetworkAcls
ec2:DescribeNetworkInterfaces
ec2:DescribeRouteTables
ec2:DescribeSecurityGroups
ec2:DescribeSubnets
ec2:DescribeVolumes
ec2:DescribeVpcs
ec2:DescribeVpnConnections
ec2:DescribeVpnGateways
elasticbeanstalk:DescribeApplications
elasticbeanstalk:ListTagsForResource
```

Somente detalhes do recurso

```
elasticloadbalancing:DescribeInstanceHealth
elasticloadbalancing:DescribeListeners
elasticloadbalancing:DescribeLoadBalancers
elasticloadbalancing:DescribeTags
iam:GetGroup
iam:GetPolicy
iam:GetRole
iam:GetUser
lambda:GetFunction
rds:DescribeDBClusters
rds:DescribeDBInstances
rds:DescribeDBSecurityGroups
rds:DescribeDBSnapshots
rds:DescribeDBSubnetGroups
rds:DescribeEventSubscriptions
rds:ListTagsForResource
redshift:DescribeClusterParameters
redshift:DescribeClusterSecurityGroups
redshift:DescribeClusterSnapshots
redshift:DescribeClusterSubnetGroups
redshift:DescribeClusters
s3:GetBucketTagging
```

A tabela a seguir lista as operações de API que o Systems Manager usa para fazer alterações em aplicações e recursos listados no Application Manager ou para exibir informações de operações de uma aplicação ou recurso selecionado.

Ações e detalhes da aplicação

```
applicationinsights:CreateApplication
applicationinsights:DescribeApplication
applicationinsights:ListProblems
ce:GetCostAndUsage
ce:GetTags
ce:ListCostAllocationTags
ce:UpdateCostAllocationTagsStatus
cloudformation:CreateStack
cloudformation:DeleteStack
```

Ações e detalhes da aplicação

```
cloudformation:DescribeStackDriftDetectionStatus
cloudformation:DescribeStackEvents
cloudformation:DescribeStacks
cloudformation:DetectStackDrift
cloudformation:GetTemplate
cloudformation:GetTemplateSummary
cloudformation:ListStacks
cloudformation:UpdateStack
cloudwatch:DescribeAlarms
cloudwatch:DescribeInsightRules
cloudwatch:DisableAlarmActions
cloudwatch:EnableAlarmActions
cloudwatch:GetMetricData
cloudwatch:ListTagsForResource
cloudwatch:PutMetricAlarm
config:DescribeComplianceByConfigRule
config:DescribeComplianceByResource
config:DescribeConfigRules
config:DescribeRemediationConfigurations
config:GetComplianceDetailsByConfigRule
config:GetComplianceDetailsByResource
config:GetResourceConfigHistory
config:ListDiscoveredResources
config:PutRemediationConfigurations
config:SelectResourceConfig
config:StartConfigRulesEvaluation
config:StartRemediationExecution
ec2:DescribeInstances
ecs:DescribeCapacityProviders
ecs:DescribeClusters
ecs:DescribeContainerInstances
ecs:ListClusters
ecs:ListContainerInstances
ecs:TagResource
eks:DescribeCluster
eks:DescribeFargateProfile
eks:DescribeNodegroup
eks:ListClusters
eks:ListFargateProfiles
eks:ListNodegroups
eks:TagResource
iam:CreateServiceLinkedRole
```

Ações e detalhes da aplicação

```
iam:ListRoles
logs:DescribeLogGroups
resource-groups:CreateGroup
resource-groups:DeleteGroup
resource-groups:GetGroup
resource-groups:GetGroupQuery
resource-groups:GetTags
resource-groups:ListGroupResources
resource-groups:ListGroups
resource-groups:Tag
resource-groups:Untag
resource-groups:UpdateGroup
s3:ListAllMyBuckets
s3:ListBucket
s3:ListBucketVersions
servicecatalog:GetApplication
servicecatalog:ListApplications
sns:CreateTopic
sns:ListSubscriptionsByTopic
sns:ListTopics
sns:Subscribe
ssm:AddTagsToResource
ssm:CreateDocument
ssm:CreateOpsMetadata
ssm:DeleteDocument
ssm:DeleteOpsMetadata
ssm:DescribeAssociation
ssm:DescribeAutomationExecutions
ssm:DescribeDocument
ssm:DescribeDocumentPermission
ssm:GetDocument
ssm:GetInventory
ssm:GetOpsMetadata
ssm:GetOpsSummary
ssm:GetServiceSetting
ssm:ListAssociations
ssm:ListComplianceItems
ssm:ListDocuments
ssm:ListDocumentVersions
ssm:ListOpsMetadata
ssm:ListResourceComplianceSummaries
ssm:ListTagsForResource
```

Ações e detalhes da aplicação

```
ssm:ModifyDocumentPermission
ssm:RemoveTagsFromResource
ssm:StartAssociationsOnce
ssm:StartAutomationExecution
ssm:UpdateDocument
ssm:UpdateDocumentDefaultVersion
ssm:UpdateOpsItem
ssm:UpdateOpsMetadata
ssm:UpdateServiceSetting
tag:GetTagKeys
tag:GetTagValues
tag:TagResources
tag:UntagResources
```

Exemplo de política para todas as permissões do Application Manager

Para configurar permissões do Application Manager para uma entidade do IAM (por exemplo, usuários, grupos ou perfis), crie uma política do IAM usando o exemplo a seguir. Este exemplo de política inclui todas as operações de API usadas pelo Application Manager.

JSON

```
                {
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "acm:DescribeCertificate",
            "acm:ListTagsForCertificate",
            "applicationinsights:CreateApplication",
            "applicationinsights:DescribeApplication",
            "applicationinsights:ListProblems",
            "autoscaling:DescribeAutoScalingGroups",
            "ce:GetCostAndUsage",
            "ce:GetTags",
            "ce:ListCostAllocationTags",
            "ce:UpdateCostAllocationTagsStatus",
```

```
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:DescribeStackDriftDetectionStatus",
                "cloudformation:DescribeStackEvents",
                "cloudformation:DescribeStacks",
                "cloudformation:DetectStackDrift",
                "cloudformation:GetTemplate",
                "cloudformation:GetTemplateSummary",
                "cloudformation:ListStacks",
                "cloudformation:ListStackResources",
                "cloudformation:UpdateStack",
                "cloudfront:GetDistribution",
                "cloudfront:ListTagsForResource",
                "cloudtrail:DescribeTrails",
                "cloudtrail:ListTags",
                "cloudtrail:LookupEvents",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:DescribeInsightRules",
                "cloudwatch:DisableAlarmActions",
                "cloudwatch:EnableAlarmActions",
                "cloudwatch:GetMetricData",
                "cloudwatch:ListTagsForResource",
                "cloudwatch:PutMetricAlarm",
                "codebuild:BatchGetProjects",
                "codepipeline:GetPipeline",
                "codepipeline:ListTagsForResource",
                "config:DescribeComplianceByConfigRule",
                "config:DescribeComplianceByResource",
                "config:DescribeConfigRules",
                "config:DescribeRemediationConfigurations",
                "config:GetComplianceDetailsByConfigRule",
                "config:GetComplianceDetailsByResource",
                "config:GetResourceConfigHistory",
                "config:ListDiscoveredResources",
                "config:PutRemediationConfigurations",
                "config:SelectResourceConfig",
                "config:StartConfigRulesEvaluation",
                "config:StartRemediationExecution",
                "dynamodb:DescribeTable",
                "dynamodb:ListTagsOfResource",
                "ec2:DescribeAddresses",
                "ec2:DescribeCustomerGateways",
                "ec2:DescribeHosts",
                "ec2:DescribeInstances",
```

```
                "ec2:DescribeInternetGateways",
                "ec2:DescribeNetworkAcls",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVolumes",
                "ec2:DescribeVpcs",
                "ec2:DescribeVpnConnections",
                "ec2:DescribeVpnGateways",
                "ecs:DescribeCapacityProviders",
                "ecs:DescribeClusters",
                "ecs:DescribeContainerInstances",
                "ecs:ListClusters",
                "ecs:ListContainerInstances",
                "ecs:TagResource",
                "eks:DescribeCluster",
                "eks:DescribeFargateProfile",
                "eks:DescribeNodegroup",
                "eks:ListClusters",
                "eks:ListFargateProfiles",
                "eks:ListNodegroups",
                "eks:TagResource",
                "elasticbeanstalk:DescribeApplications",
                "elasticbeanstalk:ListTagsForResource",
                "elasticloadbalancing:DescribeInstanceHealth",
                "elasticloadbalancing:DescribeListeners",
                "elasticloadbalancing:DescribeLoadBalancers",
                "elasticloadbalancing:DescribeTags",
                "iam:CreateServiceLinkedRole",
                "iam:GetGroup",
                "iam:GetPolicy",
                "iam:GetRole",
                "iam:GetUser",
                "iam:ListRoles",
                "lambda:GetFunction",
                "logs:DescribeLogGroups",
                "rds:DescribeDBClusters",
                "rds:DescribeDBInstances",
                "rds:DescribeDBSecurityGroups",
                "rds:DescribeDBSnapshots",
                "rds:DescribeDBSubnetGroups",
                "rds:DescribeEventSubscriptions",
                "rds:ListTagsForResource",
```

```
                "redshift:DescribeClusterParameters",
                "redshift:DescribeClusters",
                "redshift:DescribeClusterSecurityGroups",
                "redshift:DescribeClusterSnapshots",
                "redshift:DescribeClusterSubnetGroups",
                "resource-groups:CreateGroup",
                "resource-groups:DeleteGroup",
                "resource-groups:GetGroup",
                "resource-groups:GetGroupQuery",
                "resource-groups:GetTags",
                "resource-groups:ListGroupResources",
                "resource-groups:ListGroups",
                "resource-groups:Tag",
                "resource-groups:Untag",
                "resource-groups:UpdateGroup",
                "s3:GetBucketTagging",
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "s3:ListBucketVersions",
                "servicecatalog:GetApplication",
                "servicecatalog:ListApplications",
                "sns:CreateTopic",
                "sns:ListSubscriptionsByTopic",
                "sns:ListTopics",
                "sns:Subscribe",
                "ssm:AddTagsToResource",
                "ssm:CreateDocument",
                "ssm:CreateOpsMetadata",
                "ssm:DeleteDocument",
                "ssm:DeleteOpsMetadata",
                "ssm:DescribeAssociation",
                "ssm:DescribeAutomationExecutions",
                "ssm:DescribeDocument",
                "ssm:DescribeDocumentPermission",
                "ssm:GetDocument",
                "ssm:GetInventory",
                "ssm:GetOpsMetadata",
                "ssm:GetOpsSummary",
                "ssm:GetServiceSetting",
                "ssm:ListAssociations",
                "ssm:ListComplianceItems",
                "ssm:ListDocuments",
                "ssm:ListDocumentVersions",
                "ssm:ListOpsMetadata",
```

```
                "ssm:ListResourceComplianceSummaries",
                "ssm:ListTagsForResource",
                "ssm:ModifyDocumentPermission",
                "ssm:RemoveTagsFromResource",
                "ssm:StartAssociationsOnce",
                "ssm:StartAutomationExecution",
                "ssm:UpdateDocument",
                "ssm:UpdateDocumentDefaultVersion",
                "ssm:UpdateOpsMetadata",
                "ssm:UpdateOpsItem",
                "ssm:UpdateServiceSetting",
                "tag:GetResources",
                "tag:GetTagKeys",
                "tag:GetTagValues",
                "tag:TagResources",
                "tag:UntagResources"
            ],
            "Resource": "*"
        }
    ]
}
```

Note

Você pode restringir a capacidade de um usuário fazer alterações em aplicações e recursos no Application Manager, removendo as seguintes operações de API da política de permissões do IAM anexadas ao usuário, grupo ou função. A remoção dessas ações cria uma experiência somente leitura no Application Manager. A seguir estão todas as APIs que permitem que os usuários façam alterações na aplicação ou em qualquer outro recurso relacionado.

```
applicationinsights:CreateApplication
ce:UpdateCostAllocationTagsStatus
cloudformation:CreateStack
cloudformation:DeleteStack
cloudformation:UpdateStack
cloudwatch:DisableAlarmActions
cloudwatch:EnableAlarmActions
cloudwatch:PutMetricAlarm
config:PutRemediationConfigurations
```

```
config:StartConfigRulesEvaluation
config:StartRemediationExecution
ecs:TagResource
eks:TagResource
iam:CreateServiceLinkedRole
resource-groups:CreateGroup
resource-groups:DeleteGroup
resource-groups:Tag
resource-groups:Untag
resource-groups:UpdateGroup
sns:CreateTopic
sns:Subscribe
ssm:AddTagsToResource
ssm:CreateDocument
ssm:CreateOpsMetadata
ssm:DeleteDocument
ssm:DeleteOpsMetadata
ssm:ModifyDocumentPermission
ssm:RemoveTagsFromResource
ssm:StartAssociationsOnce
ssm:StartAutomationExecution
ssm:UpdateDocument
ssm:UpdateDocumentDefaultVersion
ssm:UpdateOpsMetadata
ssm:UpdateOpsItem
ssm:UpdateServiceSetting
tag:TagResources
tag:UntagResources
```

Para obter informações sobre como criar e editar políticas do IAM, consulte Criar políticas do IAM, no Manual do usuário do IAM. Para obter informações sobre como atribuir essa política a uma entidade do IAM (por exemplo, usuários, grupos ou perfis), consulte Adicionar e remover permissões de identidade do IAM.

Adicionar aplicações e clusters de contêiner ao Application Manager

O Application Manager é um componente do AWS Systems Manager. No Application Manager, uma aplicação é um grupo lógico de recursos da AWS que você deseja operar como uma unidade. Esse grupo lógico pode representar diferentes versões de uma aplicação, limites de propriedade para operadores ou ambientes de desenvolvedor, entre outras.

Na primeira vez que você abrir o Application Manager, a página O que o Application Manager pode fazer por você será exibida. Quando você escolhe Começar, o Application Manager importa automaticamente os metadados sobre os recursos criados em outros Serviços da AWS ou em ferramentas do Systems Manager. Em seguida, o Application Manager exibe esses recursos em uma lista agrupada por categorias predefinidas.

Para Applications (Aplicações), a lista inclui o seguinte:

- Pilhas do AWS CloudFormation
- Aplicações personalizadas
- Aplicações do AWS Launch Wizard
- Aplicações do AppRegistry
- Aplicações do AWS SAP Enterprise Workload
- Clusters do Amazon ECS
- Clusters do Amazon EKS

Após a conclusão da importação, você pode exibir informações de operações para uma aplicação ou um recurso específico nessas categorias predefinidas. Ou, se você quiser fornecer mais contexto sobre um conjunto de recursos, você pode criar manualmente uma aplicação no Application Manager. Em seguida, você pode adicionar recursos ou grupos de recursos a essa aplicação. Depois de criar uma aplicação no Application Manager, você poderá exibir informações de operações sobre o recurso no contexto de uma aplicação.

Criar uma aplicação do Application Manager

Use o procedimento a seguir para criar uma aplicação no Application Manager e adicione recursos a essa aplicação..

Como criar um aplicativo no Application Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Application Manager.
3. Escolha Criar aplicação.
4. Escolha uma opção na lista suspensa e preencha os campos na página que é exibida.

Como trabalhar com aplicações do

O Application Manager é um componente do AWS Systems Manager. Esta seção inclui tópicos para ajudar você a trabalhar com as aplicações do Application Manager e a exibir informações de operações sobre os recursos da AWS.

Conteúdo

- Visão geral da aplicação no Application Manager
- Gerenciar instâncias do EC2 da sua aplicação
- Visualizar recursos associados à aplicação
- Gerenciar a conformidade da sua aplicação
- Usar o CloudWatch Application Insights para monitorar uma aplicação
- Visualizar OpsItems para uma aplicação
- Gerenciar seus logs de aplicação
- Usar runbooks do Automation para corrigir problemas da aplicação
- Recursos de tag no Application Manager
- Trabalhar com modelos e pilhas do AWS CloudFormation no Application Manager.
- Trabalhar com clusters no Application Manager

Visão geral da aplicação no Application Manager

No Application Manager, um componente do AWS Systems Manager, a guia Overview (Visão geral) exibe um resumo dos alarmes do Amazon CloudWatch, itens de trabalho operacionais (OpsItems), CloudWatch Application Insights e histórico de runbooks. Selecione View all (Visualizar tudo) para qualquer cartão, a fim de abrir a guia correspondente, onde você poderá ver todos os insights de aplicações, alarmes, OpsItems ou o histórico do runbook.

Sobre o Application Insights

O CloudWatch Application Insights identifica e configura as principais métricas, logs e alarmes na pilha de tecnologia e nos recursos da aplicação. O Application Insights monitora continuamente os logs e as métricas para detectar e correlacionar anomalias e erros. Quando erros e anomalias são detectados, o Application Insights gera o CloudWatch Events que você pode usar para configurar notificações ou executar ações. Se você escolher o botão Edit configuration (Editar configuração) na guia Monitoring (Monitoramento), o sistema abrirá o console do CloudWatch Application Insights.

Para obter mais informações sobre o Application Insights, consulte O que é o Amazon CloudWatch Application Insights no Manual do usuário do Amazon CloudWatch.

Sobre o Cost Explorer

O Application Manager é integrado ao AWS Cost Explorer, um atributo do Gerenciamento de Custos da AWS, por meio do widget Cost e da guia Custo. Após habilitar o Explorador de Custos no console do Gerenciamento de Custos, o widget Cost e a guia Custo no Application Manager mostrarão os dados de custo para uma aplicação fora de um contêiner específico ou um componente de aplicação. Você pode usar filtros no widget ou na guia para exibir dados de custo de acordo com diferentes períodos de tempo, níveis de granularidades e tipos de custo em um gráfico de barras ou linhas.

É possível habilitar esse atributo escolhendo o botão Ir para o console do Gerenciamento de Custos da AWS. Por padrão, os dados são filtrados nos últimos três meses. Para um aplicativo que não esteja em contêiner, se você escolher o botão View all (Visualizar tudo), o Application Manager abrirá a guia Resources (Recursos). Para aplicações de contêiner, o botão View all (Visualizar tudo) abre o console do AWS Cost Explorer.

Você pode executar qualquer uma das seguintes ações nesta página:

É possível ativar e acessar informações sobre os widgets a seguir na guia Overview (Visão geral) desta página. Quando um widget estiver habilitado, escolha View All (Visualizar tudo) para ver os detalhes relevantes da aplicação para essa área.

- Na seção Insights and Alarms (Insights e alarmes), escolha um número de gravidade para abrir a guia Monitoring (Monitoramento) onde é possível visualizar mais detalhes sobre os alarmes da gravidade escolhida.
- Na seção Cost (Custo), escolha View all (Visualizar tudo) para abrir a guia Resources (Recursos), na qual é possível visualizar os dados de custos de uma aplicação ou componente de aplicação específico.
- Na seção Compliance (Conformidade), escolha View all (Visualizar tudo) para abrir a guia Compliance (Conformidade), na qual é possível visualizar as informações de conformidade de AWS Config e associações do State Manager.

> **Note**
>
> Para visualizar os detalhes de conformidade do patch, escolha diretamente a guia Compliance (Conformidade). Em seguida, será possível visualizar os detalhes de conformidade do patch para os nós gerenciados usados pela aplicação selecionada.

- Na seção Runbooks, escolha um runbook para abrir na página Documents (Documentos) do Systems Manager, onde você poderá ver mais detalhes sobre o documento.
- Na seção OpsItems, escolha uma gravidade para abrir a guia OpsItems, onde você verá todos os OpsItems da gravidade selecionada.
- Escolha um botão View all (Visualizar tudo) para abrir a página correspondente. Você pode exibir todos os alarmes, OpsItems ou entradas do histórico do runbook para a aplicação.

Para abrir a guia Overview (Visão geral)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Application Manager.
3. Na seção Applications (Aplicações) escolha uma categoria. Se você quiser abrir uma aplicação que você criou manualmente no Application Manager, escolha Custom applications (Aplicações personalizadas).
4. Selecione a aplicação na lista. O Application Manager abrirá a guia Overview (Visão geral).

Gerenciar instâncias do EC2 da sua aplicação

O Application Manager se integra ao Amazon Elastic Compute Cloud (Amazon EC2) para exibir informações sobre suas instâncias no contexto de uma aplicação. O Application Manager exibe o estado da instância, o status e a integridade do Amazon EC2 Auto Scaling para uma aplicação selecionada em um formato gráfico. A guia Instances (Instâncias) também inclui uma tabela com as seguintes informações para cada instância em sua aplicação:

- Estado da instância (pendente, sendo interrompida, em execução, interrompida)
- Status de ping para SSM Agent
- Status e nome do runbook mais recente do Systems Manager Automation processado na instância
- Uma contagem de alarmes do Amazon CloudWatch Logs por estado.

  - `ALARM`: a métrica ou a expressão está fora do limite definido.
  - `OK`: a métrica ou a expressão está dentro do limite definido.
  - `INSUFFICIENT_DATA`: o alarme acabou de ser acionado, a métrica não está disponível ou não há dados suficientes para a métrica determinar o estado do alarme.
- Integridade do grupo do Auto Scaling para os grupos de escalabilidade automática pai e individual

Se você escolher uma instância na tabela All instances (Todas as instâncias), o Application Manager exibirá as informações sobre essa instância em quatro guias:

- Details (Detalhes): todos os detalhes da instância do Amazon EC2, incluindo a imagem de máquina da Amazon (AMI), informações de DNS, informações de endereço IP e muito mais.
- Health (Integridade): o status atual, conforme fornecido pelas verificações de status do sistema e da instância do EC2.
- Execution history (Histórico de execução): logs de execução dos runbooks do Systems Manager Automation e das chamadas de API processadas pela instância.
- CloudWatch alarms (Alarmes do CloudWatch): o nome, o estado e outras informações de todos os alarmes do CloudWatch gerados pela instância.

Você pode executar qualquer uma das seguintes ações nesta página:

É possível executar as seguintes ações nesta página:

- Iniciar, interromper e encerrar instâncias.
- Aplique uma receita do Chef.
- Anexe instâncias ou desanexe instâncias de um grupo do Auto Scaling.
- Habilite atualizações automatizadas para o SSM Agent.

Para abrir a guia Instances (Instâncias)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Application Manager.
3. Na seção Applications (Aplicações) escolha uma categoria. Se você quiser abrir uma aplicação que tenha criado manualmente no Application Manager, escolha Custom applications (Aplicações personalizadas).

4. Selecione a aplicação na lista. O Application Manager abrirá a guia Overview (Visão geral).
5. Escolha a guia Instâncias.

Para visualizar os detalhes das suas instâncias de aplicação

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Application Manager.
3. Na seção Applications (Aplicações) escolha uma categoria. Se você quiser abrir uma aplicação que tenha criado manualmente no Application Manager, escolha Custom applications (Aplicações personalizadas).
4. Selecione a aplicação na lista. O Application Manager abrirá a guia Overview (Visão geral).
5. Escolha a guia Instâncias.
6. Selecione o botão ao lado da instância cujos detalhes você deseja visualizar.
7. Analise os detalhes da sua instância na parte de baixo da página.

Como atualizar automaticamente o SSM Agent

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Application Manager.
3. Na seção Applications (Aplicações) escolha uma categoria. Se você quiser abrir uma aplicação que tenha criado manualmente no Application Manager, escolha Custom applications (Aplicações personalizadas).
4. Selecione a aplicação na lista. O Application Manager abrirá a guia Overview (Visão geral).
5. Escolha a guia Instâncias.
6. No menu suspenso Ações do agente, escolha Configurar atualização do SSM Agent.
7. Escolha Todas as instâncias para configurar atualizações automáticas do SSM Agent para todas as instâncias gerenciadas. Como alternativa, escolha Instância para configurar atualizações de automação do SSM Agent para uma única instância na sua aplicação.
8. Selecione o botão Habilitar atualização automática.
9. No menu suspenso Especificar programação, escolha a programação que você deseja usar para atualizações do SSM Agent.
10. Selecione Configurar.

Visualizar recursos associados à aplicação

No Application Manager, um componente do AWS Systems Manager, a guia Resources (Recursos) exibe os recursos da AWS na aplicação. Se você escolher um componente de nível superior, esta página exibirá todos os recursos desse componente e quaisquer subcomponentes. Se você escolher um subcomponente, esta página mostrará apenas os recursos atribuídos a esse subcomponente.

Você pode executar qualquer uma das seguintes ações nesta página:

É possível executar as seguintes ações nesta página:

- Escolha um nome de recurso para exibir informações sobre ele, incluindo detalhes fornecidos pelo console onde ele foi criado, tags, alarmes do Amazon CloudWatch, detalhes do AWS Config e informações de log do AWS CloudTrail.
- Selecione o botão de opção ao lado do nome de um recurso. Escolha então o botão Resource timeline (Cronograma do recurso) para abrir o console do AWS Config onde você poderá exibir informações de conformidade sobre um recurso selecionado.
- Se você ativou o AWS Cost Explorer, a seção Cost Explorer mostrará os dados de custo de uma aplicação sem contêiner ou de um componente de aplicação específico. É possível habilitar esse atributo escolhendo o botão Ir para o console do Gerenciamento de Custos da AWS. Use os filtros nesta seção para exibir informações de custo sobre sua aplicação.

Para abrir a guia Resources (Recursos)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Application Manager.
3. Na seção Applications (Aplicações) escolha uma categoria. Se você quiser abrir uma aplicação que você criou manualmente no Application Manager, escolha Custom applications (Aplicações personalizadas).
4. Selecione a aplicação na lista. O Application Manager abrirá a guia Overview (Visão geral).
5. Escolha a guia Resources (Recursos).

Gerenciar a conformidade da sua aplicação

No Application Manager, um componente do AWS Systems Manager, a página Configurations (Configurações) AWS Configexibe as informações de conformidade de regras de configuração

e recursos. Esta página também exibe informações de conformidade das associações do AWS Systems Manager State Manager. Você pode escolher um recurso, uma regra ou uma associação para abrir o console correspondente para obter mais informações. Esta página exibe informações de conformidade dos últimos 90 dias.

Você pode executar qualquer uma das seguintes ações nesta página:

É possível executar as seguintes ações nesta página:

- Escolha um nome de recurso para abrir o console do AWS Config onde você poderá exibir informações de conformidade sobre um recurso selecionado.
- Selecione o botão de opção ao lado do nome de um recurso. Escolha então o botão Resource timeline (Cronograma do recurso) para abrir o console do AWS Config onde você poderá exibir informações de conformidade sobre um recurso selecionado.
- Na seção Config rules compliance (Conformidade das regras do Config), faça o seguinte:
  - Escolha um nome de regra para abrir o console do AWS Config, onde você poderá visualizar informações sobre essa regra.
  - Selecione Add rules (Adicionar regras) para abrir o console do AWS Config onde você pode criar uma regra.
  - Selecione o botão de opção ao lado do nome de uma regra, escolha Actions (Ações) e, em seguida, escolha Manage remediation (Gerenciar correção) para alterar a ação de correção de uma regra.
  - Selecione o botão de opção ao lado do nome de uma regra, escolha Actions (Ações) e, em seguida, escolha Re-evaluate (Reavaliar) para que o AWS Config execute uma verificação de conformidade na regra selecionada.
- Na sessão Association compliance (Conformidade das associações), você pode fazer o seguinte:
  - Escolha um nome de associação para abrir a página Associations (Associações) na qual você poderá visualizar informações sobre essa associação.
  - SelecioneCreate association (Criar associação) para abrir o Systems Manager State Manager onde você pode criar uma associação.
  - Selecione o botão de opção ao lado de um nome de associação e escolhaApply association (Aplicar associação) para iniciar imediatamente todas as ações especificadas na associação.

Para abrir a guia Compliance (Conformidade)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.

2. No painel de navegação, escolha Application Manager.
3. Na seção Applications (Aplicações) escolha uma categoria. Se você quiser abrir uma aplicação que você criou manualmente no Application Manager, escolha Custom applications (Aplicações personalizadas).
4. Selecione a aplicação na lista. O Application Manager abrirá a guia Overview (Visão geral).
5. Escolha a guia Compliance (Conformidade).

Usar o CloudWatch Application Insights para monitorar uma aplicação

No Application Manager, um componente do AWS Systems Manager, a guia Monitoring (Monitoramento) exibe o Amazon CloudWatch Application Insights e os detalhes do alarme para recursos em uma aplicação.

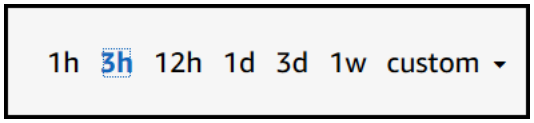
Sobre o Application Insights

O CloudWatch Application Insights identifica e configura as principais métricas, logs e alarmes na pilha de tecnologia e nos recursos da aplicação. O Application Insights monitora continuamente os logs e as métricas para detectar e correlacionar anomalias e erros. Quando erros e anomalias são detectados, o Application Insights gera o CloudWatch Events que você pode usar para configurar notificações ou executar ações. Se você escolher o botão Edit configuration (Editar configuração) na guia Monitoring (Monitoramento), o sistema abrirá o console do CloudWatch Application Insights. Para obter mais informações sobre o Application Insights, consulte O que é o Amazon CloudWatch Application Insights no Manual do usuário do Amazon CloudWatch.

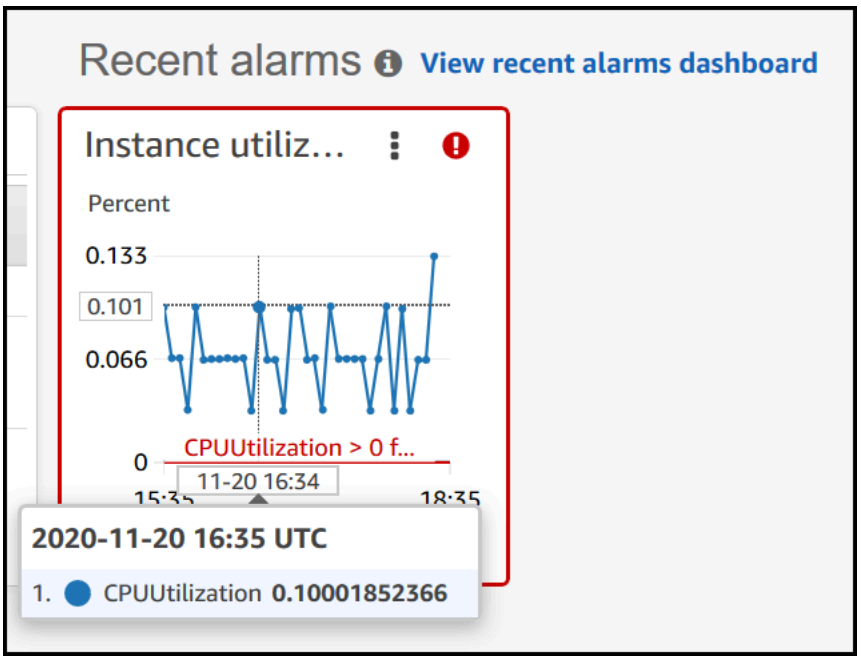
Você pode executar qualquer uma das seguintes ações nesta página:

É possível executar as seguintes ações nesta página:

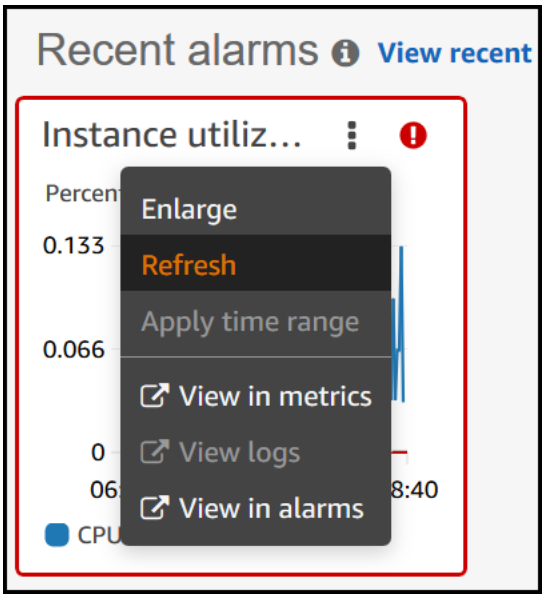
- Escolha um nome de serviço na seção Alarms by AWS service (Alarmes dos serviços) para abrir o CloudWatch para o serviço e o alarme selecionados.
- Ajuste o período de tempo para os dados exibidos em widgets na seção Recent alarms (Alarmes recentes), selecionando um dos valores predefinidos do período de tempo. Você pode escolher custom (personalizar) para definir seu próprio período de tempo.



- Passe o cursor sobre um widget na seção Recent alarms (Alarmes recentes) para exibir um pop-up de dados para um horário específico.



- Selecione o menu de opções em um widget para exibir as opções de exibição. Selecione Enlarge (Ampliar) para expandir um widget. Selecione Refresh (Atualizar) para atualizar os dados em um widget. Clique e arraste o cursor em uma exibição de dados do widget para selecionar um intervalo específico. Você pode então escolher Apply time range (Aplicar intervalo de tempo).

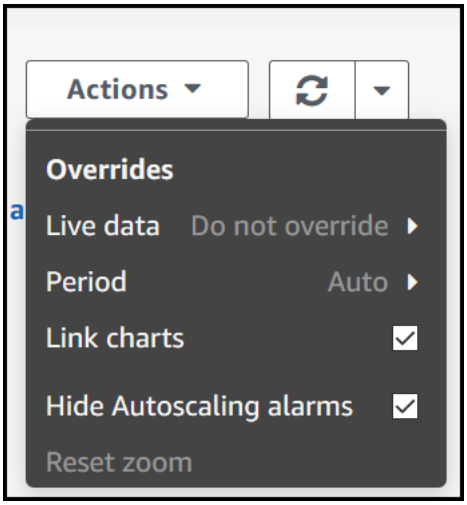


- Selecione o menu Actions (Ações) para visualizar as opções de Substituição dos dados de alarmes, que incluem o seguinte:
  - Selecione se os widgets exibirão dados em tempo real. Os dados em tempo real são os dados publicados no último minuto em que eles não foram totalmente agregados. Se os dados em tempo real estiverem desativados, somente os pontos de dados com um período de agregação

de pelo menos um minuto no passado serão exibidos. Por exemplo, ao usar períodos de 5 minutos, o ponto de dados para 12:35 seria agregado de 12:35 a 12:40 e exibido às 12:41.

Se os dados em tempo real estiverem ativados, o ponto de dados mais recente será exibido assim que todos os dados forem publicados no intervalo de agregação correspondente. Cada vez que você atualiza a exibição, o ponto de dados mais recente poderá ser alterado à medida que novos dados dentro desse período de agregação são publicados.

- Especifique um período de tempo para os dados dinâmicos.
- Vincule os gráficos na seção Recent alarms (Alarmes recentes) para que, ao ampliar ou reduzir um gráfico, o outro seja ampliado ou amplie ao mesmo tempo. É possível desvincular gráficos para limitar o zoom a um gráfico.
- Ocultar alarmes Auto Scaling.



Para abrir a guia Monitoring (Monitoramento)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Application Manager.
3. Na seção Applications (Aplicações) escolha uma categoria. Se você quiser abrir uma aplicação que você criou manualmente no Application Manager, escolha Custom applications (Aplicações personalizadas).
4. Selecione a aplicação na lista. O Application Manager abrirá a guia Overview (Visão geral).
5. Escolha a guia Monitoramento.

Visualizar OpsItems para uma aplicação

No Application Manager, um componente do AWS Systems Manager, a guia OpsItems exibe itens de trabalho operacionais (OpsItems) para obter recursos na aplicação selecionada. Você pode configurar o OpsCenter do Systems Manager para criar automaticamente o OpsItems com base nos alarmes do Amazon CloudWatch e eventos do Amazon EventBridge. Você também pode criar OpsItems manualmente.

Você pode executar qualquer uma das seguintes ações nesta guia:

É possível executar as seguintes ações nesta página:

- Filtrar a lista de OpsItems usando o campo de pesquisa. Você pode filtrar pelo nome, ID, ID de origem ou gravidade do OpsItem. Você também pode filtrar a lista com base no status. O OpsItems suporta os seguintes status: Open (Aberto), In progress (Em andamento), Aberto e em andamento (Open and In progress), Resolved (Resolvido) ou All (Todos).
- Altere o status de um OpsItem escolhendo o botão de opção ao lado dele e, em seguida, escolhendo uma opção no menu Set status (Definir status).
- Abra o Systems Manager OpsCenter para criar um OpsItem escolhendo Create OpsItem (Criar item operacional).

Para abrir a guia OpsItems

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Application Manager.
3. Na seção Applications (Aplicações) escolha uma categoria. Se você quiser abrir uma aplicação que você criou manualmente no Application Manager, escolha Custom applications (Aplicações personalizadas).
4. Selecione a aplicação na lista. O Application Manager abrirá a guia Overview (Visão geral).
5. Escolha a guia OpsItems.

Gerenciar seus logs de aplicação

No Application Manager, um componente do AWS Systems Manager, a guia Logs exibe uma lista de grupos de logs do Amazon CloudWatch Logs.

Você pode executar qualquer uma das seguintes ações nesta guia:

É possível executar as seguintes ações nesta página:

- Escolha um nome de grupo de logs para abri-lo no CloudWatch Logs. Em seguida, você pode escolher um fluxo de log para exibir logs de um recurso no contexto de uma aplicação.
- Selecione Create log groups (Criar grupos de log) para criar um grupo de logs no CloudWatch Logs.

Para abrir a guia Logs

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Application Manager.
3. Na seção Applications (Aplicações) escolha uma categoria. Se você quiser abrir uma aplicação que você criou manualmente no Application Manager, escolha Custom applications (Aplicações personalizadas).
4. Selecione a aplicação na lista. O Application Manager abrirá a guia Overview (Visão geral).
5. Escolha a guia Logs.

Usar runbooks do Automation para corrigir problemas da aplicação

É possível corrigir problemas com recursos da AWS no Application Manager, uma ferramenta do AWS Systems Manager, usando os runbooks do Automation. Um runbook do Automation define as ações que o Systems Manager realizará nas instâncias gerenciadas e outros recursos da AWS quando uma automação for executada. O Automation é uma ferramenta do AWS Systems Manager. Um runbook contém uma ou mais etapas que são executadas em ordem sequencial. Cada etapa se baseia em uma única ação. A saída de uma etapa pode ser usada como entrada de uma etapa posterior.

Quando você escolher Start runbook (Iniciar o runbook) em uma aplicação ou cluster do Application Manager, o sistema exibirá uma lista filtrada de runbooks disponíveis com base no tipo de recursos da aplicação ou cluster. Quando você escolhe o runbook que deseja iniciar, o Systems Manager abre a página Execute automation document (Executar o documento de automação).

O Application Manager contém as melhorias a seguir para trabalhar com runbooks.

- Se você escolher o nome de um recurso no Application Manager e, depois, escolher Execute runbook (Executar runbook), o sistema exibirá uma lista filtrada de runbooks para esse tipo de recurso.

- Você pode iniciar uma automação em todos os recursos do mesmo tipo escolhendo um runbook na lista e, em seguida, escolhendo Run for resources of same type (Executar para recursos do mesmo tipo).

Antes de começar

Antes de começar um runbook do Application Manager, faça o seguinte:

- Verifique se você tem as permissões corretas para iniciar runbooks. Para ter mais informações, consulte Configurar a automação.
- Examine a documentação de procedimentos do Automation sobre como iniciar runbooks. Para ter mais informações, consulte Executar uma operação automatizada com tecnologia do Systems Manager Automation.

Para iniciar um runbook no Application Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Application Manager.
3. Na seção Applications (Aplicações) escolha uma categoria. Se você quiser abrir uma aplicação que você criou manualmente no Application Manager, escolha Custom applications (Aplicações personalizadas).
4. Selecione a aplicação na lista. O Application Manager abrirá a guia Overview (Visão geral).
5. Escolha Iniciar runbook. O Application Manager abrirá o pop-up do widget Automation. Para obter informações sobre as opções no widget Automation, consulte Executar uma operação automatizada com tecnologia do Systems Manager Automation.

Recursos de tag no Application Manager

Você pode adicionar ou excluir tags rapidamente em aplicações e recursos da AWS no Application Manager. Use o procedimento a seguir para adicionar ou excluir uma tag de uma aplicação e todos os recursos da AWS nessa aplicação.

Para adicionar ou excluir uma tag de uma aplicação e todos os recursos nessa aplicação.

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Application Manager.

3. Na seção Applications (Aplicações) escolha uma categoria. Se você quiser abrir uma aplicação que você criou manualmente no Application Manager, escolha Custom applications (Aplicações personalizadas).
4. Selecione a aplicação na lista. O Application Manager abrirá a guia Overview (Visão geral).
5. Na seção Application information (Informações sobre a aplicação), selecione o número abaixo de Application tags (Tags de aplicações). Se nenhuma tag for atribuída à aplicação, o número será zero.
6. Para adicionar uma tag, escolha Add new tag (Adicionar nova tag). Especifique uma chave e um valor opcional. Para excluir uma tag, escolha Remove (Remover).
7. Escolha Save (Salvar).

Use o procedimento a seguir para adicionar ou excluir uma tag de um recurso específico no Application Manager.

Para adicionar ou excluir uma tag de um recurso

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Application Manager.
3. Na seção Applications (Aplicações) escolha uma categoria. Se você quiser abrir uma aplicação que você criou manualmente no Application Manager, escolha Custom applications (Aplicações personalizadas).
4. Selecione a aplicação na lista. O Application Manager abrirá a guia Overview (Visão geral).
5. Escolha a guia Resources (Recursos).
6. Escolha o nome de um recurso.
7. Na seção Tags, selecione Edit (Editar).
8. Para adicionar uma tag, escolha Add new tag (Adicionar nova tag). Especifique uma chave e um valor opcional. Para excluir uma tag, escolha Remove (Remover).
9. Escolha Salvar.

Trabalhar com modelos e pilhas do AWS CloudFormation no Application Manager.

O Application Manager, uma ferramenta do AWS Systems Manager, ajuda você a provisionar e gerenciar recursos para suas aplicações integrando-se ao AWS CloudFormation. Você pode criar, editar e excluir modelos e pilhas do AWS CloudFormation no Application Manager. Uma pilha é um conjunto de recursos da AWS que você pode gerenciar como uma unidade. Isso significa que você

pode criar, atualizar ou excluir uma coleção de recursos da AWS usando pilhas do CloudFormation. O modelo é um arquivo de texto formatado no JSON ou YAML que especifica os recursos que você quer provisionar nas pilhas.

O Application Manager também inclui uma biblioteca de modelos na qual você pode clonar, criar e armazenar modelos. O Application Manager e o CloudFormation exibem as mesmas informações sobre o status atual de uma pilha. Modelos e atualizações de modelos são armazenados no Systems Manager, até que você provisione a pilha, quando então as alterações também serão exibidas no CloudFormation.

Depois de criar uma pilha no Application Manager, a página CloudFormation stacks (Pilhas do CloudFormation) exibirá informações úteis sobre ela. Isso inclui o modelo usado para criá-lo, uma contagem de OpsItems para os recursos na pilha, o status da pilha e o status do desvio.

Sobre o Cost Explorer

O Application Manager é integrado ao AWS Cost Explorer, um recurso do Gerenciamento de Custos da AWS, por meio do widget Cost (Custo). Após habilitar o Cost Explorer no console do Gerenciamento de Custos, o widget Cost (Custo) no Application Manager mostrará os dados de custo para um aplicativo fora de um contêiner específico ou um componente de aplicativo. Você pode usar filtros no widget para exibir dados de custo de acordo com diferentes períodos de tempo, granularidades e tipos de custo em um gráfico de barras ou linhas.

É possível habilitar esse atributo escolhendo o botão Ir para o console do Gerenciamento de Custos da AWS. Por padrão, os dados são filtrados nos últimos três meses. Para um aplicativo que não esteja em contêiner, se você escolher o botão View all (Visualizar tudo), o Application Manager abrirá a guia Resources (Recursos). Para aplicações de contêiner, o botão View all (Visualizar tudo) abre o console do AWS Cost Explorer.

> **Note**
>
> O Cost Explorer usa tags para rastrear os custos por aplicativo. Se o seu aplicativo baseado em pilha do AWS CloudFormation não estiver configurado com a chave de tag `AppManagerCFNStackKey`, o Cost Explorer não apresentará dados de custo precisos no Application Manager. Quando a chave da etiqueta `AppManagerCFNStackKey` não for detectada, você receberá uma solicitação no console para adicionar a etiqueta à sua pilha do CloudFormation para habilitar o controle de custos. Essa adição mapeia a chave de tag para o nome do recurso da Amazon (ARN) da sua pilha e habilita o widget Cost (Custo) a exibir dados de custo precisos.

> **Important**
>
> Adicionar a etiqueta `AppManagerCFNStackKey` acionará uma atualização da pilha. As configurações manuais que forem executadas após a implantação original da pilha não serão refletidas após a adição da etiqueta do usuário. Para obter mais informações sobre o comportamento de atualização de recursos, consulte Update behaviors of stack resources no Guia do usuário do AWS CloudFormation

Antes de começar

Use os links a seguir para saber mais sobre os conceitos do CloudFormation antes de criar, editar ou excluir modelos e pilhas do CloudFormation, usando o Application Manager.

- O que é AWS CloudFormation?
- AWS CloudFormation melhores práticas
- Saiba mais sobre noções básicas de modelo
- Trabalhar com pilhas do AWS CloudFormation
- Trabalhar com modelos do AWS CloudFormation
- Modelos de exemplo

Tópicos

- Usar Application Manager para gerenciar modelos do AWS CloudFormation
- Usar o Application Manager para gerenciar pilhas do AWS CloudFormation

Usar Application Manager para gerenciar modelos do AWS CloudFormation

O Application Manager, uma ferramenta do AWS Systems Manager, inclui uma biblioteca de modelos e outras ferramentas para ajudar você a gerenciar modelos do AWS CloudFormation. Esta seção inclui as seguintes informações:

Tópicos

- Trabalhar com a biblioteca de modelos
- Criar modelos
- Edite um modelo

Trabalhar com a biblioteca de modelos

A biblioteca de modelos do Application Manager fornece ferramentas para ajudar você a exibir, criar, editar, excluir e clonar modelos. Você também pode provisionar pilhas diretamente da biblioteca de modelos. Os modelos são armazenados como documentos do Systems Manager (SSM), do tipo `CloudFormation`. Ao armazenar modelos como documentos SSM, você poderá usar controles de versão para trabalhar com diferentes versões de um modelo. Você também pode definir permissões e compartilhar modelos. Depois de provisionar uma pilha com êxito, a pilha e o modelo estarão disponíveis no Application Manager e no CloudFormation.

Antes de começar

Recomendamos que você leia os tópicos a seguir para saber mais sobre os documentos SSM antes de começar a trabalhar com modelos do CloudFormation no Application Manager.

- Documentos do AWS Systems Manager
- Compartilhar documentos do Systems Manager
- Práticas recomendadas para documentos compartilhados do SSM

Para visualizar a biblioteca de modelos no Application Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Application Manager.
3. Selecione Biblioteca de modelos do CloudFormation.

Criar modelos

O procedimento a seguir descreve como criar um modelo do CloudFormation no Application Manager. Ao criar um modelo, você insere os detalhes da pilha do modelo em JSON ou YAML. Se você não estiver familiarizado com JSON ou YAML, poderá usar o AWS Infrastructure Composer, uma ferramenta para criar e modificar modelos de maneira visual. Para obter mais informações, consulte Criar modelos visualmente com o Infrastructure Composer no Guia do usuário do AWS CloudFormation. Para obter informações sobre a estrutura e a sintaxe de um modelo, consulte Seções de modelos do AWS CloudFormation, no Guia do usuário do AWS CloudFormation.

Você também pode construir um modelo a partir de vários trechos do modelo. Os trechos do modelo fornecem exemplos que demonstram como criar modelos para um recurso específico. Por exemplo, é possível visualizar trechos de instâncias do Amazon Elastic Compute Cloud (Amazon EC2),

domínios do Amazon Simple Storage Service (Amazon S3), mapeamentos do AWS CloudFormation e muito mais. Os trechos são agrupados por recurso. Você pode encontrar trechos do AWS CloudFormation para fins gerais, na seção Trechos de modelos gerais do Manual do usuário do AWS CloudFormation.

Criar um modelo do CloudFormation usando o Application Manager (console)

Use o procedimento a seguir para criar um novo modelo do CloudFormation no Application Manager, usando o AWS Management Console.

Para criar um modelo do CloudFormation no Application Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Application Manager.
3. Selecione Biblioteca de modelos do CloudFormation e depois Criar modelo ou escolha um modelo existente e depois Ações e Clonar.
4. Para Name (Nome), insira um nome para o modelo que ajude você a identificar os recursos que ele cria ou a finalidade da pilha.
5. (Opcional) Em Version name (Nome da versão), insira um nome ou um número para identificar a versão do modelo.
6. Na seção Code editor (Editor de código), escolha YAML ou JSON e, em seguida, insira ou copie e cole o código do modelo.
7. (Opcional) Na seção Tags, aplique um ou mais pares de nome/valor de chave de tag ao modelo.
8. (Opcional) Na seção Permissions (Permissões), insira uma ID da Conta da AWS e escolha Add account (Adicionar conta). Essa ação fornece permissão de leitura para o modelo. O proprietário da conta pode provisionar e clonar o modelo, mas não pode editá-lo ou excluí-lo.
9. Escolha Criar. O modelo é salvo no serviço de documentos do Systems Manager (SSM).

Criar um modelo do CloudFormation usando o Application Manager (linha de comando)

Depois de criar o conteúdo do modelo do CloudFormation no JSON ou no YAML, você poderá usar o AWS Command Line Interface (AWS CLI) ou Ferramentas da AWS para PowerShell para salvar o modelo como um documento do SSM. Substitua cada *`espaço reservado para recurso de exemplo`* por suas próprias informações.

Antes de começar

Instale e configure a AWS CLI ou o Ferramentas da AWS para PowerShell, caso ainda não o tenha feito. Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI e Instalar o Ferramentas da AWS para PowerShell.

Linux & macOS

```
aws ssm create-document \
    --content file://path/to/template_in_json_or_yaml \
    --name "a_name_for_the_template" \
    --document-type "CloudFormation" \
    --document-format "JSON_or_YAML" \
    --tags "Key=tag-key,Value=tag-value"
```

Windows

```
aws ssm create-document ^
    --content file://C:\path\to\template_in_json_or_yaml ^
    --name "a_name_for_the_template" ^
    --document-type "CloudFormation" ^
    --document-format "JSON_or_YAML" ^
    --tags "Key=tag-key,Value=tag-value"
```

PowerShell

```
$json = Get-Content -Path "C:\path\to\template_in_json_or_yaml | Out-String
New-SSMDocument `
    -Content $json `
    -Name "a_name_for_the_template" `
    -DocumentType "CloudFormation" `
    -DocumentFormat "JSON_or_YAML" `
    -Tags "Key=tag-key,Value=tag-value"
```

Se houver êxito, o comando retornará uma resposta semelhante à seguinte.

```
{
    "DocumentDescription": {
        "Hash": "c1d9640f15fbdba6deb41af6471d6ace0acc22f213bdd1449f03980358c2d4fb",
        "HashType": "Sha256",
        "Name": "MyTestCFTemplate",
        "Owner": "428427166869",
```

```
        "CreatedDate": "2021-06-04T09:44:18.931000-07:00",
        "Status": "Creating",
        "DocumentVersion": "1",
        "Description": "My test template",
        "PlatformTypes": [],
        "DocumentType": "CloudFormation",
        "SchemaVersion": "1.0",
        "LatestVersion": "1",
        "DefaultVersion": "1",
        "DocumentFormat": "YAML",
        "Tags": [
            {
                "Key": "Templates",
                "Value": "Test"
            }
        ]
    }
```

Edite um modelo

Use o procedimento a seguir para editar um modelo do CloudFormation no Application Manager. As alterações do modelo estão disponíveis no CloudFormation depois de provisionar uma pilha que usa o modelo atualizado.

Para editar um modelo do CloudFormation no Application Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Application Manager.
3. Selecione Biblioteca de modelos do CloudFormation
4. Escolha um modelo e selecione Actions (Ações), Edit (Editar). Não é possível alterar o nome de um modelo, mas é possível alterar todos os outros detalhes.
5. Escolha Salvar. O modelo é salvo no serviço de documentos do Systems Manager.

Usar o Application Manager para gerenciar pilhas do AWS CloudFormation

O Application Manager, uma ferramenta do AWS Systems Manager, ajuda você a provisionar e gerenciar recursos para suas aplicações integrando-se ao AWS CloudFormation. Você pode criar, editar e excluir modelos e pilhas do CloudFormation no Application Manager. Uma pilha é um conjunto de recursos da AWS que você pode gerenciar como uma unidade. Isso significa que você

pode criar, atualizar ou excluir uma coleção de recursos da AWS usando pilhas do CloudFormation. O modelo é um arquivo de texto formatado no JSON ou YAML que especifica os recursos que você quer provisionar nas pilhas. Esta seção inclui as seguintes informações:

Tópicos

- Criar uma pilha
- Atualizar uma pilha

Criar uma pilha

O procedimento a seguir descreve como criar uma pilha do CloudFormation usando o Application Manager. Uma pilha é baseada em um modelo. Ao criar uma pilha, você poderá escolher um modelo existente ou criar um novo. Depois de criar a pilha, o sistema tenta imediatamente criar os recursos identificados nela. Depois que o sistema provisionar os recursos com êxito, o modelo e a pilha estarão disponíveis para visualização e edição no Application Manager e no CloudFormation

> **Note**
>
> Não há nenhum custo para usar o Application Manager na criação de uma pilha, mas você será cobrado pelos recursos da AWS criados na pilha.

Criar uma pilha do CloudFormation usando o Application Manager (console)

Use o procedimento a seguir para criar uma pilha usando o Application Manager no AWS Management Console.

Para criar uma pilha do CloudFormation

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Application Manager.
3. Selecione Criar aplicação, Pilha do CloudFormation.
4. Na seção Prepare a template (Preparar um modelo), escolha uma opção. Se escolher Use an existing template (Usar um modelo existente), você poderá usar as guias na seção Choose a template (Escolher um modelo) para localizar o modelo que você quiser.. (Se você escolher uma das outras opções, conclua o assistente para preparar um modelo.)
5. Selecione o botão que está ao lado do nome do modelo e depois selecione Próximo.

6. Na página Specify template details (Especificar detalhes do modelo), verifique os detalhes do modelo para garantir que o processo crie os recursos desejados.

   - (Opcional) Na seção Tags, aplique um ou mais pares de nome/valor de chave de tag ao modelo.
   - Tags são metadados opcionais que você atribui a um recurso. Usando tags, você pode categorizar um recurso de diferentes formas, como por finalidade, proprietário ou ambiente.
   - Escolha Próximo.

7. Na página Edit stack details (Editar detalhes da pilha), para Stack name (Nome da pilha), insira um nome que ajude a identificar os recursos criados pela pilha ou pela sua finalidade.

   - A seção Parameters (Parâmetros) inclui todos os parâmetros opcionais e obrigatórios especificados no modelo. Insira um ou mais parâmetros em cada campo.
   - (Opcional) Na seção Tags, aplique um ou mais pares de nome/valor de chave de tag à pilha.
   - (Opcional) Na seção Permissions (Permissões), especifique um nome de função (IAM) do AWS Identity and Access Management ou um nome do recurso da Amazon (ARN) do IAM. O sistema usa a função de serviço especificada para criar todos os recursos especificados em sua pilha. Se você não especificar uma função do IAM, o AWS CloudFormation usará uma sessão temporária que o sistema gera a partir de suas credenciais de usuário. Para obter mais informações sobre essa função do IAM, consulte Função de serviço do AWS CloudFormation no Manual do usuário do AWS CloudFormation.
   - Escolha Próximo.

8. Na página Review and provision (Análise e provisão), examine todos os detalhes da pilha. Escolha um botão Edit (Editar) nesta página para fazer alterações.

9. Selecione Provision stack (Provisionar pilha).

O Application Manager exibe a página CloudFormation stacks (Pilhas do CloudFormation) e o status da criação e implantação da pilha. Se o CloudFormation não conseguir criar e provisionar a pilha, consulte os tópicos a seguir no Manual do usuário do AWS CloudFormation.

- Códigos de status da pilha
- Resolução de problemasAWS CloudFormation

Após os recursos de pilha serem provisionados e executados, os usuários podem editar recursos diretamente usando o serviço subjacente que o criou. Por exemplo, um usuário pode usar o console

do Amazon Elastic Compute Cloud (Amazon EC2) para atualizar uma instância de servidor que foi criada como parte de uma pilha do CloudFormation. Algumas mudanças podem ser acidentais e algumas podem ser feitas intencionalmente para responder a eventos operacionais sensíveis ao tempo. Independentemente disso, as alterações feitas fora do CloudFormation podem complicar as operações de atualização ou exclusão de pilha. Você pode usar o status do desvio para identificar os recursos da pilha para os quais as alterações de configuração foram feitas fora do gerenciamento do CloudFormation. Para obter informações sobre status de desvios, consulte Detectar alterações de configuração não gerenciadas para pilhas e recursos.

Criar uma pilha do CloudFormation usando o Application Manager (linha de comando)

Use o seguinte procedimento da AWS Command Line Interface(AWS CLI) para provisionar uma pilha usando um modelo do CloudFormation armazenado como um documento do SSM no Systems Manager. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações. Para obter informações sobre outros procedimentos da AWS CLI para criar pilhas, consulte Criar uma pilha no Manual do usuário do AWS CloudFormation.

Antes de começar

Instale e configure a AWS CLI ou o Ferramentas da AWS para PowerShell, caso ainda não o tenha feito. Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI e Instalar o Ferramentas da AWS para PowerShell.

Linux & macOS

```
aws cloudformation create-stack \
    --stack-name a_name_for_the_stack \
    --template-url "ssm-doc://arn:aws:ssm:Region:account_ID:document/template_name"
 \
```

Windows

```
aws cloudformation create-stack ^
     --stack-name a_name_for_the_stack ^
     --template-url "ssm-doc://arn:aws:ssm:Region:account_ID:document/template_name"
 ^
```

PowerShell

```
New-CFNStack `
```

```
    -StackName "a_name_for_the_stack" `
    -TemplateURL "ssm-doc://arn:aws:ssm:Region:account_ID:document/template_name" `
```

Atualizar uma pilha

Você pode implantar atualizações em uma pilha do CloudFormation editando diretamente a pilha no Application Manager. Com uma atualização direta, você especifica atualizações para um modelo ou parâmetros de entrada. Depois de salvar e implantar as alterações, o CloudFormation atualiza os recursos da AWS de acordo com as alterações que você especificou.

Você pode visualizar as alterações que o CloudFormation fará em sua pilha antes de atualizá-la, usando conjuntos de alterações. Para obter mais informações, consulte Atualizar pilhas usando conjuntos de alterações no Manual do usuário do AWS CloudFormation.

Para atualizar uma pilha do CloudFormation no Application Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Application Manager.
3. Selecione o botão de opção que está ao lado do nome da aplicação e depois selecione Ações, Atualizar pilha.
4. Na página Specify template source (Especificar origem do modelo), escolha uma das seguintes opções e selecione Next (Próximo).
   - Selecione Use the template code currently provisioned in the stack (Usar o código de modelo provisionado atualmente na pilha) para exibir um modelo. Use a lista Versions (Versões) para selecionar um modelo de versão e escolha Next (Próximo).
   - Selecione Switch to a different template (Alternar para um modelo diferente) para escolher ou criar um novo modelo para a pilha.
5. Após terminar de fazer as alterações no modelo, selecione Next (Próximo).
6. Na página Edit stack details (Editar detalhes da pilha), você pode editar parâmetros, tags e permissões. Não é possível mudar o nome de uma pilha. Faça suas alterações e escolha Next (Salvar).
7. Na página Review and provision (Análise e provisão), examine todos os detalhes da pilha e escolha Provision stack (Provisionar pilha).

Trabalhar com clusters no Application Manager

Esta seção inclui tópicos para ajudar você a trabalhar com os clusters de contêiner do Amazon Elastic Container Service (Amazon ECS) e do Amazon Elastic Kubernetes Service (Amazon EKS) no Application Manager, um componente do AWS Systems Manager.

Conteúdo

- Trabalhar com o Amazon ECS no Application Manager
- Trabalhar com o Amazon EKS no Application Manager
- Trabalhar com runbooks para clusters

Trabalhar com o Amazon ECS no Application Manager

Com o Application Manager, uma ferramenta do AWS Systems Manager, é possível visualizar e gerenciar sua infraestrutura de cluster do Amazon Elastic Container Service (Amazon ECS). O Application Manager aplica uma tag ao seu cluster do Amazon ECS usando o Nome do Recurso da Amazon (ARN) do cluster como valor da tag. O Application Manager fornece uma visualização do runtime do componente da rede dos recursos de computação, rede e armazenamento em um cluster.

> **Note**
>
> Você não pode gerenciar ou visualizar informações de operações sobre seus pods ou contêineres no Application Manager. Você só pode gerenciar e visualizar informações de operações sobre a infraestrutura que estiver hospedando os recursos do Amazon ECS.

Você pode executar qualquer uma das seguintes ações nesta página:

É possível executar as seguintes ações nesta página:

- Selecione Manage cluster (Gerenciar clusters) para abrir o cluster no Amazon ECS.
- Selecione View all (Visualizar tudo) para visualizar uma lista de recursos em seu cluster.
- Selecione View in CloudWatch (Visualizar no CloudWatch) para visualizar alarmes de recursos no Amazon CloudWatch.
- Selecione Manage nodes (Gerenciar nós) ou Manage Fargate profiles (Gerenciar perfis do Fargate) para visualizar esses recursos no Amazon ECS.

- Escolha um ID de recurso para exibir informações detalhadas sobre ele no console onde ele foi criado.
- Visualizar uma lista de OpsItems relacionados aos clusters.
- Visualizar um histórico de runbooks que foram executados nos clusters.

Para abrir um cluster do ECS

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Application Manager.
3. Na seção Container clusters (Clusters de contêineres), escolha ECS clusters (Clusters do ECS).
4. Escolha um cluster na lista. O Application Manager abre a guia Overview (Visão geral).

Trabalhar com o Amazon EKS no Application Manager

O Application Manager, uma ferramenta do AWS Systems Manager, integra-se ao Amazon Elastic Kubernetes Service (Amazon EKS) para fornecer informações sobre a integridade das suas infraestruturas de cluster do Amazon EKS. O Application Manager aplica uma tag ao seu cluster do Amazon EKS usando o Nome do Recurso da Amazon (ARN) do cluster como valor da tag. O Application Manager fornece uma visualização de runtime de componentes dos recursos de computação, rede e armazenamento em um cluster.

> **Note**
>
> Você não pode gerenciar ou visualizar informações de operações sobre seus pods ou contêineres do Amazon EKS no Application Manager. Você só pode gerenciar e visualizar informações de operações sobre a infraestrutura que estiver hospedando os recursos do Amazon EKS.

Você pode executar qualquer uma das seguintes ações nesta página:

É possível executar as seguintes ações nesta página:

- Selecione Manage cluster (Gerenciar clusters) para abrir o cluster no Amazon EKS.
- Selecione View all (Visualizar tudo) para visualizar uma lista de recursos em seu cluster.
- Selecione View in CloudWatch (Visualizar no CloudWatch) para visualizar alarmes de recursos no Amazon CloudWatch.

- Selecione Manage nodes (Gerenciar nós) ou Manage Fargate profiles (Gerenciar perfis do Fargate) para visualizar esses recursos no Amazon EKS.
- Escolha um ID de recurso para exibir informações detalhadas sobre ele no console onde ele foi criado.
- Visualizar uma lista de OpsItems relacionados aos clusters.
- Visualizar um histórico de runbooks que foram executados nos clusters.

Para abrir uma aplicação do EKS clusters (Clusters da aplicação).

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Application Manager.
3. Na seção Container clusters (Clusters de contêineres), escolha EKS clusters (Clusters do EKS).
4. Escolha um cluster na lista. O Application Manager abre a guia Overview (Visão geral).

Trabalhar com runbooks para clusters

É possível corrigir problemas com recursos da AWS no Application Manager, uma ferramenta do AWS Systems Manager, usando os runbooks do Systems Manager Automation. Quando você escolher Start runbook (Iniciar o runbook) em um cluster do Application Manager, o sistema exibirá uma lista filtrada de runbooks com base no tipo de recursos do cluster. Quando você escolhe o runbook que deseja iniciar, o Systems Manager abre a página Execute automation document (Executar o documento de automação).

Antes de começar

Antes de começar um runbook do Application Manager, faça o seguinte:

- Verifique se você tem as permissões corretas para iniciar runbooks. Para ter mais informações, consulte Configurar a automação.
- Examine a documentação de procedimentos do Automation sobre como iniciar runbooks. Para ter mais informações, consulte Executar uma operação automatizada com tecnologia do Systems Manager Automation.
- Se você pretende iniciar runbooks em vários recursos ao mesmo tempo, revise a documentação sobre o uso de destinos e controles de taxa. Para ter mais informações, consulte Executar operações automatizadas em escala.

Para iniciar um runbook para clusters do Application Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Application Manager.
3. Na seção Container clusters (Clusters de contêineres), selecione um tipo de contêiner.
4. Selecione um cluster na lista. O Application Manager abrirá a guia Overview (Visão geral).
5. Na guia Runbooks, escolha Start runbook (Iniciar runbook). O Application Manager abrirá a página Execute automation document (Executar documento de automação) em uma nova guia. Para obter informações sobre as opções no Documento de execução da automação, consulte Executar uma operação automatizada com tecnologia do Systems Manager Automation.

AWS Systems Manager Parameter Store

O Parameter Store, uma ferramenta do AWS Systems Manager, oferece armazenamento hierárquico seguro para gerenciamento de dados de configuração e gerenciamento de segredos. Você pode armazenar dados, como senhas, strings de banco de dados, IDs de Amazon Machine Images (AMIs) e códigos de licença como valores de parâmetro. É possível armazenar valores como texto sem formatação ou dados criptografados. Você pode referenciar parâmetros do Systems Manager em seus scripts, comandos, documentos do SSM e fluxos de trabalho de configuração e automação usando o nome exclusivo especificado ao criar o parâmetro. Para começar a usar o Parameter Store, abra o Systems Manager console (Console do gerenciador de sistemas). No painel de navegação, escolha Parameter Store.

O Parameter Store também está integrado ao Secrets Manager. Você pode recuperar segredos do Secrets Manager quando estiver usando outros Serviços da AWS que já oferecem suporte às referências dos parâmetros do Parameter Store. Para obter mais informações, consulte Fazer referência a segredos do AWS Secrets Manager em parâmetros do Parameter Store.

> **Note**
>
> Para implementar ciclos de vida de rotação de senha, useAWS Secrets Manager. Você pode alternar, gerenciar e recuperar credenciais de banco de dados, chaves de API e outros segredos durante seu ciclo de vida, usando o Secrets Manager. Para obter mais informações, consulte O que é o AWS Secrets Manager? no Guia do usuário do AWS Secrets Manager.

Como o Parameter Store beneficia minha organização?

Parameter Store oferece estes benefícios:

- Use um serviço de gerenciamento escalável de segredos hospedados sem servidores para gerenciar.
- Melhora sua postura de segurança, separando dados e código.
- Armazena dados de configuração e strings criptografadas em hierarquias e versões de trilha.
- Controla e audita o acesso em níveis específicos.
- Armazena parâmetros de forma confiável porque o Parameter Store é hospedado em várias zonas de disponibilidade em uma Região da AWS.

Quem deve usar o Parameter Store?

- Qualque cliente da AWS que quiser ter uma maneira centralizada de gerenciar os dados de configuração.
- Desenvolvedores de software que desejam armazenar logins diferentes e fluxos de referência.
- Administradores que desejam receber notificações quando seus segredos e senhas forem ou não alterados.

Quais são os recursos do Parameter Store?

- Notificação de alterações

  Configure notificações de alteração e acione ações automatizadas para ambos os parâmetros e políticas de parâmetro. Para obter mais informações, consulte Configurar notificações ou acionar ações com base nos eventos do Parameter Store.

- Organizar parâmetros

  É possível atribuir uma tag aos parâmetros para ajudar você a identificar rapidamente um ou mais parâmetros de acordo com as tags que tiver atribuído a eles. Por exemplo, você pode marcar parâmetros para ambientes ou departamentos específicos.

- Versões do rótulo

  Você pode associar um alias para versões do seu parâmetro criando rótulos. Um rótulo pode ajudar você a lembrar-se do objetivo de uma versão de parâmetro quando houver várias versões.

- Validação de dados

  Você pode criar parâmetros que apontam para uma instância do Amazon Elastic Compute Cloud (Amazon EC2) e o Parameter Store valida esses parâmetros para garantir que ele faz referência ao tipo de recurso esperado, que o recurso existe e que o cliente tem permissão para usar o recurso. Por exemplo, você pode criar um parâmetro com o ID da Amazon Machine Image (AMI) como um valor com tipo de dados `aws:ec2:image`, e o Parameter Store executará uma operação de validação assíncrona para garantir que o valor do parâmetro atenda aos requisitos de formatação para um ID da AMI, e que o AMI esteja disponível na sua Conta da AWS.

- Secretos de referência

  O Parameter Store está integrado com o AWS Secrets Manager para que você possa recuperar segredos do Secrets Manager ao usar outros Serviços da AWS que já oferecem suporte a referências a parâmetros do Parameter Store.

- Compartilhe parâmetros com outras contas

  Opcionalmente, você pode centralizar os dados de configuração em uma única Conta da AWS e compartilhar parâmetros com outras contas que precisam acessá-los.

- Acessível de outros Serviços da AWS

  Você pode usar os parâmetros da Parameter Store com outras ferramentas do Systems Manager e Serviços da AWS para recuperar segredos e dados de configuração de um armazenamento central. Os parâmetros funcionam com ferramentas do Systems Manager, como Run Command, Automation e State Manager, ferramentas do AWS Systems Manager. Você também pode fazer referência a parâmetros em vários outros Serviços da AWS, incluindo os seguintes:

  - Amazon Elastic Compute Cloud (Amazon EC2)
  - Amazon Elastic Container Service (Amazon ECS)
  - AWS Secrets Manager
  - AWS Lambda
  - AWS CloudFormation
  - AWS CodeBuild
  - AWS CodePipeline
  - AWS CodeDeploy

- Integrar com outros Serviços da AWS

Configure a integração com os seguintes Serviços da AWS para criptografia, notificação, monitoramento e auditoria:

- AWS Key Management Service (AWS KMS)
- Amazon Simple Notification Service (Amazon SNS)
- Amazon CloudWatch: pbter mais informações, consulte Configurar regras do EventBridge para parâmetros e políticas de parâmetros
- Amazon EventBridge: para obter mais informações, consulte Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS e Referência: padrões e tipos de eventos do Amazon EventBridge para o Systems Manager.
- AWS CloudTrail: para obter mais informações, consulte Registrar em log chamadas de API do AWS Systems Manager com o AWS CloudTrail.

O que é um parâmetro ?

Um parâmetro da Parameter Store é qualquer parte de dados que é salva no Parameter Store, como um bloco de texto, uma lista de nomes, uma senha, um ID de AMI, uma chave de licença e assim por diante. Você pode fazer referência a esses dados de forma centralizada e segura em seus scripts, comandos e documentos do SSM.

Ao referenciar um parâmetro, você especifica o nome desse parâmetro usando a seguinte convenção.

{{ssm:*parameter-name*}}

> **Note**
>
> Os parâmetros não podem ser referenciados ou aninhados nos valores de outros parâmetros. Não é possível incluir {{}} ou {{ssm:*parameter-name*}} em um valor de parâmetro.

O Parameter Store fornece suporte a três tipos de parâmetros: `String`, `StringList` e `SecureString`.

Com uma exceção, ao criar ou atualizar um parâmetro, você insere o valor do parâmetro como texto sem formatação, e o Parameter Store não executa nenhuma validação no texto inserido. No

entanto, para parâmetros `String`, você pode especificar o tipo de dados como `aws:ec2:image`, e o Parameter Store valida se o valor inserido é o formato adequado para uma AMI do Amazon EC2. por exemplo, `ami-12345abcdeEXAMPLE`.

Tipo de parâmetro: String

Por padrão, o valor de um parâmetro `String` consiste em qualquer bloco de texto inserido. Por exemplo:

- `abc123`
- `Example Corp`
- `<img src="images/bannerImage1.png"/>`

Tipo de parâmetro: StringList

Os valores de parâmetros `StringList` contêm uma lista de valores separada por vírgulas, conforme mostrado nos exemplos a seguir.

```
Monday,Wednesday,Friday
```

```
CSV,TSV,CLF,ELF,JSON
```

Tipo de parâmetro: SecureString

O valor de um parâmetro `SecureString` é qualquer dado sensível que precisa ser armazenado e referenciado com segurança. Se você tem dados que não deseja que os usuários alterem ou consultem em texto sem formatação, como senhas ou chaves de licença, crie esses parâmetros usando o tipo de dados `SecureString`.

> **Important**
>
> Não armazene dados confidenciais em um parâmetro `String` ou `StringList`. Para todos os dados confidenciais que devem permanecer criptografados, use somente o tipo de parâmetro `SecureString`.
> Para obter mais informações, consulte Criar um parâmetro SecureString usando a AWS CLI.

Recomendamos usar parâmetros do `SecureString` nos seguintes cenários:

- Você deseja usar dados/parâmetros em todos os Serviços da AWS sem expor os valores como texto sem formatação em comandos, funções, logs de agentes ou logs do CloudTrail.
- Você deseja controlar quem tem acesso a dados confidenciais.
- Você deseja fazer uma auditoria quando dados confidenciais forem acessados (CloudTrail).
- Você deseja criptografar seus dados confidenciais e trazer suas próprias chaves de criptografia para gerenciar o acesso.

> **Important**
>
> Somente o valor de um parâmetro `SecureString` é criptografado. O nome do parâmetro, a descrição e outras propriedades não são criptografados.

O tipo de parâmetro `SecureString` pode ser usado para dados textuais que você deseja criptografar, como senhas, segredos de aplicações, dados de configuração confidenciais ou outros tipos de dados que você precisa proteger. Os dados do `SecureString` são criptografados e descriptografados usando uma chave AWS KMS. Você pode usar uma chave KMS padrão fornecida pela AWS ou criar e usar sua própria AWS KMS key. (Use seu próprio AWS KMS key se você quiser restringir o acesso do usuário aos parâmetros `SecureString`. Para obter mais informações, consulte Permissões do IAM para usar chaves padrão da AWS e chaves gerenciadas pelo cliente).

Você também pode usar parâmetros de `SecureString` com outros Serviços da AWS. No exemplo a seguir, a função do Lambda recupera um parâmetro `SecureString` usando a API GetParameters.

```
import json
import boto3
ssm = boto3.client('ssm', 'us-east-2')
def get_parameters():
    response = ssm.get_parameters(
        Names=['LambdaSecureString'],WithDecryption=True
    )
    for parameter in response['Parameters']:
        return parameter['Value']

def lambda_handler(event, context):
    value = get_parameters()
    print("value1 = " + value)
```

```
    return value  # Echo back the first key value
```

Criptografia de definição de preço do AWS KMS

Se você escolher o tipo de parâmetro `SecureString` ao criar seu parâmetro, o Systems Manager usará o AWS KMS para criptografar o valor do parâmetro.

> **Important**
>
> O Parameter Store só oferece suporte a chaves de criptografia simétricas KMS. Não é possível usar uma chave de criptografia KMS assimétrica para criptografar os parâmetros. Para obter ajuda para determinar se uma KMS é simétrica ou assimétrica, consulte Identificar KMSs simétricas e assimétricas no Manual do desenvolvedor do AWS Key Management Service.

Não há cobrança do Parameter Store para criar um parâmetro `SecureString`, mas as cobranças pelo uso da criptografia do AWS KMS são aplicáveis. Para obter mais informações, consulte Definição de preço do AWS Key Management Service.

Para obter mais informações sobre Chaves gerenciadas pela AWS e chaves gerenciadas pelo cliente, consulte Conceitos do AWS Key Management Service, no Guia do desenvolvedor do AWS Key Management Service. Para obter mais informações sobre a criptografia do Parameter Store e do AWS KMS, consulte Como o AWS Systems Manager usa o Parameter StoreAWS KMS.

> **Note**
>
> Para visualizar uma Chave gerenciada pela AWS, use a operação `DescribeKey` do AWS KMS. Este exemplo de AWS Command Line Interface (AWS CLI) usa `DescribeKey` para visualizar uma Chave gerenciada pela AWS.
>
> ```
> aws kms describe-key --key-id alias/aws/ssm
> ```

Mais informações

- Criar um parâmetro SecureString no Parameter Store e integrar um nó a um domínio (PowerShell)

- Use o Parameter Store para acessar segredos e dados de configuração com segurança no CodeDeploy
- Artigos interessantes sobre o Parameter Store do Amazon EC2 Systems Manager

Limites de tamanhos de parâmetros

O Parameter Store tem limites de tamanho diferentes para os valores dos parâmetros dependendo do nível de parâmetros utilizado:

- Parâmetros padrão: tamanho máximo do valor de 4 KB
- Parâmetros avançados: tamanho máximo do valor de 8 KB

Se precisar armazenar valores de parâmetros maiores que 4 KB, será necessário usar o nível de parâmetros avançados. Os parâmetros avançados fornecem recursos adicionais, mas geram cobranças em sua conta da AWS. Para obter mais informações sobre níveis de parâmetros e seus recursos, consulte Gerenciar camadas de parâmetros.

Para obter uma lista completa de cotas e limites do Parameter Store, consulte Endpoints e cotas do AWS Systems Manager na Referência geral da AWS.

Configurar o Parameter Store

Para configurar parâmetros no Parameter Store, uma ferramenta do AWS Systems Manager, configure primeiro as políticas do AWS Identity and Access Management (IAM) que fornecem aos usuários em sua conta as permissões para executar as ações especificadas.

Esta seção inclui informações sobre como configurar manualmente essas políticas usando o console do IAM, e como atribuí-las a usuários e grupos de usuários. Você também pode criar e atribuir políticas para controlar quais ações de parâmetro podem ser executadas em um nó gerenciado.

Esta seção também inclui informações sobre como criar regras do Amazon EventBridge que permitem que você receba notificações sobre alterações nos parâmetros do Systems Manager. Você também pode usar as regras do EventBridge para acionar outras ações na AWS com base nas alterações da Parameter Store.

Conteúdo

- Restringir o acesso a parâmetros do Parameter Store usando políticas do IAM
- Gerenciar camadas de parâmetros

- Aumentar ou redefinir o throughput do Parameter Store
- Configurar notificações ou acionar ações com base nos eventos do Parameter Store

Restringir o acesso a parâmetros do Parameter Store usando políticas do IAM

Restrinja o acesso aos parâmetros AWS Systems Manager usando o AWS Identity and Access Management (IAM). Mais especificamente, você cria políticas do IAM que restringem o acesso às seguintes operações de API:

- DeleteParameter
- DeleteParameters
- DescribeParameters
- GetParameter
- GetParameters
- GetParameterHistory
- GetParametersByPath
- PutParameter

Ao usar políticas do IAM para restringir o acesso a parâmetros do Systems Manager, recomendamos a criação e o uso de políticas restritivas do IAM. Por exemplo, a seguinte política permite que um usuário chame as operações de API `DescribeParameters` e `GetParameters` para um conjunto limitado de recursos. Isso significa que o usuário pode obter informações e usar todos os parâmetros que começam com `prod-*`.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeParameters"
            ],
            "Resource": "*"
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameters"
            ],
            "Resource": "arn:aws:ssm:us-east-2:123456789012:parameter/prod-*"
        }
    ]
}
```

> **Important**
>
> Se um usuário tiver acesso a um caminho, o usuário poderá acessar todos os níveis desse caminho. Por exemplo, se um usuário tiver permissão para acessar um caminho `/a`, ele também pode acessar `/a/b`. Mesmo se o acesso de um usuário tiver sido explicitamente negado no IAM para o parâmetro `/a/b`, ele ainda poderá chamar a operação de API `GetParametersByPath` recursivamente para `/a` e visualizar `/a/b`.

Para administradores confiáveis, é possível fornecer acesso a todas as operações de API de parâmetros do Systems Manager usando uma política semelhante ao exemplo a seguir. Esta política fornece ao usuário o acesso total a todos os parâmetros de produção que começam com `dbserver-prod-*`.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter",
                "ssm:DeleteParameter",
                "ssm:GetParameterHistory",
                "ssm:GetParametersByPath",
                "ssm:GetParameters",
                "ssm:GetParameter",
```

```
                "ssm:DeleteParameters"
            ],
            "Resource": "arn:aws:ssm:us-east-2:111122223333:parameter/dbserver-
prod-*"
        },
        {
            "Effect": "Allow",
            "Action": "ssm:DescribeParameters",
            "Resource": "*"
        }
    ]
}
```

Negar permissões

Cada API é exclusiva e tem operações e permissões distintas que você pode permitir ou negar individualmente. Uma negação explícita em qualquer política substitui a permissão.

> **Note**
>
> O valorAWS Key Management Service(AWS KMS) tem`Decrypt`permissão para todos os principais do IAM dentro doConta da AWS. Se você quiser ter níveis de acesso diferentes ao`SecureString`Na conta, não recomendamos que você use a chave padrão.

Se você quiser que todas as operações de API que recuperam valores de parâmetro tenham o mesmo comportamento, então você pode usar um padrão como`GetParameter*`em uma política. O exemplo a seguir mostra como negar`GetParameter`,`GetParameters`,`GetParameterHistory`, e`GetParametersByPath`para todos os parâmetros que começam com`prod-*`.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "ssm:GetParameter*"
            ],
```

```
            "Resource": "arn:aws:ssm:us-east-2:123456789012:parameter/prod-*"
        }
    ]
}
```

O exemplo a seguir mostra como negar alguns comandos enquanto permite que o usuário execute outros comandos em todos os parâmetros que começam com `prod-*`.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "ssm:PutParameter",
                "ssm:DeleteParameter",
                "ssm:DeleteParameters",
                "ssm:DescribeParameters"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParametersByPath",
                "ssm:GetParameters",
                "ssm:GetParameter",
                "ssm:GetParameterHistory"
            ],
            "Resource": "arn:aws:ssm:us-east-2:111122223333:parameter/prod-*"
        }
    ]
}
```

> **Note**
>
> O histórico de parâmetros inclui todas as versões de parâmetros, incluindo a atual. Portanto, se um usuário tiver permissão negada para `GetParameter`, `GetParameters`,

eGetParameterByPathmas é permitido permissão paraGetParameterHistory, eles podem ver o parâmetro atual, incluindoSecureStringParâmetros, usandoGetParameterHistory.

Permitir que apenas parâmetros específicos sejam executados em nós

Você pode controlar o acesso para que os nós gerenciados só possam executar os parâmetros especificados.

Se você escolher o tipo de parâmetro `SecureString` ao criar o parâmetro, o Systems Manager usará o AWS KMS para criptografar o valor do parâmetro. O AWS KMS criptografa o valor usando uma Chave gerenciada pela AWS ou uma chave gerenciada pelo cliente. Para obter mais informações sobre o AWS KMS e o AWS KMS key, consulte o Guia do desenvolvedor do AWS Key Management Service.

Você pode visualizar a Chave gerenciada pela AWS executando o comando a seguir na AWS CLI.

```
aws kms describe-key --key-id alias/aws/ssm
```

O exemplo a seguir permite que os nós obtenham um valor de parâmetro somente para parâmetros que começam com `prod-`. Se o parâmetro for um parâmetro `SecureString`, o nó descriptografará a string usando o AWS KMS.

> **Note**
>
> As políticas de instâncias, como no exemplo a seguir, são atribuídas à função de instância no IAM. Para obter mais informações sobre como configurar o acesso a recursos do Systems Manager, incluindo como atribuir políticas a usuários e instâncias, consulte Gerenciar instâncias do EC2 com o Systems Manager.

JSON

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

```
            "Action": [
                "ssm:GetParameters"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-2:111122223333:parameter/prod-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-2:111122223333:key/4914ec06-e888-4ea5-
a371-5b88eEXAMPLE"
            ]
        }
    ]
}
```

Permissões do IAM para usar chaves padrão da AWS e chaves gerenciadas pelo cliente

Os parâmetros Parameter Store `SecureString` são criptografados e descriptografados usando as chaves do AWS KMS. É possível optar por criptografar os parâmetros `SecureString` usando uma AWS KMS key ou a chave KMS padrão fornecida pela AWS.

Ao usar uma chave gerenciada pelo cliente, a política do IAM que concede a um usuário acesso a um parâmetro ou um caminho de parâmetro deve fornecer permissões `kms:Encrypt` explícitas para a chave. Por exemplo, a política a seguir permite que um usuário crie, atualize e visualize parâmetros `SecureString` que começam com `prod-` na Região da AWS e Conta da AWS especificadas.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter",
                "ssm:GetParameter",
```

```
                "ssm:GetParameters"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-2:111122223333:parameter/prod-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:Encrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": [
                "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-12345EXAMPLE"
            ]
        }
    ]
}
```

[1]A permissão `kms:GenerateDataKey` é necessária para criar parâmetros avançados criptografados usando a chave específica gerenciada pelo cliente.

Por outro lado, todos os usuários da conta do cliente têm acesso à chave padrão gerenciada da AWS. Se você usar essa chave padrão para criptografar parâmetros `SecureString` e não quiser que os usuários trabalhem com parâmetros `SecureString`, suas políticas do IAM devem negar explicitamente o acesso à chave padrão, conforme demonstrado no exemplo de política a seguir.

> **Note**
>
> É possível localizar o nome do recurso da Amazon (ARN) da chave padrão no console do AWS KMS na página AWS managed keys (chaves gerenciadas). A chave padrão é aquela identificada com `aws/ssm` na coluna Alias.

JSON

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

```
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": [
                "arn:aws:kms:us-east-2:111122223333:key/abcd1234-ab12-cd34-ef56-
abcdeEXAMPLE"
            ]
        }
    ]
}
```

Se você precisar de controle de acesso minucioso sobre os parâmetros `SecureString` em sua conta, será necessário usar uma CMK gerenciada pelo cliente para proteger e restringir o acesso a esses parâmetros. Também recomendamos o uso do AWS CloudTrail para monitorar atividades de parâmetros `SecureString`.

Para obter mais informações, consulte os tópicos a seguir.

- Lógica da avaliação de políticas no Guia do usuário do IAM
- Usar políticas de chaves no AWS KMS no Guia do desenvolvedor do AWS Key Management Service
- Visualizar eventos com o histórico de eventos do CloudTrail no Guia do usuário do AWS CloudTrail

Gerenciar camadas de parâmetros

O Parameter Store, uma ferramenta do AWS Systems Manager, inclui parâmetros padrão e parâmetros avançados. Você configura parâmetros individualmente para usar o nível de parâmetros padrão (o padrão) ou o nível de parâmetros avançados.

Você pode transformar um parâmetro padrão em um parâmetro avançado a qualquer momento, mas não pode reverter um parâmetro avançado para um parâmetro padrão. Reverter um parâmetro avançado para um parâmetro padrão resultaria na perda de dados, pois o sistema truncaria o tamanho do parâmetro de 8 KB para 4 KB. A operação de reversão também removeria todas as políticas anexadas ao parâmetro. Além disso, parâmetros avançados usam um formato de criptografia diferente dos parâmetros padrão. Para obter mais informações, consulte Como o AWS

Systems Manager Parameter Store usa o AWS KMS no Guia do desenvolvedor do AWS Key Management Service.

Se você não precisar mais de um parâmetro avançado, ou se não quiser mais ser cobrado por um parâmetro avançado, deverá excluí-lo e recriá-lo como um novo parâmetro padrão.

A tabela a seguir descreve as diferenças entre os níveis.

	Padrão	Advanced (Avançado)
O número total de parâmetros permitidos (por Conta da AWS e Região da AWS)	10.000	100.000
Tamanho máximo de um valor de parâmetro.	4 KB	8 KB
Políticas de parâmetros disponíveis	Não	Sim Para obter mais informaçõ es, consulte Atribuir políticas de parâmetros no Parameter Store.
Custo	Sem custos adicionais	Cobranças são aplicáveis (Para obter mais informaçõ es, consulte Preços do AWS Systems Manager para Parameter Store.)

Tópicos

- Especificar um nível de parâmetro padrão
- Alterar um parâmetro padrão para um parâmetro avançado

Especificar um nível de parâmetro padrão

Em solicitações para criar ou atualizar um parâmetro (ou seja, a ação `PutParameter`), você pode especificar o nível de parâmetro a ser usado na solicitação. Veja a seguir um exemplo de como usar a AWS Command Line Interface (AWS CLI).

Linux & macOS

```
aws ssm put-parameter \
    --name "default-ami" \
    --type "String" \
    --value "t2.micro" \
    --tier "Standard"
```

Windows

```
aws ssm put-parameter ^
    --name "default-ami" ^
    --type "String" ^
    --value "t2.micro" ^
    --tier "Standard"
```

Sempre que você especifica um nível na solicitação, o Parameter Store cria ou atualiza o parâmetro de acordo com a solicitação. No entanto, se você não especificar explicitamente um nível em uma solicitação, a configuração do nível padrão do Parameter Store determinará o nível em que o parâmetro será criado.

O nível padrão quando você começa a usar o Parameter Store é o nível de parâmetro padrão. Se você usar o nível de parâmetro avançado, poderá especificar um dos seguintes como o padrão:

- Advanced (Avançado): com essa opção, o repositório de parâmetros avalia todas as solicitações como parâmetros avançados.
- Intelligent-Tiering: com essa opção, o Parameter Store avalia cada solicitação para determinar se o parâmetro é padrão ou avançado.

  Se a solicitação não incluir opções que exijam um parâmetro avançado, o parâmetro será criado no nível de parâmetro padrão. Se uma ou mais opções que exijam um parâmetro avançado forem incluídas na solicitação, o Parameter Store criará um parâmetro no nível de parâmetro avançado.

Benefícios do nível Intelligent-Tiering

Os seguintes motivos podem fazer você escolher o Intelligent-Tiering como o nível padrão.

Controle de custos - o Intelligent-Tiering ajuda a controlar os custos relacionados a parâmetros criando sempre parâmetros padrão, a menos que um parâmetro avançado seja absolutamente necessário.

Atualização automática para o nível de parâmetro avançado - quando você faz uma alteração no código que requer a atualização de um parâmetro padrão para um parâmetro avançado, o Intelligent-Tiering lida com a conversão para você. Você não precisa alterar o código para lidar com a atualização.

Veja alguns exemplos de atualização automática:

- Os modelos do AWS CloudFormation provisionam vários parâmetros quando são executados. Quando esse processo faz com que você atinja a cota de 10.000 parâmetros no nível de parâmetro padrão, o Intelligent-Tiering atualiza automaticamente você para o nível de parâmetro avançado, e os processos do AWS CloudFormation não são interrompidos.
- Você armazena um valor de certificado em um parâmetro, alterna o valor do certificado regularmente, e o conteúdo é inferior à cota de 4 KB do nível de parâmetro padrão. Se o valor de um certificado de substituição exceder 4 KB, o Intelligent-Tiering atualizará automaticamente o parâmetro para o nível de parâmetro avançado.
- Você deseja associar vários parâmetros padrão existentes a uma política de parâmetros, que requer o nível de parâmetro avançado. Em vez de você precisar incluir a opção `--tier Advanced` em todas as chamadas para atualizar os parâmetros, o Intelligent-Tiering atualiza automaticamente os parâmetros para o nível de parâmetro avançado. A opção Intelligent-Tiering atualiza os parâmetros de padrão para avançados sempre que os critérios do nível de parâmetro avançado forem introduzidos.

As opções que exigem um parâmetro avançado incluem as seguintes:

- O tamanho do conteúdo do parâmetro é superior a 4 KB.
- O parâmetro usa uma política de parâmetros.
- Já existem mais de 10.000 parâmetros em sua Conta da AWS na Região da AWS atual.

Opções de nível padrão

As opções de nível que você pode especificar como padrão incluem o seguinte.

- Padrão: o nível de parâmetro padrão é o nível padrão quando você começa a usar o Parameter Store. Usando o nível de parâmetro padrão, você pode criar 10.000 parâmetros para cada Região da AWS em uma Conta da AWS. O tamanho do conteúdo de cada parâmetro pode ser igual a um máximo de 4 KB. Os parâmetros padrão não são compatíveis com políticas de parâmetros. Não há custo adicional para usar o nível de parâmetro padrão. A escolha de Standard (Padrão) como o nível padrão significa que o Parameter Store sempre tentará criar um parâmetro padrão para solicitações que não especificam um nível.
- Advanced (Avançado): use o nível de parâmetro avançado para criar no máximo 100.000 parâmetros para cada Região da AWS em uma Conta da AWS. O tamanho do conteúdo de cada parâmetro pode ser igual a um máximo de 8 KB. Os parâmetros avançados oferecem suporte a políticas de parâmetros. Para compartilhar um parâmetro, ele deve estar no nível de parâmetros avançados. Há uma cobrança para o uso do nível de parâmetro avançado. (Para obter mais informações, consulte Preços do AWS Systems Manager para Parameter Store.) A escolha de Advanced (Avançado) como o nível padrão significa que o Parameter Store sempre tentará criar um parâmetro avançado para solicitações que não especificam um nível.

  > **Note**
  >
  > Ao escolher o nível de parâmetro avançado, autorize explicitamente a AWS a cobrar sua conta por qualquer parâmetro avançado que você criar.

- Intelligent-Tiering a opção Intelligent-Tiering permite que o Parameter Store determine se o nível de parâmetro padrão ou o nível de parâmetro avançado deve ser usado com base no conteúdo da solicitação. Por exemplo, se você executar um comando para criar um parâmetro com conteúdo inferior a 4 KB, houver menos de 10.000 parâmetros na Região da AWS atual em sua Conta da AWS, e você não especificar uma política de parâmetro, um parâmetro padrão será criado. Se você executar um comando para criar um parâmetro com mais de 4 KB de conteúdo, já tiver mais de 10.000 parâmetros na Região da AWS atual de sua Conta da AWS ou especificar uma política de parâmetro, um parâmetro avançado será criado.

  > **Note**
  >
  > Ao escolher Intelligent-Tiering, você deve autorizar explicitamente a AWS a cobrar sua conta por todos os parâmetros avançados criados.

Você pode alterar a configuração de nível padrão do Parameter Store a qualquer momento.

Configurar permissões para especificar um nível padrão do Parameter Store

Verifique se você tem permissão no AWS Identity and Access Management (IAM) para alterar o nível de parâmetro padrão no Parameter Store executando uma das seguintes ações:

- Certifique-se de anexar a política `AdministratorAccess` à sua entidade do IAM (como usuário, grupo ou perfil).
- Certifique-se de ter permissão para alterar a configuração de nível padrão usando as seguintes ações da API:
  - GetServiceSetting
  - UpdateServiceSetting
  - ResetServiceSetting

Conceda as permissões a seguir à entidade do IAM para permitir que um usuário visualize e altere a configuração de nível padrão para parâmetros em uma Região da AWS específica em uma Conta da AWS.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetServiceSetting"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:servicesetting/ssm/
parameter-store/default-parameter-tier"
        }
    ]
```

```
}
```

Os administradores poderão especificar a permissão de somente leitura ao atribuir as permissões a seguir.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetServiceSetting"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "ssm:ResetServiceSetting",
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "*"
        }
    ]
}
```

Para conceder acesso, adicione as permissões aos seus usuários, grupos ou perfis:

- Usuários e grupos no AWS IAM Identity Center:

  Crie um conjunto de permissões. Siga as instruções em Criação de um conjunto de permissões no Guia do usuário do AWS IAM Identity Center.
- Usuários gerenciados no IAM com provedor de identidades:

  Crie um perfil para a federação de identidades. Siga as instruções em Criando um perfil para um provedor de identidades de terceiros (federação) no Guia do Usuário do IAM.
- Usuários do IAM:

- Crie um perfil que seu usuário possa assumir. Siga as instruções em Criação de um perfil para um usuário do IAM no Guia do usuário do IAM.
- (Não recomendado) Vincule uma política diretamente a um usuário ou adicione um usuário a um grupo de usuários. Siga as instruções em Adição de permissões a um usuário (console) no Guia do usuário do IAM.

Especificar ou alterar o nível padrão do Parameter Store usando o console

O seguinte procedimento mostra como usar o console do Systems Manager para especificar ou alterar o nível de parâmetro padrão para a Conta da AWS e a Região da AWS atuais.

> **Tip**
>
> Se você ainda não criou um parâmetro, será possível usar a AWS Command Line Interface (AWS CLI) ou o AWS Tools for Windows PowerShell para alterar a camada de parâmetro padrão. Para obter informações, consulte Especificar ou alterar o nível padrão do Parameter Store usando a AWS CLI e Especificar ou alterar o nível padrão do Parameter Store (PowerShell).

Para especificar ou alterar o nível padrão do Parameter Store

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Parameter Store.
3. Escolha a guia Configurações.
4. Selecione Change default tier (Alterar a camada padrão).
5. Escolha uma das seguintes opções.
   - Padrão
   - Advanced (Avançado)
   - Intelligent-Tiering

   Para obter informações sobre essas opções, consulte Especificar um nível de parâmetro padrão.
6. Revise a mensagem e escolha Confirm (Confirmar).

Se você quiser alterar a configuração de nível padrão posteriormente, repita esse procedimento e especifique outra opção de nível padrão.

Especificar ou alterar o nível padrão do Parameter Store usando a AWS CLI

O seguinte procedimento mostra como usar a AWS CLI (Conta da AWS) para alterar a configuração do nível de parâmetro padrão para a conta e a região atuais da Região da AWS.

Para especificar ou alterar o nível padrão do Parameter Store usando a AWS CLI

1. Abra a AWS CLI e execute o seguinte comando para alterar a configuração do nível de parâmetro padrão para uma Região da AWS específica em uma Conta da AWS.

   ```
   aws ssm update-service-setting --setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/parameter-store/default-parameter-tier --setting-value tier-option
   ```

   A *região* representa o identificador da região para uma região da Região da AWS compatível com o AWS Systems Manager, como `us-east-2` para a região Leste dos EUA (Ohio). Para ver uma lista dos valores de *região* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

   Os valores de *tier-option* incluem `Standard`, `Advanced` e `Intelligent-Tiering`. Para obter informações sobre essas opções, consulte Especificar um nível de parâmetro padrão.

   Não haverá saída se o comando for bem-sucedido.

2. Execute o seguinte comando para visualizar as configurações de serviço de nível de parâmetro padrão atual para o Parameter Store na Conta da AWS e Região da AWS atuais.

   ```
   aws ssm get-service-setting --setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/parameter-store/default-parameter-tier
   ```

   O sistema retorna informações semelhantes às seguintes.

   ```
   {
       "ServiceSetting": {
           "SettingId": "/ssm/parameter-store/default-parameter-tier",
           "SettingValue": "Advanced",
           "LastModifiedDate": 1556551683.923,
   ```

```
        "LastModifiedUser": "arn:aws:sts::123456789012:assumed-role/Administrator/
Jasper",
        "ARN": "arn:aws:ssm:us-east-2:123456789012:servicesetting/ssm/parameter-
store/default-parameter-tier",
        "Status": "Customized"
    }
}
```

Se você quiser alterar a configuração do nível padrão novamente, repita esse procedimento e especifique outra opção de `SettingValue`.

Especificar ou alterar o nível padrão do Parameter Store (PowerShell)

O seguinte procedimento mostra como usar o Tools for Windows PowerShell para alterar a configuração do nível de parâmetro padrão para uma Região da AWS específica em uma conta da Amazon Web Services.

Para especificar ou alterar o nível padrão do Parameter Store usando o PowerShell

1. Altere a camada padrão do Parameter Store na Conta da AWS e Região da AWS usando o Ferramentas da AWS para PowerShell (Tools for PowerShell).

   ```
   Update-SSMServiceSetting -SettingId "arn:aws:ssm:region:account-id:servicesetting/
   ssm/parameter-store/default-parameter-tier" -SettingValue "tier-option" -
   Region region
   ```

   A *`região`* representa o identificador da região para uma região da Região da AWS compatível com o AWS Systems Manager, como `us-east-2` para a região Leste dos EUA (Ohio). Para ver uma lista dos valores de *`região`* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

   Os valores de *`tier-option`* incluem `Standard`, `Advanced` e `Intelligent-Tiering`. Para obter informações sobre essas opções, consulte Especificar um nível de parâmetro padrão.

   Não haverá saída se o comando for bem-sucedido.

2. Execute o seguinte comando para visualizar as configurações de serviço de nível de parâmetro padrão atual para o Parameter Store na Conta da AWS e Região da AWS atuais.

```
Get-SSMServiceSetting -SettingId "arn:aws:ssm:region:account-id:servicesetting/ssm/
parameter-store/default-parameter-tier" -Region region
```

A *região* representa o identificador da região para uma região da Região da AWS compatível com o AWS Systems Manager, como `us-east-2` para a região Leste dos EUA (Ohio). Para ver uma lista dos valores de *região* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

O sistema retorna informações semelhantes às seguintes.

```
ARN : arn:aws:ssm:us-east-2:123456789012:servicesetting/ssm/parameter-store/
default-parameter-tier
LastModifiedDate : 4/29/2019 3:35:44 PM
LastModifiedUser : arn:aws:sts::123456789012:assumed-role/Administrator/Jasper
SettingId        : /ssm/parameter-store/default-parameter-tier
SettingValue     : Advanced
Status           : Customized
```

Se você quiser alterar a configuração do nível padrão novamente, repita esse procedimento e especifique outra opção de `SettingValue`.

Alterar um parâmetro padrão para um parâmetro avançado

Use o procedimento a seguir para transformar um parâmetro padrão em um parâmetro avançado. Para obter informações sobre como criar um novo parâmetro avançado, consulte Criar parâmetros do Parameter Store no Systems Manager.

Para transformar um parâmetro padrão em um parâmetro avançado

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Parameter Store.
3. Escolha um parâmetro e depois Edit (Editar).
4. Em Description (Descrição), insira informações sobre esse parâmetro.
5. Escolha Advanced (Avançado).
6. Em Value (Valor), insira o valor desse parâmetro. Parâmetros avançados têm um limite de valor máximo de 8 KB.
7. Escolha Salvar alterações.

Aumentar ou redefinir o throughput do Parameter Store

Aumentar o throughput do Parameter Store aumenta o número máximo de transações por segundo (TPS) que o Parameter Store, uma ferramenta do AWS Systems Manager, pode processar. Uma throughput maior permite operar o Parameter Store em volumes mais altos para oferecer suporte a aplicações e cargas de trabalho que precisam de acesso simultâneo a vários parâmetros. Você pode aumentar a cota até a throughput máxima na guia Settings (Configurações).

A configuração de throughput do Parameter Store se aplica a todas as transações criadas por todos os usuários do (IAM) na Conta da AWS e na Região da AWS atuais. A configuração de throughput aplica-se a parâmetros padrão e avançados.

> **Note**
>
> Normalmente, as atualizações são imediatamente visíveis no Service Quotas. Em casos raros, pode levar até 24 horas para que uma atualização seja refletida.

Para obter mais informações sobre throughput máxim, consulte Endpoints e cotas do AWS Systems Manager.

Aumentar a cota da throughput gera cobranças na sua Conta da AWS. Para obter mais informações, consulte Preços do AWS Systems Manager.

Tópicos

- Configurar permissões para alterar o throughput do Parameter Store
- Aumentar ou redefinir o throughput usando o console
- Aumentar ou redefinir o throughput usando o AWS CLI
- Aumentar ou redefinir o throughput (PowerShell)

Configurar permissões para alterar o throughput do Parameter Store

Verifique se você tem permissão no IAM para alterar o throughput do Parameter Store de uma das seguintes maneiras:

- Certifique-se de que a política `AdministratorAccess` esteja anexada à sua entidade do IAM (usuário, grupo ou perfil).

- Certifique-se de ter permissão para alterar a configuração do serviço de throughput usando as seguintes ações de API:
  - GetServiceSetting
  - UpdateServiceSetting
  - ResetServiceSetting

Conceda as permissões a seguir à entidade do IAM para permitir que um usuário visualize e altere a configuração de throughput para parâmetros em uma Região da AWS específica em uma Conta da AWS.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetServiceSetting"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:servicesetting/ssm/
parameter-store/high-throughput-enabled"
        }
    ]
}
```

Os administradores poderão especificar a permissão de somente leitura ao atribuir as permissões a seguir.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetServiceSetting"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "ssm:ResetServiceSetting",
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "*"
        }
    ]
}
```

Para conceder acesso, adicione as permissões aos seus usuários, grupos ou perfis:

- Usuários e grupos no AWS IAM Identity Center:

  Crie um conjunto de permissões. Siga as instruções em Criação de um conjunto de permissões no Guia do usuário do AWS IAM Identity Center.
- Usuários gerenciados no IAM com provedor de identidades:

  Crie um perfil para a federação de identidades. Siga as instruções em Criando um perfil para um provedor de identidades de terceiros (federação) no Guia do Usuário do IAM.
- Usuários do IAM:
  - Crie um perfil que seu usuário possa assumir. Siga as instruções em Criação de um perfil para um usuário do IAM no Guia do usuário do IAM.
  - (Não recomendado) Vincule uma política diretamente a um usuário ou adicione um usuário a um grupo de usuários. Siga as instruções em Adição de permissões a um usuário (console) no Guia do usuário do IAM.

Aumentar ou redefinir o throughput usando o console

O seguinte procedimento mostra como usar o console do Systems Manager para aumentar o número de transações por segundo que o Parameter Store pode processar para a Conta da AWS e a Região da AWS atuais. Ele também mostra como reverter para as configurações padrão se o throughput aumentado não for mais necessário ou se você não quiser mais incorrer em cobranças.

Aumentar ou redefinir o Parameter Store de throughput usando o console

> **Tip**
>
> Se você ainda não criou um parâmetro, será possível usar a AWS Command Line Interface (AWS CLI) ou o AWS Tools for Windows PowerShell para aumentar a throughput. Para obter informações, consulte Aumentar ou redefinir o throughput usando o AWS CLI e Aumentar ou redefinir o throughput (PowerShell).

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Parameter Store.
3. Escolha a guia Configurações.
4. Para aumentar o throughput, escolha Definir limite.

   - ou -

   Para reverter para o limite padrão, escolha Redefinir limite.
5. Se você estiver aumentando o limite, faça o seguinte:

   - Marque a caixa de seleção Aceito que a alteração dessa configuração incorra em cobranças em minha Conta da AWS.
   - Escolha Set limit (Definir limite).

   - ou -

   Se você estiver redefinindo o limite para o padrão, faça o seguinte:

   - Marque a caixa de seleção Aceito que a redefinição para o limite de throughput padrão faça com que o Parameter Store processe menos transações por segundo.
   - Escolha Redefinir limite.

Aumentar ou redefinir o throughput usando o AWS CLI

O seguinte procedimento mostra como usar o AWS CLI para aumentar o número de transações por segundo que o Parameter Store pode processar para a Conta da AWS e a Região da AWS atuais. Você também pode reverter para o limite padrão.

Para aumentar a throughput do Parameter Store usando a AWS CLI

1. Abra a AWS CLI e execute o seguinte comando para aumentar as transações por segundo que o Parameter Store pode processar na Conta da AWS e Região da AWS atuais.

   ```
   aws ssm update-service-setting --setting-id arn:aws:ssm:region:account-
   id:servicesetting/ssm/parameter-store/high-throughput-enabled --setting-value true
   ```

   Não haverá saída se o comando for bem-sucedido.

2. Execute o seguinte comando para visualizar as configurações de serviço de throughput atual para o Parameter Store na Conta da AWS e Região da AWS atuais.

   ```
   aws ssm get-service-setting --setting-id arn:aws:ssm:region:account-
   id:servicesetting/ssm/parameter-store/high-throughput-enabled
   ```

   O sistema retorna informações semelhantes às seguintes:

   ```
   {
       "ServiceSetting": {
           "SettingId": "/ssm/parameter-store/high-throughput-enabled",
           "SettingValue": "true",
           "LastModifiedDate": 1556551683.923,
           "LastModifiedUser": "arn:aws:sts::123456789012:assumed-role/Administrator/
   Jasper",
           "ARN": "arn:aws:ssm:us-east-2:123456789012:servicesetting/ssm/parameter-
   store/high-throughput-enabled",
           "Status": "Customized"
       }
   }
   ```

Se não precisar mais da throughput maior, ou se não quiser mais acumular cobranças, você poderá reverter para as configurações padrão. Para reverter as configurações, execute o comando a seguir.

```
aws ssm reset-service-setting --setting-id arn:aws:ssm:region:account-
id:servicesetting/ssm/parameter-store/high-throughput-enabled
```

```
{
    "ServiceSetting": {
        "SettingId": "/ssm/parameter-store/high-throughput-enabled",
        "SettingValue": "false",
        "LastModifiedDate": 1555532818.578,
        "LastModifiedUser": "System",
        "ARN": "arn:aws:ssm:us-east-2:123456789012:servicesetting/ssm/parameter-store/
high-throughput-enabled",
        "Status": "Default"
    }
}
```

Aumentar ou redefinir o throughput (PowerShell)

O seguinte procedimento mostra como usar o Tools for Windows PowerShell para aumentar o número de transações por segundo que o Parameter Store pode processar para a Conta da AWS e Região da AWS atuais. Você também pode reverter para o limite padrão.

Para aumentar a throughput do Parameter Store usando o PowerShell

1. Aumentar throughput do Parameter Store na Conta da AWS e Região da AWS atuais usando o Ferramentas da AWS para PowerShell (Tools for PowerShell).

   ```
   Update-SSMServiceSetting -SettingId "arn:aws:ssm:region:account-id:servicesetting/
   ssm/parameter-store/high-throughput-enabled" -SettingValue "true" -Region region
   ```

   Não haverá saída se o comando for bem-sucedido.

2. Execute o seguinte comando para visualizar as configurações de serviço de throughput atual para o Parameter Store na Conta da AWS e Região da AWS atuais.

   ```
   Get-SSMServiceSetting -SettingId "arn:aws:ssm:region:account-id:servicesetting/ssm/
   parameter-store/high-throughput-enabled" -Region region
   ```

   O sistema retorna informações semelhantes às seguintes:

```
ARN              : arn:aws:ssm:us-east-2:123456789012:servicesetting/ssm/parameter-
store/high-throughput-enabled
LastModifiedDate : 4/29/2019 3:35:44 PM
LastModifiedUser : arn:aws:sts::123456789012:assumed-role/Administrator/Jasper
SettingId        : /ssm/parameter-store/high-throughput-enabled
SettingValue     : true
Status           : Customized
```

Se não precisar mais da throughput maior, ou se não quiser mais acumular cobranças, você poderá reverter para as configurações padrão. Para reverter as configurações, execute o comando a seguir.

```
Reset-SSMServiceSetting -SettingId "arn:aws:ssm:region:account-id:servicesetting/ssm/
parameter-store/high-throughput-enabled" -Region region
```

O sistema retorna informações semelhantes às seguintes:

```
ARN              : arn:aws:ssm:us-east-2:123456789012:servicesetting/ssm/parameter-
store/high-throughput-enabled
LastModifiedDate : 4/17/2019 8:26:58 PM
LastModifiedUser : System
SettingId        : /ssm/parameter-store/high-throughput-enabled
SettingValue     : false
Status           : Default
```

Configurar notificações ou acionar ações com base nos eventos do Parameter Store

Os tópicos desta seção explicam como usar o Amazon EventBridge e o Amazon Simple Notification Service (Amazon SNS) para notificá-lo sobre alterações nos parâmetros do AWS Systems Manager. Você pode criar uma regra do EventBridge para receber notificações em caso de criação, atualização ou exclusão de um parâmetro ou uma versão de rótulo de parâmetro. Os eventos são emitidos com base no melhor esforço. Você pode ser notificado sobre alterações ou status relacionados às políticas do parâmetro, como quando um parâmetro expirar, a data de expiração dele ou se ele não tiver sido alterado por um período especificado.

> **Note**
>
> Políticas de parâmetros estão disponíveis apenas para parâmetros que usam o nível avançado de parâmetros. Há cobranças aplicáveis. Para ter mais informações, consulte Atribuir políticas de parâmetros no Parameter Store e Gerenciar camadas de parâmetros.

Os tópicos a seguir também explicam como acionar outras ações em um destino para eventos específicos do parâmetros. Por exemplo, você pode executar uma função AWS Lambda para recriar um parâmetro automaticamente quando ele expira ou é excluído. Você também pode configurar uma notificação para chamar uma função do Lambda quando a senha do banco de dados for atualizada. A função do Lambda pode forçar a redefinição das conexões do banco de dados ou a reconexão com a nova senha. O EventBridge também oferece suporte ao comandos Run Command, execuções do Automation e ações em muitos outros Serviços da AWS. O Run Command e o Automation são ambos ferramentas do AWS Systems Manager. Para obter mais informações, consulte o Guia do Usuário do Amazon EventBridge.

Antes de começar

Crie todos os recursos necessários para especificar a ação de destino para a regra que você criar. Por exemplo, se a regra criada tiver como objetivo enviar uma notificação, crie primeiro um tópico do Amazon SNS. Para obter mais informações, consulte Conceitos básicos do Amazon SNS no Guia do desenvolvedor do Amazon Simple Notification Service.

Configurar regras do EventBridge para parâmetros e políticas de parâmetros

Este tópico explica o seguinte:

- Como criar uma regra do EventBridge que invoca um destino com base em eventos que ocorrem em um ou mais parâmetros em sua Conta da AWS.
- Como criar regras do EventBridge que invocam destinos com base em eventos que ocorrem em uma ou mais políticas de parâmetro na sua Conta da AWS. Ao criar um parâmetro avançado, você especifica quando um parâmetro expira, quando você recebe uma notificação antes de um parâmetro expirar e o tempo de espera antes de uma notificação ser enviada informando que um parâmetro não foi alterado. Configure uma notificação para esses eventos usando o procedimento a seguir. Configure a notificação para esses eventos usando o procedimento a seguir. Para ter mais informações, consulte Atribuir políticas de parâmetros no Parameter Store e Gerenciar camadas de parâmetros.

Para configurar o EventBridge para um parâmetro ou uma política de parâmetro do Systems Manager

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Rules (Regras) e Create rule (Criar regras).

   - ou -

   Se a página inicial do EventBridge abrir primeiro, escolha Create rule (Criar regra).
3. Insira um nome e uma descrição para a regra.

   Uma regra não pode ter o mesmo nome que outra na mesma Região e barramento de eventos.
4. Em Event bus (Barramento de eventos), escolha o barramento de eventos que deseja associar a essa regra. Se desejar que essa regra seja iniciada em eventos correspondentes provenientes da sua Conta da AWS, selecione default (padrão). Quando um AWS service (Serviço da AWS) na sua conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta.
5. Em Rule type (Tipo de regra), mantenha o valor padrão Rule with an event pattern (Regra com um padrão de evento selecionado).
6. Escolha Próximo.
7. Em Origem do evento, mantenha o valor padrão Eventos da AWS ou eventos de parceiros do EventBridge selecionado. Você pode ignorar a seção Sample event (Evento de exemplo).
8. Em Event pattern (Padrão de evento), faça o seguinte:

   - Escolha Custom patterns (JSON editor) (Padrões personalizados (editor JSON)).
   - Em Event pattern (Padrão de evento), cole um dos seguintes conteúdos na caixa, dependendo se você está criando uma regra para um parâmetro ou uma política de parâmetros:

     Parameter

```
{
    "source": [
        "aws.ssm"
    ],
    "detail-type": [
        "Parameter Store Change"
    ],
    "detail": {
```

```
        "name": [
            "parameter-1-name",
            "/parameter-2-name/level-2",
            "/parameter-3-name/level-2/level-3"
        ],
        "operation": [
            "Create",
            "Update",
            "Delete",
            "LabelParameterVersion"
        ]
    }
}
```

Parameter policy

```
{
    "source": [
        "aws.ssm"
    ],
    "detail-type": [
        "Parameter Store Policy Action"
    ],
    "detail": {
        "parameter-name": [
            "parameter-1-name",
            "/parameter-2-name/level-2",
            "/parameter-3-name/level-2/level-3"
        ],
        "policy-type": [
            "Expiration",
            "ExpirationNotification",
            "NoChangeNotification"
        ]
    }
}
```

- Modifique o conteúdo dos parâmetros e operações nos quais deseja executar uma ação, conforme mostrado nos exemplos a seguir.

Parameter

Neste exemplo, uma ação é executada quando qualquer um dos parâmetros chamados /Oncall ou /Project/Teamlead é atualizado:

```
{
    "source": [
        "aws.ssm"
    ],
    "detail-type": [
        "Parameter Store Change"
    ],
    "detail": {
        "name": [
            "/Oncall",
            "/Project/Teamlead"
        ],
        "operation": [
            "Update"
        ]
    }
}
```

Parameter policy

Neste exemplo, uma ação é executada sempre que o parâmetro chamado /OncallDuties expira e é excluído:

```
{
    "source": [
        "aws.ssm"
    ],
    "detail-type": [
        "Parameter Store Policy Action"
    ],
    "detail": {
        "parameter-name": [
            "/OncallDuties"
        ],
        "policy-type": [
            "Expiration"
        ]
```

```
    }
}
```

9. Escolha Próximo.
10. Para Target 1 (Destino 1), escolha um tipo de destino e um recurso com suporte. Por exemplo, se você escolher SNS topic (Tópico do SNS), faça uma seleção para o Topic (Tópico). Se você escolher CodePipeline, insira um ARN de pipeline para Pipeline ARN (ARN do pipeline). Forneça valores de configuração adicionais conforme necessário.

> **Tip**
>
> Selecione Add another target (Adicionar outro destino) se houver necessidade de destinos adicionais para a regra.

11. Escolha Próximo.
12. (Opcional) Insira uma ou mais tags para a regra. Para mais informações, consulte Tags Amazon EventBridge em Guia de Usuário Amazon EventBridge.
13. Escolha Next (Próximo).
14. Selecione Criar regra.

Mais informações

- Use parameter labels for easy configuration update across environments
- Tutorial: Usar o EventBridge para retransmitir eventos para AWS Systems ManagerRun Command, no Guia do usuário do Amazon EventBridge
- Tutorial: Definir automação do AWS Systems Manager com um destino do EventBridge no Guia do usuário do Amazon EventBridge

Trabalhar com o Parameter Store

Esta seção descreve como organizar e criar parâmetros de tag e como criar diferentes versões de parâmetros.

É possível usar o console do AWS Systems Manager, a AWS Command Line Interface (AWS CLI), o Ferramentas da AWS para PowerShell e os AWS SDKs para criar e trabalhar com parâmetros. Para obter mais informações sobre parâmetros, consulte O que é um parâmetro ?.

Tópicos

- Criar parâmetros do Parameter Store no Systems Manager
- Pesquisar parâmetros do Parameter Store no Systems Manager
- Atribuir políticas de parâmetros no Parameter Store
- Trabalhar com hierarquias de parâmetros no Parameter Store
- Impedir o acesso às operações da API Parameter Store
- Trabalhar com rótulos de parâmetros no Parameter Store
- Como trabalhar com versões de parâmetros no Parameter Store
- Trabalhar com parâmetros compartilhados no Parameter Store
- Trabalhar com parâmetros no Parameter Store usando comandos do Run Command
- Como usar o suporte a parâmetros nativos no Parameter Store para IDs de imagem de máquina da Amazon
- Excluir parâmetros do Parameter Store

Criar parâmetros do Parameter Store no Systems Manager

Use as informações nos tópicos a seguir para ajudar você a criar parâmetros do Systems Manager usando o console do AWS Systems Manager, a AWS Command Line Interface (AWS CLI) ou o AWS Tools for Windows PowerShell(Tools for Windows PowerShell).

As demonstrações nesta seção mostram como criar, armazenar e executar parâmetros com o Parameter Store em um ambiente de teste. Ela também demonstra como usar o Parameter Store com outras ferramentas do Systems Manager em Serviços da AWS. Para ter mais informações, consulte O que é um parâmetro ?.

Entender requisitos e restrições para nomes de parâmetros

Use as informações contidas neste tópico para ajudar você a especificar os valores válidos para os nomes de parâmetro quando você cria um parâmetro.

Essas informações complementam os detalhes no tópico PutParameter na Referência da API do AWS Systems Manager, que também fornece informações sobre os valores AllowedPattern, Description, KeyId, Overwrite, Type e Value.

Os requisitos e restrições para nomes de parâmetros incluem os seguintes:

- Diferenciação de letras maiúsculas e minúsculas: os nomes de parâmetro diferenciam maiúsculas de minúsculas.

- Espaços: nomes de parâmetro não podem incluir espaços.
- Caracteres válidos: nomes de parâmetro podem conter somente os seguintes símbolos e letras: `a-zA-Z0-9_.-`

  Além disso, o caractere de barra (/) é usado para delinear hierarquias em nomes de parâmetros. Por exemplo: `/Dev/Production/East/Project-ABC/MyParameter`
- Formato válido da AMI: quando você escolhe `aws:ec2:image` como o tipo de dados para um parâmetro `String`, o ID inserido deve ser validado para o formato de ID da AMI `ami-12345abcdeEXAMPLE`.
- Totalmente qualificado: quando você cria ou faz referência a um parâmetro em uma hierarquia, é necessário incluir um caractere de barra (/) inicial. Quando você faz referência a um parâmetro que faz parte de uma hierarquia, é necessário especificar todo o caminho da hierarquia, inclusive a barra (/) inicial.
  - Nomes de parâmetros totalmente qualificados: `MyParameter1`, `/MyParameter2`, `/Dev/Production/East/Project-ABC/MyParameter`
  - Nome do parâmetro não totalmente qualificado: `MyParameter3/L1`
- Comprimento: o comprimento máximo de um nome de parâmetro que você especifica é 1.011 caracteres. Esse número de 1.011 caracteres inclui os caracteres no ARN que precedem o nome especificado, como os 45 caracteres no `arn:aws:ssm:us-east-2:111122223333:parameter/`.
- Prefixos: um nome de parâmetro não pode ser prefixado com "`aws`" ou "`ssm`" (sem distinção entre maiúsculas e minúsculas). Por exemplo, as tentativas de criar parâmetros com os seguintes nomes falaharão sem exceção:
  - `awsTestParameter`
  - `SSM-testparameter`
  - `/aws/testparam1`

> **Note**
>
> Quando você especifica um parâmetro em um documento, comando ou script do SSM, deve incluir `ssm` como parte da sintaxe, conforme mostrado nos exemplos a seguir. Válido: {{ssm:*parameter-name*}} e {{ ssm:*parameter-name* }}, como `{{ssm:MyParameter}}` e `{{ ssm:MyParameter }}`.

- Exclusividade: um nome de parâmetro deve ser exclusivo em uma região da Região da AWS. Por exemplo, o Systems Manager trata os seguintes parâmetros como parâmetros separados, se eles existirem na mesma região:
  - `/Test/TestParam1`
  - `/TestParam1`

  Os exemplos a seguir também são exclusivos:
  - `/Test/TestParam1/Logpath1`
  - `/Test/TestParam1`

  No entanto, se estiverem na mesma região, os seguintes exemplos não serão considerados como exclusivos:
  - `/TestParam1`
  - `TestParam1`
- Profundidade da hierarquia: se você especificar uma hierarquia de parâmetros, ela poderá ter uma profundidade máxima de quinze níveis. É possível definir um parâmetro em qualquer nível da hierarquia. Ambos os exemplos a seguir são estruturalmente válidos:
  - `/Level-1/L2/L3/L4/L5/L6/L7/L8/L9/L10/L11/L12/L13/L14/parameter-name`
  - `parameter-name`

  A tentativa de criar o seguinte parâmetro falharia com uma exceção `HierarchyLevelLimitExceededException`:
  - `/Level-1/L2/L3/L4/L5/L6/L7/L8/L9/L10/L11/L12/L13/L14/L15/L16/parameter-name`

> **Important**
>
> Se um usuário tiver acesso a um caminho, o usuário poderá acessar todos os níveis desse caminho. Por exemplo, se um usuário tiver permissão para acessar um caminho `/a`, ele também pode acessar `/a/b`. Mesmo se o acesso de um usuário tiver sido explicitamente negado no AWS Identity and Access Management (IAM) para o parâmetro `/a/b`, ele ainda poderá chamar a operação de API GetParametersByPath recursivamente para `/a` e visualizar o `/a/b`.

Tópicos

- Criar um parâmetro do Parameter Store usando o console
- Criar um parâmetro do Parameter Store usando a AWS CLI
- Criar um parâmetro do Parameter Store usando as Ferramentas para Windows PowerShell

Criar um parâmetro do Parameter Store usando o console

Você pode usar o console do AWS Systems Manager para criar e executar tipos de parâmetros `String`, `StringList` e `SecureString`. Depois de excluir um parâmetro, aguarde pelo menos 30 segundos para criar um parâmetro com o mesmo nome.

> **Note**
>
> Parâmetros só estão disponíveis na região da Região da AWS em que foram criados.

O procedimento a seguir demonstra o processo de criação e armazenamento de um parâmetro usando o console do Parameter Store. É possível criar tipos de parâmetro `String`, `StringList` e `SecureString` a partir do console.

Para criar um parâmetro

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Parameter Store.
3. Escolha Criar Parâmetro.
4. Na caixa Name (Nome), insira uma hierarquia e um nome. Por exemplo, digite **`/Test/helloWorld`**.

   Para obter mais informações sobre hierarquias de parâmetros, consulte Trabalhar com hierarquias de parâmetros no Parameter Store.
5. Na caixa Description, digite uma descrição que identifique esse parâmetro como um parâmetro de teste.
6. Para Parameter tier (Nível do parâmetro), escolha Standard (Padrão) ou Advanced (Avançado). Para obter mais informações sobre parâmetros avançados, consulte Gerenciar camadas de parâmetros.
7. Para Tipo, escolha String, StringList ou SecureString.

- Se você escolher String, o campo Data type (Tipo de dados) será exibido. Se você estiver criando um parâmetro para manter o ID do recurso para uma Amazon Machine Image (AMI), selecione `aws:ec2:image`. Caso contrário, deixe o `text` padrão selecionado.
- Se escolher SecureString, o campo KMS Key ID será exibido. Se você não fornecer um ID AWS Key Management Service AWS KMS key, um AWS KMS key nome do recurso da Amazon (ARN), um nome de alias ou um ARN de alias, o sistema usará `alias/aws/ssm`, que é oChave gerenciada pela AWS para o Systems Manager. Se não quiser usar essa chave, você poderá usar uma CMK gerenciada pelo cliente. Para obter mais informações sobre Chaves gerenciadas pela AWS e chaves gerenciadas pelo cliente, consulte Conceitos do AWS Key Management Service, no Guia do desenvolvedor do AWS Key Management Service. Para obter mais informações sobre a criptografia do Parameter Store e do AWS KMS, consulte Como o AWS Systems Manager usa o Parameter StoreAWS KMS.

  > **Important**
  >
  > O Parameter Store só oferece suporte a chaves de criptografia simétricas KMS. Não é possível usar uma chave de criptografia KMS assimétrica para criptografar os parâmetros. Para obter ajuda para determinar se uma KMS é simétrica ou assimétrica, consulte Identificar KMSs simétricas e assimétricas no Manual do desenvolvedor do AWS Key Management Service.

- Ao criar um parâmetro `SecureString` no console usando o parâmetro `key-id` com um nome de alias de CMK gerenciada pelo cliente ou com um ARN de alias, especifique o prefixo `alias/` antes desse alias. Veja um exemplo de ARN a seguir.

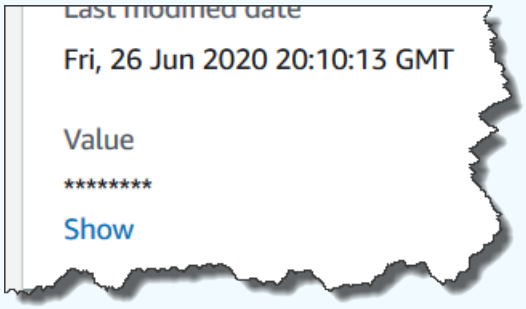
  ```
  arn:aws:kms:us-east-2:123456789012:alias/abcd1234-ab12-cd34-ef56-abcdeEXAMPLE
  ```

  Veja a seguir um exemplo de nome de alias.

  ```
  alias/MyAliasName
  ```

8. Na caixa Value, digite um valor. Por exemplo, digite **`This is my first parameter`** ou **`ami-0dbf5ea29aEXAMPLE`**.

> **Note**
>
> Os parâmetros não podem ser referenciados ou aninhados nos valores de outros parâmetros. Não é possível incluir `{{}}` ou `{{ssm:parameter-name}}` em um valor de parâmetro.
>
> Se você escolheu SecureString, o valor do parâmetro será mascarado por padrão (“******”) quando você visualizá-lo posteriormente na guia Visão geral do parâmetro, conforme mostrado na ilustração a seguir. Selecione Show (Mostrar) para exibir o valor do parâmetro.



9. (Opcional) Na área Tags, aplique um ou mais pares de chave-valor de tag ao parâmetro.

   Tags são metadados opcionais que você atribui a um recurso. As tags permitem categorizar um recurso de diferentes formas, como por finalidade, proprietário ou ambiente. Por exemplo, você pode marcar um parâmetro do Systems Manager para identificar o tipo de recurso ao qual ele se aplica, o ambiente ou o tipo de dados de configuração referenciado pelo parâmetro. Nesse caso, você pode especificar os seguintes pares de chave-valor:

   - `Key=Resource,Value=S3bucket`
   - `Key=OS,Value=Windows`
   - `Key=ParameterType,Value=LicenseKey`

10. Escolha Criar Parâmetro.
11. Na lista de parâmetros, escolha o nome do parâmetro que você acabou de criar. Verifique os detalhes na guia Overview. Se tiver criado um parâmetro `SecureString`, escolha Show para visualizar o valor não criptografado.

> **Note**
>
> Não é possível transformar um parâmetro avançado em um parâmetro padrão. Se você não precisar mais de um parâmetro avançado, ou se não quiser mais ser cobrado por um parâmetro avançado, deverá excluí-lo e recriá-lo como um novo parâmetro padrão.

Criar um parâmetro do Parameter Store usando a AWS CLI

Você pode usar a AWS Command Line Interface (AWS CLI) para criar tipos de parâmetro `String`, `StringList` e `SecureString`. Depois de excluir um parâmetro, aguarde pelo menos 30 segundos para criar um parâmetro com o mesmo nome.

Os parâmetros não podem ser referenciados ou aninhados nos valores de outros parâmetros. Não é possível incluir `{{}}` ou `{{ssm:parameter-name}}` em um valor de parâmetro.

> **Note**
>
> Parâmetros só estão disponíveis na região da Região da AWS em que foram criados.

Tópicos

- Criar um parâmetro do String usando a AWS CLI
- Criar um parâmetro StringList usando a AWS CLI
- Criar um parâmetro SecureString usando a AWS CLI
- Criar um parâmetro de várias linhas usando a AWS CLI

Criar um parâmetro do **`String`** usando a AWS CLI

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.
2. Execute o comando a seguir para criar um parâmetro do tipo `String`. Substitua cada *`espaço reservado para recurso de exemplo`* por suas próprias informações.

   Linux & macOS

   ```
   aws ssm put-parameter \
   ```

```
    --name "parameter-name" \
    --value "parameter-value" \
    --type String \
    --tags "Key=tag-key,Value=tag-value"
```

Windows

```
aws ssm put-parameter ^
    --name "parameter-name" ^
    --value "parameter-value" ^
    --type String ^
    --tags "Key=tag-key,Value=tag-value"
```

- ou -

Execute o comando a seguir para criar um parâmetro que contenha um ID da Amazon Machine Image (AMI) como o valor do parâmetro.

Linux & macOS

```
aws ssm put-parameter \
    --name "parameter-name" \
    --value "an-AMI-id" \
    --type String \
    --data-type "aws:ec2:image" \
    --tags "Key=tag-key,Value=tag-value"
```

Windows

```
aws ssm put-parameter ^
    --name "parameter-name" ^
    --value "an-AMI-id" ^
    --type String ^
    --data-type "aws:ec2:image" ^
    --tags "Key=tag-key,Value=tag-value"
```

A opção `--name` oferece suporte a hierarquias. Para obter informações sobre hierarquias, consulte Trabalhar com hierarquias de parâmetros no Parameter Store.

A opção `--data-type` deverá ser especificada somente se você estiver criando um parâmetro que contenha um ID de AMI. Ele valida se o valor do parâmetro inserido é um ID AMI Amazon Elastic Compute Cloud (Amazon EC2) formatado corretamente. Para todos os outros parâmetros, o tipo de dados padrão é `text` e é opcional especificar um valor. Para obter mais informações, consulte Como usar o suporte a parâmetros nativos no Parameter Store para IDs de imagem de máquina da Amazon.

> **Important**
>
> Se for bem-sucedido, o comando retornará o número da versão do parâmetro. Exceção: se você tiver especificado `aws:ec2:image` como o tipo de dados, um novo número de versão na resposta não significa que o valor do parâmetro já tenha sido validado. Para obter mais informações, consulte Como usar o suporte a parâmetros nativos no Parameter Store para IDs de imagem de máquina da Amazon.

O exemplo a seguir adiciona duas tags de par de chave-valor a um parâmetro.

Linux & macOS

```
aws ssm put-parameter \
    --name parameter-name \
    --value "parameter-value" \
    --type "String" \
    --tags '[{"Key":"Region","Value":"East"},{"Key":"Environment",
 "Value":"Production"}]'
```

Windows

```
aws ssm put-parameter ^
    --name parameter-name ^
    --value "parameter-value" ^
    --type "String" ^
    --tags [{\"Key\":\"Region1\",\"Value\":\"East1\"},{\"Key\":\"Environment1\",
\"Value\":\"Production1\"}]
```

O exemplo a seguir usa uma hierarquia de parâmetros no nome para criar um texto sem formataçãoStringparâmetro . O comando retornará o número da versão do parâmetro. Para

obter mais informações sobre hierarquias de parâmetros, consulte Trabalhar com hierarquias de parâmetros no Parameter Store.

Linux & macOS

Parâmetro não em uma hierarquia

```
aws ssm put-parameter \
    --name "golden-ami" \
    --type "String" \
    --value "ami-12345abcdeEXAMPLE"
```

Parâmetro em uma hierarquia

```
aws ssm put-parameter \
    --name "/amis/linux/golden-ami" \
    --type "String" \
    --value "ami-12345abcdeEXAMPLE"
```

Windows

Parâmetro não em uma hierarquia

```
aws ssm put-parameter ^
    --name "golden-ami" ^
    --type "String" ^
    --value "ami-12345abcdeEXAMPLE"
```

Parâmetro em uma hierarquia

```
aws ssm put-parameter ^
    --name "/amis/windows/golden-ami" ^
    --type "String" ^
    --value "ami-12345abcdeEXAMPLE"
```

3. Execute o comando a seguir para visualizar o valor do parâmetro mais recente e verificar os detalhes do novo parâmetro.

```
aws ssm get-parameters --names "/Test/IAD/helloWorld"
```

O sistema retorna informações como estas.

```
{
    "InvalidParameters": [],
    "Parameters": [
        {
            "Name": "/Test/IAD/helloWorld",
            "Type": "String",
            "Value": "My updated parameter value",
            "Version": 2,
            "LastModifiedDate": "2020-02-25T15:55:33.677000-08:00",
            "ARN": "arn:aws:ssm:us-east-2:123456789012:parameter/Test/IAD/
helloWorld"
        }
    ]
}
```

Execute o seguinte comando para alterar o valor do parâmetro. O comando retornará o número da versão do parâmetro.

```
aws ssm put-parameter --name "/Test/IAD/helloWorld" --value "My updated 1st parameter"
 --type String --overwrite
```

Execute o seguinte comando para visualizar o histórico de valores de parâmetros.

```
aws ssm get-parameter-history --name "/Test/IAD/helloWorld"
```

Execute o seguinte comando para usar esse parâmetro em um comando do

```
aws ssm send-command --document-name "AWS-RunShellScript" --parameters '{"commands":
["echo {{ssm:/Test/IAD/helloWorld}}"]}' --targets "Key=instanceids,Values=instance-ids"
```

Execute o seguinte comando se você quiser recuperar apenas o parâmetro Value.

```
aws ssm get-parameter --name testDataTypeParameter --query "Parameter.Value"
```

Execute o seguinte comando se você quiser somente recuperar o parâmetro Value usando`get-parameters`.

```
aws ssm get-parameters --names "testDataTypeParameter" --query "Parameters[*].Value"
```

Execute o seguinte comando para visualizar os metadados de parâmetros.

```
aws ssm describe-parameters --filters "Key=Name,Values=/Test/IAD/helloWorld"
```

> **Note**
>
> Name deve estar em maiúscula.

O sistema retorna informações como estas.

```
{
    "Parameters": [
        {
            "Name": "helloworld",
            "Type": "String",
            "LastModifiedUser": "arn:aws:iam::123456789012:user/JohnDoe",
            "LastModifiedDate": 1494529763.156,
            "Version": 1,
            "Tier": "Standard",
            "Policies": []
        }
    ]
}
```

Criar um parâmetro StringList usando a AWS CLI

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.
2. Execute o seguinte comando para criar um parâmetro. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   Linux & macOS

   ```
   aws ssm put-parameter \
       --name "parameter-name" \
       --value "a-comma-separated-list-of-values" \
   ```

```
    --type StringList \
    --tags "Key=tag-key,Value=tag-value"
```

Windows

```
aws ssm put-parameter ^
    --name "parameter-name" ^
    --value "a-comma-separated-list-of-values" ^
    --type StringList ^
    --tags "Key=tag-key,Value=tag-value"
```

> **Note**
>
> Se for bem-sucedido, o comando retornará o número da versão do parâmetro.

Este exemplo adiciona duas tags de par de chave-valor a um parâmetro. (Dependendo do tipo de sistema operacional na sua máquina local, execute um dos comandos a seguir. A versão a ser executada em uma máquina local do Windows inclui os caracteres de escape ("\") dos quais o comando deve serr executado na sua ferramenta da linha de comando).

Veja um `StringList` de exemplo que usa uma hierarquia de parâmetros.

Linux & macOS

```
aws ssm put-parameter \
    --name /IAD/ERP/Oracle/addUsers \
    --value "Milana,Mariana,Mark,Miguel" \
    --type StringList
```

Windows

```
aws ssm put-parameter ^
    --name /IAD/ERP/Oracle/addUsers ^
    --value "Milana,Mariana,Mark,Miguel" ^
    --type StringList
```

> **Note**
>
> Os itens em uma `StringList` devem ser separados por uma vírgula (,). Não é possível usar outros sinais de pontuação ou caracteres especiais para inserir um caractere de escape para os itens da lista. Se você tiver um valor de parâmetro que requer uma vírgula, use o tipo `String`.

3. Execute o comando `get-parameters` para verificar os detalhes do parâmetro. Por exemplo:

```
aws ssm get-parameters --name "/IAD/ERP/Oracle/addUsers"
```

Criar um parâmetro SecureString usando a AWS CLI

Use o procedimento a seguir para criar um parâmetro `SecureString`. Substitua cada *`espaço reservado para recurso de exemplo`* por suas próprias informações.

> **Important**
>
> Somente o valor de um parâmetro `SecureString` é criptografado. O nome do parâmetro, a descrição e outras propriedades não são criptografados.

> **Important**
>
> O Parameter Store só oferece suporte a chaves de criptografia simétricas KMS. Não é possível usar uma chave de criptografia KMS assimétrica para criptografar os parâmetros. Para obter ajuda para determinar se uma KMS é simétrica ou assimétrica, consulte Identificar KMSs simétricas e assimétricas no Manual do desenvolvedor do AWS Key Management Service.

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.
2. Execute um dos seguintes comandos para criar um parâmetro que use o tipo de dados `SecureString`.

Linux & macOS

Criar um parâmetro **SecureString** usando a Chave gerenciada pela AWS padrão

```
aws ssm put-parameter \
    --name "parameter-name" \
    --value "parameter-value" \
    --type "SecureString"
```

Crie um parâmetro **SecureString** que use uma chave gerenciada personalizada

```
aws ssm put-parameter \
    --name "parameter-name" \
    --value "a-parameter-value, for example P@ssW%rd#1" \
    --type "SecureString"
    --tags "Key=tag-key,Value=tag-value"
```

Criar um parâmetro **SecureString** que usa uma chave personalizada do AWS KMS

```
aws ssm put-parameter \
    --name "parameter-name" \
    --value "a-parameter-value, for example P@ssW%rd#1" \
    --type "SecureString" \
    --key-id "your-account-ID/the-custom-AWS KMS-key" \
    --tags "Key=tag-key,Value=tag-value"
```

Windows

Criar um parâmetro **SecureString** usando a Chave gerenciada pela AWS padrão

```
aws ssm put-parameter ^
    --name "parameter-name" ^
    --value "parameter-value" ^
    --type "SecureString"
```

Crie um parâmetro **SecureString** que use uma chave gerenciada personalizada

```
aws ssm put-parameter ^
    --name "parameter-name" ^
    --value "a-parameter-value, for example P@ssW%rd#1" ^
```

```
    --type "SecureString" ^
    --tags "Key=tag-key,Value=tag-value"
```

Criar um parâmetro **`SecureString`** que usa uma chave personalizada do AWS KMS

```
aws ssm put-parameter ^
    --name "parameter-name" ^
    --value "a-parameter-value, for example P@ssW%rd#1" ^
    --type "SecureString" ^
    --key-id " ^
    --tags "Key=tag-key,Value=tag-value"account-ID/the-custom-AWS KMS-key"
```

Se você criar um parâmetro `SecureString` usando a chave do Chave gerenciada pela AWS na sua conta e região, não será necessário fornecer um valor para o parâmetro `--key-id`.

> **Note**
>
> Para usar a AWS KMS key atribuída à sua Conta da AWS e Região da AWS, remova o parâmetro `key-id` do comando. Para obter mais informações sobre AWS KMS keys, consulte Conceitos do AWS Key Management Service no Guia do desenvolvedor do AWS Key Management Service.

Para usar uma CMK gerenciada pelo cliente, em vez da CMK gerenciada pela Chave gerenciada pela AWS atribuída à sua conta, você deve especificar a chave usando o parâmetro `--key-id`. O parâmetro oferece suporte aos formatos de parâmetros do KMS a seguir.

- Exemplo de nome do recurso da Amazon (ARN):

  `arn:aws:kms:us-east-2:123456789012:key/key-id`

- Exemplo de ARN alias:

  `arn:aws:kms:us-east-2:123456789012:alias/alias-name`

- Exemplo de ID de chave:

  `12345678-1234-1234-1234-123456789012`

- Exemplo de nome de alias:

```
alias/MyAliasName
```

Você pode criar uma chave gerenciada pelo cliente usando o AWS Management Console ou a API do AWS KMS. Os seguintes comandos da AWS CLI criam uma chave gerenciada pelo cliente na Região da AWS atual da sua Conta da AWS.

```
aws kms create-key
```

Use um comando no seguinte formato para criar um parâmetro `SecureString` usando a chave que você acabou de criar.

O exemplo a seguir usa um nome ofuscado (`3l3vat3131`) para um parâmetro de senha e uma AWS KMS key.

Linux & macOS

```
aws ssm put-parameter \
    --name /Finance/Payroll/3l3vat3131 \
    --value "P@sSwW)rd" \
    --type SecureString \
    --key-id arn:aws:kms:us-
east-2:123456789012:key/1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e
```

Windows

```
aws ssm put-parameter ^
    --name /Finance/Payroll/3l3vat3131 ^
    --value "P@sSwW)rd" ^
    --type SecureString ^
    --key-id arn:aws:kms:us-
east-2:123456789012:key/1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e
```

3. Execute o seguinte comando para verificar os detalhes do parâmetro.

   Se você não especificar o parâmetro `with-decryption` ou se especificar o parâmetro `no-with-decryption`, o comando retornará um GUID criptografado.

Linux & macOS

```
aws ssm get-parameters \
    --name "the-parameter-name-you-specified" \
    --with-decryption
```

Windows

```
aws ssm get-parameters ^
    --name "the-parameter-name-you-specified" ^
    --with-decryption
```

4. Execute o seguinte comando para visualizar os metadados de parâmetros.

   Linux & macOS

   ```
   aws ssm describe-parameters \
       --filters "Key=Name,Values=the-name-that-you-specified"
   ```

   Windows

   ```
   aws ssm describe-parameters ^
       --filters "Key=Name,Values=the-name-that-you-specified"
   ```

5. Execute o comando a seguir para alterar o valor do parâmetro se você não estiver usando uma AWS KMS key gerenciada pelo cliente.

   Linux & macOS

   ```
   aws ssm put-parameter \
       --name "the-name-that-you-specified" \
       --value "a-new-parameter-value" \
       --type "SecureString" \
       --overwrite
   ```

   Windows

   ```
   aws ssm put-parameter ^
       --name "the-name-that-you-specified" ^
       --value "a-new-parameter-value" ^
   ```

```
    --type "SecureString" ^
    --overwrite
```

- ou -

Execute um dos comandos a seguir para alterar o valor do parâmetro se você estiver usando uma AWS KMS key gerenciada pelo cliente.

Linux & macOS

```
aws ssm put-parameter \
    --name "the-name-that-you-specified" \
    --value "a-new-parameter-value" \
    --type "SecureString" \
    --key-id "the-KMSkey-ID" \
    --overwrite
```

```
aws ssm put-parameter \
    --name "the-name-that-you-specified" \
    --value "a-new-parameter-value" \
    --type "SecureString" \
    --key-id "account-alias/the-KMSkey-ID" \
    --overwrite
```

Windows

```
aws ssm put-parameter ^
    --name "the-name-that-you-specified" ^
    --value "a-new-parameter-value" ^
    --type "SecureString" ^
    --key-id "the-KMSkey-ID" ^
    --overwrite
```

```
aws ssm put-parameter ^
    --name "the-name-that-you-specified" ^
    --value "a-new-parameter-value" ^
    --type "SecureString" ^
    --key-id "account-alias/the-KMSkey-ID" ^
    --overwrite
```

6. Execute o seguinte comando para visualizar o valor do parâmetro mais recente.

   Linux & macOS

   ```
   aws ssm get-parameters \
       --name "the-name-that-you-specified" \
       --with-decryption
   ```

   Windows

   ```
   aws ssm get-parameters ^
       --name "the-name-that-you-specified" ^
       --with-decryption
   ```

7. Execute o seguinte comando para visualizar o histórico de valores de parâmetros.

   Linux & macOS

   ```
   aws ssm get-parameter-history \
       --name "the-name-that-you-specified"
   ```

   Windows

   ```
   aws ssm get-parameter-history ^
       --name "the-name-that-you-specified"
   ```

> **Note**
>
> Você pode criar manualmente um parâmetro com um valor criptografado. Nesse caso, como o valor já está criptografado, não é necessário escolher o tipo de parâmetro `SecureString`. Se você escolher `SecureString`, seu parâmetro será duplamente criptografado.

Por padrão, todos os valores de `SecureString` são exibidos como texto cifrado. Para descriptografar um valor de `SecureString`, um usuário deve ter permissão para chamar a operação da API Decrypt do AWS KMS. Para obter informações sobre como configurar o controle de acesso do AWS KMS, consulte Autenticação e controle de acesso para o AWS KMS no Manual de desenvolvedor do AWS Key Management Service.

> **Important**
>
> Se você alterar o alias da chave KMS para a chave KMS usada para criptografar um parâmetro, também será necessário atualizar o alias de chave que o parâmetro usa para referenciar a AWS KMS. Isso se aplica somente ao alias da chave KMS; o ID da chave que um alias anexa permanece o mesmo, a menos que você exclua a chave inteira.

Criar um parâmetro de várias linhas usando a AWS CLI

Você pode usar a AWS CLI para criar um parâmetro com quebras de linha. Use quebras de linha para dividir o texto em valores de parâmetro mais longos para melhor legibilidade ou, por exemplo, atualizar o conteúdo de parâmetro de vários parágrafos para uma página da Web. Você pode incluir o conteúdo em um arquivo JSON e usar o`--cli-input-json`, usando caracteres de quebra de linha como`\n`, conforme mostrado no exemplo a seguir.

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Execute o seguinte comando para criar um parâmetro do tipo String.

   Linux & macOS

   ```
   aws ssm put-parameter \
       --name "MultiLineParameter" \
       --type String \
       --cli-input-json file://MultiLineParameter.json
   ```

   Windows

   ```
   aws ssm put-parameter ^
       --name "MultiLineParameter" ^
       --type String ^
       --cli-input-json file://MultiLineParameter.json
   ```

   O exemplo a seguir mostra o conteúdo de um arquivo `MultiLineParameter.json`.

   ```
   {
   ```

```
    "Value": "<para>Paragraph One</para>\n<para>Paragraph Two</para>
\n<para>Paragraph Three</para>"
}
```

O valor do parâmetro salvo é armazenado da seguinte forma.

```
<para>Paragraph One</para>
<para>Paragraph Two</para>
<para>Paragraph Three</para>
```

Criar um parâmetro do Parameter Store usando as Ferramentas para Windows PowerShell

Você pode usar AWS Tools for Windows PowerShell para criar os tipos de parâmetro `String`, `StringList` e `SecureString`. Depois de excluir um parâmetro, aguarde pelo menos 30 segundos para criar um parâmetro com o mesmo nome.

Os parâmetros não podem ser referenciados ou aninhados nos valores de outros parâmetros. Não é possível incluir `{{}}` ou `{{ssm:parameter-name}}` em um valor de parâmetro.

> **Note**
>
> Parâmetros só estão disponíveis na região da Região da AWS em que foram criados.

Tópicos

- Criar um parâmetro string (Ferramentas para Windows PowerShell)
- Criar um parâmetro StringList (Ferramentas para Windows PowerShell)
- Criar um parâmetro SecureString (Ferramentas para Windows PowerShell)

Criar um parâmetro string (Ferramentas para Windows PowerShell)

1. Instale e configure o Ferramentas da AWS para PowerShell (Ferramentas para Windows PowerShell), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar o Ferramentas da AWS para PowerShell.
2. Execute o seguinte comando para criar um parâmetro que contenha um valor de texto sem formatação. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

```
Write-SSMParameter `
    -Name "parameter-name" `
    -Value "parameter-value" `
    -Type "String"
```

- ou -

Execute o comando a seguir para criar um parâmetro que contenha um ID da Amazon Machine Image (AMI) como o valor do parâmetro.

> **Note**
>
> Para criar um parâmetro com uma tag, crie o service.model.tag antes da mão como uma variável. Aqui está um exemplo.
>
> ```
> $tag = New-Object Amazon.SimpleSystemsManagement.Model.Tag
> $tag.Key = "tag-key"
> $tag.Value = "tag-value"
> ```

```
Write-SSMParameter `
    -Name "parameter-name" `
    -Value "an-AMI-id" `
    -Type "String" `
    -DataType "aws:ec2:image" `
    -Tags $tag
```

A opção `-DataType` deverá ser especificada somente se você estiver criando um parâmetro que contenha um ID de AMI. Para todos os outros parâmetros, o tipo de dados padrão é `text`. Para obter mais informações, consulte Como usar o suporte a parâmetros nativos no Parameter Store para IDs de imagem de máquina da Amazon.

Veja um de exemplo que usa uma hierarquia de parâmetros.

```
Write-SSMParameter `
    -Name "/IAD/Web/SQL/IPaddress" `
    -Value "99.99.99.999" `
    -Type "String" `
```

```
    -Tags $tag
```

3. Execute o seguinte comando para verificar os detalhes do parâmetro.

```
(Get-SSMParameterValue -Name "the-parameter-name-you-specified").Parameters
```

Criar um parâmetro StringList (Ferramentas para Windows PowerShell)

1. Instale e configure o Ferramentas da AWS para PowerShell (Ferramentas para Windows PowerShell), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar o Ferramentas da AWS para PowerShell.

2. Execute o seguinte comando para criar um parâmetro StringList. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

> **Note**
>
> Para criar um parâmetro com uma tag, crie o service.model.tag antes da mão como uma variável. Aqui está um exemplo.
>
> ```
> $tag = New-Object Amazon.SimpleSystemsManagement.Model.Tag
> $tag.Key = "tag-key"
> $tag.Value = "tag-value"
> ```

```
Write-SSMParameter `
    -Name "parameter-name" `
    -Value "a-comma-separated-list-of-values" `
    -Type "StringList" `
    -Tags $tag
```

Se for bem-sucedido, o comando retornará o número da versão do parâmetro.

Aqui está um exemplo.

```
Write-SSMParameter `
    -Name "stringlist-parameter" `
    -Value "Milana,Mariana,Mark,Miguel" `
```

```
    -Type "StringList" `
    -Tags $tag
```

> **Note**
>
> Os itens em uma `StringList` devem ser separados por uma vírgula (,). Não é possível usar outros sinais de pontuação ou caracteres especiais para inserir um caractere de escape para os itens da lista. Se você tiver um valor de parâmetro que requer uma vírgula, use o tipo `String`.

3. Execute o seguinte comando para verificar os detalhes do parâmetro.

```
(Get-SSMParameterValue -Name "the-parameter-name-you-specified").Parameters
```

Criar um parâmetro SecureString (Ferramentas para Windows PowerShell)

Para criar um parâmetro `SecureString`, primeiro leia a respeito dos requisitos para esse tipo de parâmetro. Para obter mais informações, consulte Criar um parâmetro SecureString usando a AWS CLI.

> **Important**
>
> Somente o valor de um parâmetro `SecureString` é criptografado. O nome do parâmetro, a descrição e outras propriedades não são criptografados.

> **Important**
>
> O Parameter Store só oferece suporte a chaves de criptografia simétricas KMS. Não é possível usar uma chave de criptografia KMS assimétrica para criptografar os parâmetros. Para obter ajuda para determinar se uma KMS é simétrica ou assimétrica, consulte Identificar KMSs simétricas e assimétricas no Manual do desenvolvedor do AWS Key Management Service.

1. Instale e configure o Ferramentas da AWS para PowerShell (Ferramentas para Windows PowerShell), caso ainda não o tenha feito.

Para obter informações, consulte Instalar o Ferramentas da AWS para PowerShell.

2. Execute o seguinte comando para criar um parâmetro. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

> **Note**
>
> Para criar um parâmetro com uma tag, primeiro crie o service.model.tag como uma variável. Aqui está um exemplo.
>
> ```
> $tag = New-Object Amazon.SimpleSystemsManagement.Model.Tag
> $tag.Key = "tag-key"
> $tag.Value = "tag-value"
> ```

```
Write-SSMParameter `
    -Name "parameter-name" `
    -Value "parameter-value" `
    -Type "SecureString"  `
    -KeyId "an AWS KMS key ID, an AWS KMS key ARN, an alias name, or an alias ARN"
 `
    -Tags $tag
```

Se for bem-sucedido, o comando retornará o número da versão do parâmetro.

> **Note**
>
> Para usar a Chave gerenciada pela AWS atribuída à sua conta, remova o parâmetro `-KeyId` do comando.

Veja a seguir um exemplo que usa um nome ofuscado (3l3vat3131) para um parâmetro de senha e uma Chave gerenciada pela AWS.

```
Write-SSMParameter `
    -Name "/Finance/Payroll/3l3vat3131" `
    -Value "P@sSwW)rd" `
    -Type "SecureString"`
```

```
    -Tags $tag
```

3. Execute o seguinte comando para verificar os detalhes do parâmetro.

```
(Get-SSMParameterValue -Name "the-parameter-name-you-specified" -WithDecryption
 $true).Parameters
```

Por padrão, todos os valores de `SecureString` são exibidos como texto cifrado. Para descriptografar um valor de `SecureString`, um usuário deve ter permissão para chamar a operação da API Decrypt do AWS KMS. Para obter informações sobre como configurar o controle de acesso do AWS KMS, consulte Autenticação e controle de acesso para o AWS KMS no Manual de desenvolvedor do AWS Key Management Service.

> **Important**
>
> Se você alterar o alias da chave KMS para a chave KMS usada para criptografar um parâmetro, também será necessário atualizar o alias de chave que o parâmetro usa para referenciar a AWS KMS. Isso se aplica somente ao alias da chave KMS; o ID da chave que um alias anexa permanece o mesmo, a menos que você exclua a chave inteira.

Pesquisar parâmetros do Parameter Store no Systems Manager

Quando você tem um grande número de parâmetros em sua conta, pode ser difícil encontrar informações sobre apenas um ou alguns parâmetros de cada vez. Nesse caso, use ferramentas de filtro para procurar aqueles sobre os quais você precisa de informações, de acordo com os critérios de pesquisa especificados. Você pode usar o console do AWS Systems Manager, a AWS Command Line Interface (AWS CLI), o Ferramentas da AWS para PowerShell ou a API DescribeParameters para pesquisar parâmetros.

Tópicos

- Procurar um parâmetro usando o console
- Procurar um parâmetro usando a AWS CLI

Procurar um parâmetro usando o console

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.

2. No painel de navegação, escolha Parameter Store.
3. Clique na caixa de pesquisa e escolha como você deseja pesquisar. Por exemplo, `Type` ou `Name`.
4. Forneça informações para o tipo de pesquisa selecionado. Por exemplo:

   - Se estiver pesquisando por `Type`, escolha entre `String`, `StringList` ou `SecureString`.
   - Se você estiver pesquisando por `Name`, escolha `contains`, `equals` ou `begins-with` e insira todo ou parte do nome de um parâmetro.

   > **Note**
   >
   > No console, o tipo de pesquisa padrão para `Name` é `contains`.

5. Pressione Enter.

A lista de parâmetros é atualizada com os resultados da sua pesquisa.

> **Note**
>
> Sua pesquisa pode conter mais resultados do que os exibidos na primeira página de resultados. Use a seta para a direita (>) no tópico da lista de parâmetros (se disponível) para visualizar o próximo conjunto de resultados.

Procurar um parâmetro usando a AWS CLI

Use o comando `describe-parameters` para exibir informações sobre um ou mais parâmetros na AWS CLI.

Os exemplos a seguir demonstram várias opções que você pode usar para visualizar informações sobre os parâmetros em sua Conta da AWS. Para obter mais informações sobre essas opções, consulte describe-parameters no Guia do usuário da AWS Command Line Interface.

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.
2. Substitua os valores dos exemplos nos comandos a seguir por valores que refletem parâmetros que foram criados em sua conta.

Linux & macOS

```
aws ssm describe-parameters \
    --parameter-filters "Key=Name,Values=MyParameterName"
```

Windows

```
aws ssm describe-parameters ^
    --parameter-filters "Key=Name,Values=MyParameterName"
```

> **Note**
>
> Em `describe-parameters`, o tipo de pesquisa padrão para `Name` é `Equals`. Nos filtros de parâmetros, especificar `"Key=Name,Values=MyParameterName"` é o mesmo que especificar `"Key=Name,Option=Equals,Values=MyParameterName"`.

```
aws ssm describe-parameters \
    --parameter-filters "Key=Name,Option=Contains,Values=Product"
```

```
aws ssm describe-parameters \
    --parameter-filters "Key=Type,Values=String"
```

```
aws ssm describe-parameters \
    --parameter-filters "Key=Path,Values=/Production/West"
```

```
aws ssm describe-parameters \
    --parameter-filters "Key=Tier,Values=Standard"
```

```
aws ssm describe-parameters \
    --parameter-filters "Key=tag:tag-key,Values=tag-value"
```

```
aws ssm describe-parameters \
    --parameter-filters "Key=KeyId,Values=key-id"
```

> **Note**
>
> No último exemplo, *key-id* representa o ID de uma chave do AWS Key Management Service (AWS KMS) usada para criptografar um parâmetro `SecureString` criado na sua conta. Como alternativa, você pode inserir **alias/aws/ssm** para usar a chave padrão do AWS KMS para sua conta. Para obter mais informações, consulte Criar um parâmetro SecureString usando a AWS CLI.

Se houver êxito, o comando gerará uma saída semelhante à seguinte.

```
{
    "Parameters": [
        {
            "Name": "/Production/West/Manager",
            "Type": "String",
            "LastModifiedDate": 1573438580.703,
            "LastModifiedUser": "arn:aws:iam::111122223333:user/Mateo.Jackson",
            "Version": 1,
            "Tier": "Standard",
            "Policies": []
        },
        {
            "Name": "/Production/West/TeamLead",
            "Type": "String",
            "LastModifiedDate": 1572363610.175,
            "LastModifiedUser": "arn:aws:iam::111122223333:user/Mateo.Jackson",
            "Version": 1,
            "Tier": "Standard",
            "Policies": []
        },
        {
            "Name": "/Production/West/HR",
            "Type": "String",
            "LastModifiedDate": 1572363680.503,
            "LastModifiedUser": "arn:aws:iam::111122223333:user/Mateo.Jackson",
            "Version": 1,
            "Tier": "Standard",
            "Policies": []
        }
    ]
```

```
}
```

Atribuir políticas de parâmetros no Parameter Store

Políticas de parâmetros ajudam a gerenciar um conjunto crescente de parâmetros, permitindo atribuir critérios específicos a um parâmetro, como uma data de expiração ou tempo de vida. Elas são especialmente úteis para forçar você a atualizar ou excluir senhas e dados de configuração armazenados no Parameter Store, uma ferramenta do AWS Systems Manager. O Parameter Store oferece os seguintes tipos de políticas: `Expiration`, `ExpirationNotification` e `NoChangeNotification`.

> **Note**
>
> Para implementar ciclos de vida de rotação de senha, useAWS Secrets Manager. Você pode alternar, gerenciar e recuperar credenciais de banco de dados, chaves de API e outros segredos durante seu ciclo de vida, usando o Secrets Manager. Para obter mais informações, consulte O que é o AWS Secrets Manager? no Guia do usuário do AWS Secrets Manager.

O Parameter Store impõe políticas de parâmetros usando verificações periódicas assíncronas. Depois de criar uma política, você não precisa realizar ações adicionais para impor essa política. O Parameter Store realiza independentemente a ação definida pela política, de acordo com os critérios que você especificou.

> **Note**
>
> Políticas de parâmetros estão disponíveis apenas para parâmetros que usam o nível avançado de parâmetros. Para obter mais informações, consulte Gerenciar camadas de parâmetros.

Uma política de parâmetro é uma matriz JSON, como mostra a tabela a seguir. Você pode atribuir uma política ao criar um novo parâmetro avançado ou pode aplicar uma política atualizando um parâmetro. O Parameter Store é compatível com os seguintes tipos de políticas de parâmetros.

Política	Detalhes	Exemplos

Expiração	Essa política exclui o parâmetro. Você pode especificar uma data e hora específicas usando o formato `ISO_INSTANT` ou o formato `ISO_OFFSET_DATE_TIME`. Para alterar quando deseja que o parâmetro seja excluído, você deve atualizar a política. Atualizar um parâmetro não afeta a data ou hora de expiração da política anexada a ele. Quando a data e hora de expiração forem atingidas, o Parameter Store excluirá o parâmetro.
	
	Note
	Esse exemplo usa o formato `ISO_INSTANT`. Você também pode especificar uma data e hora usando o formato `ISO_OFFSET_DATE_TIME`. Aqui está um exemplo: `2019-11-01T22:13:48.87+10:30:00`.
		`{`
		`"Type": "Expiration",`
		`"Version": "1.0",`
		`"Attributes": {`
		`"Timestamp": "2018-12-02T21:34:33.000Z"`
		`}`
		`}`

ExpirationNotification	Essa política inicia um evento no Amazon EventBridge	`{`

Política	Detalhes	Exemplos
	(EventBridge) que notifica você sobre a expiração. Ao usar essa política, você pode receber notificações antes que o tempo de expiração seja atingido, em unidades de dias ou horas.	<pre>"Type": "Expirati onNotification", "Version": "1.0", "Attributes": { "Before": "15", "Unit": "Days" } }</pre>
NoChangeNotification	Essa política inicia um evento no EventBridge quando um parâmetro não foi modificad o por um período especific ado. Esse tipo de política é útil, por exemplo, quando uma senha precisa ser alterada em um determinado período de tempo. Essa política determina quando enviar uma notificação, lendo o atributo `LastModifiedTime` do parâmetro. Se você alterar ou editar um parâmetro, o sistema redefinirá o período de notificação com base no novo valor de `LastModif iedTime` .	<pre>{ "Type": "NoChange Notification", "Version": "1.0", "Attributes": { "After": "20", "Unit": "Days" } }</pre>

É possível atribuir várias políticas a um parâmetro. Por exemplo, você pode atribuir as políticas `Expiration` e `ExpirationNotification` para que o sistema acione um evento do EventBridge para notificá-lo sobre a exclusão iminente de um parâmetro. É possível atribuir um máximo de dez (10) políticas a um parâmetro.

O exemplo a seguir mostra a sintaxe da solicitação para uma solicitação da API PutParameter que atribui quatro políticas a um novo parâmetro `SecureString` chamado `ProdDB3`.

```
{
    "Name": "ProdDB3",
    "Description": "Parameter with policies",
    "Value": "P@ssW*rd21",
    "Type": "SecureString",
    "Overwrite": "True",
    "Policies": [
        {
            "Type": "Expiration",
            "Version": "1.0",
            "Attributes": {
                "Timestamp": "2018-12-02T21:34:33.000Z"
            }
        },
        {
            "Type": "ExpirationNotification",
            "Version": "1.0",
            "Attributes": {
                "Before": "30",
                "Unit": "Days"
            }
        },
        {
            "Type": "ExpirationNotification",
            "Version": "1.0",
            "Attributes": {
                "Before": "15",
                "Unit": "Days"
            }
        },
        {
            "Type": "NoChangeNotification",
            "Version": "1.0",
            "Attributes": {
                "After": "20",
                "Unit": "Days"
            }
        }
    ]
}
```

Adicionar políticas a um parâmetro existente

Esta seção inclui informações sobre como adicionar políticas a um parâmetro usando o console do AWS Systems Manager, a AWS Command Line Interface (AWS CLI) e o AWS Tools for Windows PowerShell. Para obter informações sobre como criar um novo parâmetro que inclua políticas, consulte Criar parâmetros do Parameter Store no Systems Manager.

Tópicos

- Adicionar políticas a um parâmetro existente usando o console
- Adicionar políticas a um parâmetro existente usando a AWS CLI
- Adicionar políticas a um parâmetro existente (Ferramentas para Windows PowerShell)

Adicionar políticas a um parâmetro existente usando o console

Use o procedimento a seguir para adicionar políticas a um parâmetro existente usando o console do Systems Manager.

Para adicionar políticas a um parâmetro existente

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Parameter Store.
3. Escolha a opção ao lado do parâmetro que você deseja atualizar para incluir políticas e depois escolha Edit (Editar).
4. Escolha Advanced (Avançado).
5. (Opcional) Na seção Parameter policies (Políticas de parâmetro), escolha Enabled (Habilitado). Você pode especificar uma data de expiração e uma ou mais políticas de notificação para esse parâmetro.
6. Escolha Salvar alterações.

> **Important**
>
> - O Parameter Store manterá as políticas em um parâmetro até que você queira removê-las ou substituí-las por novas políticas.
> - Para remover todas as políticas de um parâmetro existente, edite esse parâmetro e aplique uma política vazia usando colchetes e chaves, da seguinte maneira: `[{}]`

- Se você adicionar uma nova política a um parâmetro que já tiver políticas, o Systems Manager substituirá as políticas anexadas a esse parâmetro. As políticas existentes são excluídas. Se quiser adicionar uma nova política a um parâmetro que já tenha uma ou mais políticas, copie e cole as políticas originais, digite a nova política e depois salve as alterações.

Adicionar políticas a um parâmetro existente usando a AWS CLI

Use o procedimento a seguir para adicionar políticas a um parâmetro usando a AWS CLI.

Para adicionar políticas a um parâmetro existente

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Execute o comando a seguir para adicionar políticas a um parâmetro existente. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   Linux & macOS

   ```
   aws ssm put-parameter
       --name "parameter name" \
       --value 'parameter value' \
       --type parameter type \
       --overwrite \
       --policies "[{policies-enclosed-in-brackets-and-curly-braces}]"
   ```

   Windows

   ```
   aws ssm put-parameter
       --name "parameter name" ^
       --value 'parameter value' ^
       --type parameter type ^
       --overwrite ^
       --policies "[{policies-enclosed-in-brackets-and-curly-braces}]"
   ```

Veja a seguir um exemplo que inclui uma política de expiração que exclui o parâmetro após 15 dias. O exemplo também inclui uma política de notificação que gera um evento do EventBridge 5 (cinco) dias antes de o parâmetro ser excluído. Por último, ele incluirá uma política `NoChangeNotification` se não forem feitas alterações nesse parâmetro após 60 dias. O exemplo usa um nome ofuscado (`3l3vat3131`) para uma senha e uma AWS KMS key do AWS Key Management Service. Para obter mais informações sobre AWS KMS keys, consulte Conceitos do AWS Key Management Service no Guia do desenvolvedor do AWS Key Management Service.

Linux & macOS

```
aws ssm put-parameter \
    --name "/Finance/Payroll/3l3vat3131" \
    --value "P@sSwW)rd" \
    --type "SecureString" \
    --overwrite \
    --policies "[{\"Type\":\"Expiration\",\"Version\":\"1.0\",\"Attributes\":
{\"Timestamp\":\"2020-05-13T00:00:00.000Z\"}},{\"Type\":\"ExpirationNotification
\",\"Version\":\"1.0\",\"Attributes\":{\"Before\":\"5\",\"Unit\":\"Days\"}},
{\"Type\":\"NoChangeNotification\",\"Version\":\"1.0\",\"Attributes\":{\"After
\":\"60\",\"Unit\":\"Days\"}}]"
```

Windows

```
aws ssm put-parameter ^
    --name "/Finance/Payroll/3l3vat3131" ^
    --value "P@sSwW)rd" ^
    --type "SecureString" ^
    --overwrite ^
    --policies "[{\"Type\":\"Expiration\",\"Version\":\"1.0\",\"Attributes\":
{\"Timestamp\":\"2020-05-13T00:00:00.000Z\"}},{\"Type\":\"ExpirationNotification
\",\"Version\":\"1.0\",\"Attributes\":{\"Before\":\"5\",\"Unit\":\"Days\"}},
{\"Type\":\"NoChangeNotification\",\"Version\":\"1.0\",\"Attributes\":{\"After
\":\"60\",\"Unit\":\"Days\"}}]"
```

3. Execute o seguinte comando para verificar os detalhes do parâmetro. Substitua *`nome do parâmetro`* pelas suas próprias informações.

Linux & macOS

```
aws ssm describe-parameters  \
    --parameter-filters "Key=Name,Values=parameter name"
```

Windows

```
aws ssm describe-parameters  ^
    --parameter-filters "Key=Name,Values=parameter name"
```

Important

- O Parameter Store manterá as políticas de um parâmetro até que você queira removê-las ou substituí-las por novas políticas.
- Para remover todas as políticas de um parâmetro existente, edite esse parâmetro e aplique uma política vazia de colchetes e chaves. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações. Por exemplo:

  Linux & macOS

  ```
  aws ssm put-parameter \
      --name parameter name \
      --type parameter type  \
      --value 'parameter value' \
      --policies "[{}]"
  ```

  Windows

  ```
  aws ssm put-parameter ^
      --name parameter name ^
      --type parameter type  ^
      --value 'parameter value' ^
      --policies "[{}]"
  ```

- Se você adicionar uma nova política a um parâmetro que já tiver políticas, o Systems Manager substituirá as políticas anexadas a esse parâmetro. As políticas existentes são excluídas. Se quiser adicionar uma nova política a um parâmetro que já tenha uma ou

mais políticas, copie e cole as políticas originais, digite a nova política e depois salve as alterações.

Adicionar políticas a um parâmetro existente (Ferramentas para Windows PowerShell)

Use o procedimento a seguir para adicionar políticas a um parâmetro usando o console do Tools for Windows PowerShell. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

Para adicionar políticas a um parâmetro existente

1. Abra o Tools for Windows PowerShell e execute o seguinte comando para especificar suas credenciais. Você deve ter permissões de administrador no Amazon Elastic Compute Cloud (Amazon EC2) ou deve ter recebido a permissão apropriada no AWS Identity and Access Management (IAM).

```
Set-AWSCredentials `
    -AccessKey access-key-name `
    -SecretKey secret-key-name
```

2. Execute o seguinte comando para configurar a região da sua sessão do PowerShell. O exemplo usa a região Leste dos EUA (Ohio) (us-east-2).

```
Set-DefaultAWSRegion `
    -Region us-east-2
```

3. Execute o comando a seguir para adicionar políticas a um parâmetro existente. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

```
Write-SSMParameter `
    -Name "parameter name" `
    -Value "parameter value" `
    -Type "parameter type" `
    -Policies "[{policies-enclosed-in-brackets-and-curly-braces}]" `
    -Overwrite
```

   Veja a seguir um exemplo que inclui uma política de expiração que exclui o parâmetro à meia-noite (GMT) do dia 13 de maio de 2020. O exemplo também inclui uma política de notificação que gera um evento do EventBridge 5 (cinco) dias antes de o parâmetro ser excluído. Por

último, ele incluirá uma política `NoChangeNotification` se não forem feitas alterações nesse parâmetro após 60 dias. O exemplo usa um nome ofuscado (`3l3vat3131`) para uma senha e uma Chave gerenciada pela AWS.

```
Write-SSMParameter `
    -Name "/Finance/Payroll/3l3vat3131" `
    -Value "P@sSwW)rd" `
    -Type "SecureString" `
    -Policies "[{\"Type\":\"Expiration\",\"Version\":\"1.0\",\"Attributes\":
{\"Timestamp\":\"2018-05-13T00:00:00.000Z\"}},{\"Type\":\"ExpirationNotification
\",\"Version\":\"1.0\",\"Attributes\":{\"Before\":\"5\",\"Unit\":\"Days\"}},{\"Type
\":\"NoChangeNotification\",\"Version\":\"1.0\",\"Attributes\":{\"After\":\"60\",
\"Unit\":\"Days\"}}]" `
    -Overwrite
```

4. Execute o seguinte comando para verificar os detalhes do parâmetro. Substitua *nome do parâmetro* pelas suas próprias informações.

```
(Get-SSMParameterValue -Name "parameter name").Parameters
```

Important

- O Parameter Store manterá as políticas em um parâmetro até que você queira removê-las ou substituí-las por novas políticas.
- Para remover todas as políticas de um parâmetro existente, edite esse parâmetro e aplique uma política vazia de colchetes e chaves. Por exemplo:

  ```
  Write-SSMParameter `
      -Name "parameter name" `
      -Value "parameter value" `
      -Type "parameter type" `
      -Policies "[{}]"
  ```

- Se você adicionar uma nova política a um parâmetro que já tiver políticas, o Systems Manager substituirá as políticas anexadas a esse parâmetro. As políticas existentes são excluídas. Se quiser adicionar uma nova política a um parâmetro que já tenha uma ou mais políticas, copie e cole as políticas originais, digite a nova política e depois salve as alterações.

Trabalhar com hierarquias de parâmetros no Parameter Store

Gerenciar dezenas ou centenas de parâmetros como uma lista simples é um processo demorado e propenso a erros. Também pode ser difícil identificar o parâmetro correto para uma tarefa. Isso significa que você pode usar acidentalmente o parâmetro errado ou pode criar vários parâmetros que usam os mesmos dados de configuração.

Você pode usar hierarquias de parâmetros para ajudar você a organizar e gerenciar parâmetros. Uma hierarquia é um nome de parâmetro que inclui um caminho definido usando barras (/).

Tópicos

- Entender a hierarquia de parâmetros por meio de exemplos
- Consultar parâmetros em uma hierarquia
- Gerenciar parâmetros com hierarquias usando a AWS CLI

Entender a hierarquia de parâmetros por meio de exemplos

O exemplo a seguir usa três níveis de hierarquia no nome para identificar o seguinte:

```
/Environment/Type of computer/Application/Data
```

```
/Dev/DBServer/MySQL/db-string13
```

É possível criar uma hierarquia com um máximo de 15 níveis. Sugerimos que você crie hierarquias que reflitam uma estrutura hierárquica existente no seu ambiente, conforme indicado nos exemplos a seguir:

- Seu ambiente de integração contínua e entrega contínua (fluxos de trabalho de CI/CD)

  ```
  /Dev/DBServer/MySQL/db-string
  ```

  ```
  /Staging/DBServer/MySQL/db-string
  ```

  ```
  /Prod/DBServer/MySQL/db-string
  ```

- Seus aplicativos que utilizam contêineres

  ```
  /MyApp/.NET/Libraries/my-password
  ```

- Sua organização empresarial

```
/Finance/Accountants/UserList

/Finance/Analysts/UserList

/HR/Employees/EU/UserList
```

As hierarquias de parâmetros padronizam a maneira de criar parâmetros e facilitam o gerenciamento de parâmetros ao longo do tempo. Uma hierarquia de parâmetros também pode ajudar você a identificar o parâmetro correto para uma tarefa de configuração. Isso ajuda você a evitar a criação de vários parâmetros com os mesmos dados de configuração.

Você pode criar uma hierarquia que permite compartilhar parâmetros em diferentes ambientes, como mostram os exemplos a seguir, que usam senhas em ambientes de desenvolvimento e teste.

`/DevTest/MyApp/database/my-password`

Você poderia então criar uma senha exclusiva para o seu ambiente de produção, conforme mostrado no exemplo a seguir:

`/prod/MyApp/database/my-password`

Não é necessário especificar uma hierarquia de parâmetros. Você pode criar parâmetros no nível um. Estes são chamados de parâmetros raiz. Por motivo de compatibilidade com versões anteriores, todos os parâmetros criados no Parameter Store antes do lançamento de hierarquias são parâmetros raiz. Os sistemas tratam ambos os parâmetros a seguir como parâmetros raiz.

`/parameter-name`

`parameter-name`

Consultar parâmetros em uma hierarquia

Outro benefício de usar hierarquias é a capacidade de consultar todos os parâmetros sob um determinado nível em uma hierarquia usando a operação da API GetParametersByPath. Por exemplo, se você executar o seguinte comando na AWS Command Line Interface (AWS CLI), o sistema retornará todos os parâmetros no nível do `Oncall`.

```
aws ssm get-parameters-by-path --path /Dev/Web/Oncall
```

Exemplo de resultado:

```
{
    "Parameters": [
        {
            "Name": "/Dev/Web/Oncall/Week1",
            "Type": "String",
            "Value": "John Doe",
            "Version": 1,
            "LastModifiedDate": "2024-11-22T07:18:53.510000-08:00",
            "ARN": "arn:aws:ssm:us-east-2:123456789012:parameter/Dev/Web/Oncall/Week1",
            "DataType": "text"
        },
        {
            "Name": "/Dev/Web/Oncall/Week2",
            "Type": "String",
            "Value": "Mary Major",
            "Version": 1,
            "LastModifiedDate": "2024-11-22T07:21:25.325000-08:00",
            "ARN": "arn:aws:ssm:us-east-2:123456789012:parameter/Dev/Web/Oncall/Week2",
            "DataType": "text"
        }
    ]
}
```

Para visualizar os parâmetros de `SecureString` descriptografados em uma hierarquia, você especifica o caminho e o parâmetro `--with-decryption`, como mostra o exemplo a seguir.

```
aws ssm get-parameters-by-path --path /Prod/ERP/SAP --with-decryption
```

Gerenciar parâmetros com hierarquias usando a AWS CLI

Esta demonstração explica como trabalhar com parâmetros e hierarquias de parâmetros usando a AWS CLI.

Para gerenciar parâmetros usando hierarquias

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.
2. Execute o seguinte comando para criar um parâmetro que usa o parâmetro `allowedPattern` e o tipo de parâmetro `String`. O padrão permitido neste exemplo significa que o valor para o parâmetro deve ter entre 1 e 4 dígitos.

Linux & macOS

```
aws ssm put-parameter \
    --name "/MyService/Test/MaxConnections" \
    --value 100 --allowed-pattern "\d{1,4}" \
    --type String
```

Windows

```
aws ssm put-parameter ^
    --name "/MyService/Test/MaxConnections" ^
    --value 100 --allowed-pattern "\d{1,4}" ^
    --type String
```

O comando retornará o número da versão do parâmetro.

3. Execute o seguinte comando para tentar substituir o parâmetro que você acabou de criar por um novo valor.

Linux & macOS

```
aws ssm put-parameter \
    --name "/MyService/Test/MaxConnections" \
    --value 10,000 \
    --type String \
    --overwrite
```

Windows

```
aws ssm put-parameter ^
    --name "/MyService/Test/MaxConnections" ^
    --value 10,000 ^
    --type String ^
    --overwrite
```

O sistema lança o seguinte erro porque o novo valor não atende aos requisitos do padrão permitido que você especificou na etapa anterior.

```
An error occurred (ParameterPatternMismatchException) when calling the PutParameter
 operation: Parameter value, cannot be validated against allowedPattern: \d{1,4}
```

4. Execute o comando a seguir para criar um parâmetro `SecureString` que usa uma Chave gerenciada pela AWS. O padrão permitido neste exemplo significa que o usuário pode especificar qualquer caractere, e o valor deve estar entre 8 e 20 caracteres.

   Linux & macOS

   ```
   aws ssm put-parameter \
       --name "/MyService/Test/my-password" \
       --value "p#sW*rd33" \
       --allowed-pattern ".{8,20}" \
       --type SecureString
   ```

   Windows

   ```
   aws ssm put-parameter ^
       --name "/MyService/Test/my-password" ^
       --value "p#sW*rd33" ^
       --allowed-pattern ".{8,20}" ^
       --type SecureString
   ```

5. Execute os seguintes comandos para criar mais parâmetros que usam a estrutura hierárquica da etapa anterior.

   Linux & macOS

   ```
   aws ssm put-parameter \
       --name "/MyService/Test/DBname" \
       --value "SQLDevDb" \
       --type String
   ```

   ```
   aws ssm put-parameter \
       --name "/MyService/Test/user" \
       --value "SA" \
       --type String
   ```

   ```
   aws ssm put-parameter \
   ```

```
    --name "/MyService/Test/userType" \
    --value "SQLuser" \
    --type String
```

Windows

```
aws ssm put-parameter ^
    --name "/MyService/Test/DBname" ^
    --value "SQLDevDb" ^
    --type String
```

```
aws ssm put-parameter ^
    --name "/MyService/Test/user" ^
    --value "SA" ^
    --type String
```

```
aws ssm put-parameter ^
    --name "/MyService/Test/userType" ^
    --value "SQLuser" ^
    --type String
```

6. Execute o seguinte comando para obter o valor de dois parâmetros.

   Linux & macOS

   ```
   aws ssm get-parameters \
       --names "/MyService/Test/user" "/MyService/Test/userType"
   ```

   Windows

   ```
   aws ssm get-parameters ^
       --names "/MyService/Test/user" "/MyService/Test/userType"
   ```

7. Execute o comando a seguir para consultar todos os parâmetros sob um único nível.

   Linux & macOS

   ```
   aws ssm get-parameters-by-path \
       --path "/MyService/Test"
   ```

Windows

```
aws ssm get-parameters-by-path ^
    --path "/MyService/Test"
```

8. Execute o seguinte comando para excluir dois parâmetros

   Linux & macOS

   ```
   aws ssm delete-parameters \
       --names "/IADRegion/Dev/user" "/IADRegion/Dev/userType"
   ```

   Windows

   ```
   aws ssm delete-parameters ^
       --names "/IADRegion/Dev/user" "/IADRegion/Dev/userType"
   ```

Impedir o acesso às operações da API Parameter Store

Usando condições específicas do serviço compatíveis com as políticas do Systems Manager para AWS Identity and Access Management (IAM), você pode permitir ou negar explicitamente o acesso às operações e ao conteúdo da API Parameter Store. Ao usar essas condições, você pode permitir que somente determinadas entidades do IAM (usuários e perfis) na sua organização chamem determinadas ações de API ou impedir que determinadas entidades do IAM as executem. Isso inclui ações executadas por meio do console do Parameter Store, da AWS Command Line Interface (AWS CLI) e dos SDKs.

No momento, o Systems Manager oferece suporte a três condições que são específicas do Parameter Store.

Tópicos

- Impedir alterações nos parâmetros existentes usando o ssm:Overwrite
- Impedir a criação ou a atualização de parâmetros que usam uma política de parâmetros usando ssm:Policies
- Impedir o acesso a níveis em um parâmetro hierárquico usando ssm:Recursive

Impedir alterações nos parâmetros existentes usando o **`ssm:Overwrite`**

Use a condição `ssm:Overwrite` para controlar se as entidades do IAM podem atualizar os parâmetros existentes.

No exemplo de política a seguir, a declaração `"Allow"` concede permissão para criar parâmetros ao executar a operação da API `PutParameter` na Conta da AWS 123456789012 na região Leste dos EUA (Ohio) (us-east-2).

Contudo, a declaração `"Deny"` impede que entidades alterem valores de parâmetros existentes porque a opção `Overwrite` é explicitamente negada para a operação `PutParameter`. Assim, as entidades que recebem essa política podem criar parâmetros, mas não fazer alterações nos parâmetros existentes.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:us-east-2:123456789012:parameter/*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "ssm:PutParameter"
            ],
            "Condition": {
                "StringEquals": {
                    "ssm:Overwrite": [
                        "true"
                    ]
                }
            },
            "Resource": "arn:aws:ssm:us-east-2:123456789012:parameter/*"
        }
    ]
```

```
}
```

Impedir a criação ou a atualização de parâmetros que usam uma política de parâmetros usando **`ssm:Policies`**

Use a condição `ssm:Policies` para controlar se as Entidades podem criar parâmetros que incluam uma política de parâmetros, e atualizar os parâmetros existentes que incluam uma política de parâmetros.

No exemplo de política a seguir, a instrução `"Allow"` concede permissão geral para criar parâmetros, mas a instrução `"Deny"` impede que Entidades criem ou atualizem parâmetros que incluam uma política de parâmetros na Conta da AWS 123456789012 na região Leste dos EUA (Ohio) (us-east-2). As Entidades ainda podem criar ou atualizar parâmetros que não tenham sido atribuídos a uma política de parâmetros.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:us-east-2:111122223333:parameter/*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "ssm:PutParameter"
            ],
            "Condition": {
                "StringEquals": {
                    "ssm:Policies": [
                        "true"
                    ]
                }
            },
            "Resource": "arn:aws:ssm:us-east-2:111122223333:parameter/*"
        }
```

```
    ]
}
```

Impedir o acesso a níveis em um parâmetro hierárquico usando **ssm:Recursive**

Use a condição `ssm:Recursive` para controlar se as entidades do IAM podem visualizar ou referenciar níveis em um parâmetro hierárquico. Você pode fornecer ou restringir o acesso a todos os parâmetros além de um nível específico de uma hierarquia.

No exemplo de política a seguir, a declaração `"Allow"` fornece acesso às operações do Parameter Store em todos os parâmetros no caminho `/Code/Departments/Finance/*` para a Conta da AWS 123456789012 na região Leste dos EUA (Ohio) (us-east-2).

Depois disso, a instrução `"Deny"` impede que as Entidades do IAM visualizem ou recuperem dados de parâmetros no nível de`/Code/Departments/*` ou abaixo dele. Contudo, as Entidades ainda podem criar ou atualizar parâmetros no caminho. O exemplo foi construído para ilustrar que negar recursivamente o acesso abaixo de um determinado nível em uma hierarquia de parâmetros tem precedência sobre o acesso mais permissivo na mesma política.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:*"
            ],
            "Resource": "arn:aws:ssm:us-east-2:123456789012:parameter/*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "ssm:GetParametersByPath"
            ],
            "Condition": {
                "StringEquals": {
                    "ssm:Recursive": [
                        "true"
```

```
                    ]
                }
            },
            "Resource": "arn:aws:ssm:us-east-2:123456789012:parameter/Code/
Departments/*"
        }
    ]
}
```

> **Important**
>
> Se um usuário tiver acesso a um caminho, o usuário poderá acessar todos os níveis desse caminho. Por exemplo, se um usuário tiver permissão para acessar um caminho `/a`, ele também pode acessar `/a/b`. Isso é verdadeiro, a menos que o acesso para o parâmetro `/b` no IAM tenha sido explicitamente negado ao usuário, conforme ilustrado acima.

Trabalhar com rótulos de parâmetros no Parameter Store

Um rótulo de parâmetro é um alias definido pelo usuário para ajudar você a gerenciar diferentes versões de um parâmetro. Quando você modifica um parâmetro, o AWS Systems Manager salva automaticamente uma nova versão e incrementa o número da versão em um. Um rótulo pode ajudar você a lembrar-se do objetivo de uma versão de parâmetro quando houver várias versões.

Por exemplo, digamos que você tenha um parâmetro chamado `/MyApp/DB/ConnectionString`. O valor do parâmetro é uma string de conexão a um servidor MySQL em um banco de dados local em um ambiente de teste. Depois de concluir a atualização do aplicativo, você deseja que o parâmetro use uma string de conexão para um banco de dados de produção. Você altera o valor de `/MyApp/DB/ConnectionString`. O Systems Manager cria automaticamente a versão dois com a nova string de conexão. Para ajudar você a lembrar-se do objetivo de cada versão, anexe um rótulo a cada parâmetro. Para a versão um, você anexa o rótulo Test e para a versão dois você anexa o rótulo Production.

Você pode mover rótulos de uma versão de um parâmetro para outra versão. Por exemplo, se você criar a versão 3 do parâmetro `/MyApp/DB/ConnectionString` com uma string de conexão para um novo banco de dados de produção, poderá mover o rótulo Production da versão 2 para a versão 3 do parâmetro.

Os rótulos de parâmetros são uma alternativa leve para tags de parâmetros. Sua organização pode ter diretrizes estritas para tags que devem ser aplicadas a diferentes recursos da AWS. Por outro lado, um rótulo é simplesmente uma associação de texto para uma versão de um parâmetro.

De forma semelhante a tags, você pode consultar parâmetros usando rótulos. Você pode visualizar uma lista de todas as versões de parâmetros específicos que usam o mesmo rótulo, se consultar o conjunto de parâmetros usando a ação da API GetParametersByPath, conforme descrito mais adiante nesta seção.

> **Note**
>
> Se você executar um comando que especifica uma versão de um parâmetro que não existe, o comando falhará. Ele não retornará ao valor mais recente ou padrão do parâmetro.

Requisitos e restrições de rótulos

Os rótulos de parâmetros têm os seguintes requisitos e restrições:

- Uma versão de um parâmetro pode ter no máximo 10 rótulos.
- Você não pode anexar o mesmo rótulo a diferentes versões do mesmo parâmetro. Por exemplo, se a versão 1 do parâmetro tiver o rótulo Production (Produção), você não poderá anexar Production à versão 2.
- Você pode mover um rótulo de uma versão de um parâmetro para outra.
- Você não pode criar um rótulo ao criar um parâmetro. Você deve anexar um rótulo a uma versão específica de um parâmetro.
- Se você não quiser mais usar um rótulo de parâmetro, deverá movê-lo para uma versão diferente de um parâmetro.
- Um rótulo pode ter no máximo 100 caracteres.
- Os rótulos podem conter letras (diferenciando maiúsculas de minúsculas), números, pontos (.), hifens (-) ou sublinhados (_).
- Os rótulos não podem começar com um número, "aws", ou "ssm" (sem diferenciação de maiúsculas de minúsculas). Se um rótulo não atender a esses requisitos, ele não será anexado à versão do parâmetro, e o sistema o exibirá na lista de `InvalidLabels`.

Tópicos

- Trabalhar com rótulos de parâmetros usando o console
- Trabalhar com rótulos de parâmetros usando a AWS CLI

Trabalhar com rótulos de parâmetros usando o console

Esta seção descreve como executar as seguintes tarefas usando o console do Systems Manager.

- Criar um rótulo de parâmetro usando o console
- Visualizar rótulos anexados a um parâmetro usando o console
- Mover um rótulo de parâmetro usando o console
- Excluir rótulos de parâmetros usando o console

Criar um rótulo de parâmetro usando o console

O procedimento a seguir descreve como anexar um rótulo a uma versão específica de um parâmetro existente usando o console do Systems Manager. Você não pode anexar um rótulo ao criar um parâmetro.

Anexar um rótulo à versão mais recente de um parâmetro

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Parameter Store.
3. Escolha o nome de um parâmetro para abrir a página de detalhes desse parâmetro.
4. Escolha a guia History (Histórico).
5. Escolha a versão do parâmetro à qual você deseja anexar um rótulo.
6. Selecione Manage labels (Gerenciar rótulos).
7. Selecione Add new label (Adicionar novo rótulo).
8. Na caixa de texto, insira o nome do rótulo. Para adicionar mais rótulos, escolha Add another label (Adicionar outro rótulo). Você pode anexar um máximo de dez rótulos.
9. Ao concluir, escolha Salvar alterações.

Visualizar rótulos anexados a um parâmetro usando o console

Uma versão de um parâmetro pode ter um máximo de 10 rótulos. O procedimento a seguir descreve como visualizar todos os rótulos anexados a uma versão de um parâmetro usando o console do Systems Manager.

Para visualizar rótulos anexados a uma versão do parâmetro

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Parameter Store.
3. Escolha o nome de um parâmetro para abrir a página de detalhes desse parâmetro.
4. Escolha a guia History (Histórico).
5. Localize a versão ao parâmetro do qual você deseja visualizar todos os rótulos anexados. A coluna Labels (Rótulos) mostra todos os rótulos anexados à versão do parâmetro.

Mover um rótulo de parâmetro usando o console

O procedimento a seguir descreve como mover um rótulo de um parâmetro para outra versão do mesmo parâmetro usando o console do Systems Manager.

Para mover um rótulo para outra versão de um parâmetro usando o console

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Parameter Store.
3. Escolha o nome de um parâmetro para abrir a página de detalhes desse parâmetro.
4. Escolha a guia History (Histórico).
5. Escolha a versão do parâmetro para o qual você deseja mover o rótulo.
6. Selecione Manage labels (Gerenciar rótulos).
7. Selecione Add new label (Adicionar novo rótulo).
8. Na caixa de texto, insira o nome do rótulo.
9. Ao concluir, escolha Salvar alterações.

Excluir rótulos de parâmetros usando o console

O procedimento a seguir descreve como excluir um ou vários rótulos de parâmetros usando o console do Systems Manager.

Para excluir rótulos de um parâmetro

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Parameter Store.
3. Escolha o nome de um parâmetro para abrir a página de detalhes desse parâmetro.
4. Escolha a guia History (Histórico).
5. Escolha a versão do parâmetro para o qual você deseja excluir os rótulos.
6. Selecione Manage labels (Gerenciar rótulos).
7. Selecione Remove (Remover) ao lado de cada rótulo que você quiser excluir.
8. Ao concluir, escolha Salvar alterações.
9. Confirme se as alterações estão corretas, digite `Confirm` na caixa de texto e escolha Confirm (Confirmar).

Trabalhar com rótulos de parâmetros usando a AWS CLI

Esta seção descreve como executar as seguintes tarefas usando a AWS Command Line Interface (AWS CLI).

- Criar um novo rótulo de parâmetro usando o AWS CLI
- Visualizar rótulos de um parâmetro usando a AWS CLI
- Visualizar uma lista de parâmetros atribuídos a um rótulo usando a AWS CLI
- Mover um rótulo de parâmetro usando a AWS CLI
- Excluir rótulos de parâmetros usando a AWS CLI

Criar um novo rótulo de parâmetro usando o AWS CLI

O procedimento a seguir descreve como anexar um rótulo a uma versão específica de um parâmetro existente usando a AWS CLI. Você não pode anexar um rótulo ao criar um parâmetro.

Para criar um novo rótulo de parâmetro

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.
2. Execute o seguinte comando para visualizar uma lista de parâmetros para os quais você tem permissão para anexar um rótulo.

> **Note**
>
> Parâmetros só estão disponíveis na região da Região da AWS em que foram criados. Se não vir um parâmetro ao qual você deseja anexar um rótulo, verifique sua região.

```
aws ssm describe-parameters
```

Anote o nome de um parâmetro ao qual você quer anexar um rótulo.

3. Execute o comando a seguir para visualizar todas as versões do parâmetro.

```
aws ssm get-parameter-history --name "parameter-name"
```

Anote a versão do parâmetro à qual você deseja anexar um rótulo.

4. Execute o comando a seguir para recuperar informações sobre um parâmetro pelo número da versão.

```
aws ssm get-parameters --names "parameter-name:version-number"
```

Aqui está um exemplo.

```
aws ssm get-parameters --names "/Production/SQLConnectionString:3"
```

5. Execute um dos seguintes comandos para anexar um rótulo a uma versão de um parâmetro. Se você anexar vários rótulos, deverá separar os nomes dos rótulos com um espaço.

Anexar um rótulo à versão mais recente de um parâmetro

```
aws ssm label-parameter-version --name parameter-name  --labels label-name
```

Anexar um rótulo a uma versão específica de um parâmetro

```
aws ssm label-parameter-version --name parameter-name --parameter-version version-number --labels label-name
```

Aqui estão alguns exemplos.

```
aws ssm label-parameter-version --name /config/endpoint --labels production east-
region finance
```

```
aws ssm label-parameter-version --name /config/endpoint --parameter-version 3 --
labels MySQL-test
```

> **Note**
>
> Se a saída mostrar o rótulo que você criou na lista `InvalidLabels`, o rótulo não atenderá aos requisitos descritos anteriormente neste tópico. Verifique os requisitos e tente novamente. Se a lista `InvalidLabels` estiver vazia, seu rótulo terá sido aplicado com êxito para a versão do parâmetro.

6. Você pode visualizar os detalhes do parâmetro usando um número de versão ou um nome de rótulo. Execute o seguinte comando e especifique o rótulo que você criou na etapa anterior.

```
aws ssm get-parameter --name parameter-name:label-name --with-decryption
```

O comando retorna informações como as seguintes.

```
{
    "Parameter": {
        "Version": version-number,
        "Type": "parameter-type",
        "Name": "parameter-name",
        "Value": "parameter-value",
        "Selector": ":label-name"
    }
}
```

> **Note**
>
> O seletor na saída é o número da versão ou o rótulo que você especificou no campo de entrada `Name`.

Visualizar rótulos de um parâmetro usando a AWS CLI

Você pode usar a operação da API GetParameterHistory para visualizar todo o histórico e todos os rótulos anexados a um parâmetro específico. Ou, você pode usar a ação da API GetParametersByPath para visualizar uma lista de todos os parâmetros atribuídos a um rótulo específico.

Para visualizar os rótulos de um parâmetro usando a operação da API GetParameterHistory

1. Execute o seguinte comando para visualizar uma lista dos parâmetros dos quais você pode visualizar rótulos.

   > **Note**
   >
   > Parâmetros só estão disponíveis na região em que foram criados. Se não vir um parâmetro para o qual você deseja mover um rótulo, verifique sua região.

   ```
   aws ssm describe-parameters
   ```

   Anote o nome do parâmetro que contém os rótulos que você deseja exibir.

2. Execute o comando a seguir para visualizar todas as versões do parâmetro.

   ```
   aws ssm get-parameter-history --name parameter-name --with-decryption
   ```

   O sistema retorna informações como estas.

   ```
   {
       "Parameters": [
           {
               "Name": "/Config/endpoint",
               "LastModifiedDate": 1528932105.382,
               "Labels": [
                   "Deprecated"
               ],
               "Value": "MyTestService-June-Release.example.com",
               "Version": 1,
               "LastModifiedUser": "arn:aws:iam::123456789012:user/test",
               "Type": "String"
   ```

```
        },
        {
            "Name": "/Config/endpoint",
            "LastModifiedDate": 1528932111.222,
            "Labels": [
                "Current"
            ],
            "Value": "MyTestService-July-Release.example.com",
            "Version": 2,
            "LastModifiedUser": "arn:aws:iam::123456789012:user/test",
            "Type": "String"
        }
    ]
}
```

Visualizar uma lista de parâmetros atribuídos a um rótulo usando a AWS CLI

Você pode usar a operação da API GetParametersByPath para visualizar uma lista de todos os parâmetros em um caminho, que estiverem atribuídos a um rótulo específico.

Execute o seguinte comando para visualizar uma lista de parâmetros em um caminho atribuídos a um rótulo específico. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

```
aws ssm get-parameters-by-path \
    --path parameter-path \
    --parameter-filters Key=Label,Values=label-name,Option=Equals \
    --max-results a-number \
    --with-decryption --recursive
```

O sistema retorna informações como estas. Para este exemplo, o usuário pesquisou no caminho `/Config`.

```
{
    "Parameters": [
        {
            "Version": 3,
            "Type": "SecureString",
            "Name": "/Config/DBpwd",
            "Value": "MyS@perGr&pass33"
        },
```

```
        {
            "Version": 2,
            "Type": "String",
            "Name": "/Config/DBusername",
            "Value": "TestUserDB"
        },
        {
            "Version": 2,
            "Type": "String",
            "Name": "/Config/endpoint",
            "Value": "MyTestService-July-Release.example.com"
        }
    ]
}
```

Mover um rótulo de parâmetro usando a AWS CLI

O procedimento a seguir descreve como mover um rótulo de um parâmetro para outra versão do mesmo parâmetro.

Para mover um rótulo de um parâmetro

1. Execute o comando a seguir para visualizar todas as versões do parâmetro. Substitua *nome do parâmetro* pelas suas próprias informações.

   ```
   aws ssm get-parameter-history \
       --name "parameter name"
   ```

   Observe as versões do parâmetro para o qual você deseja mover o rótulo do e do.

2. Execute o seguinte comando para atribuir um rótulo existente para uma versão diferente de um parâmetro. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   ```
   aws ssm label-parameter-version \
       --name parameter name \
       --parameter-version version number \
       --labels name-of-existing-label
   ```

> **Note**
>
> Se você desejar mover um rótulo existente para a versão mais recente de um parâmetro, remova `--parameter-version` do comando.

Excluir rótulos de parâmetros usando a AWS CLI

O procedimento a seguir descreve como excluir rótulos de parâmetros usando aAWS CLI.

Para excluir um parameter group

1. Execute o comando a seguir para visualizar todas as versões do parâmetro. Substitua *nome do parâmetro* pelas suas próprias informações.

```
aws ssm get-parameter-history \
    --name "parameter name"
```

O sistema retorna informações como estas.


```
{
    "Parameters": [
        {
            "Name": "foo",
            "DataType": "text",
            "LastModifiedDate": 1607380761.11,
            "Labels": [
                "l3",
                "l2"
            ],
            "Value": "test",
            "Version": 1,
            "LastModifiedUser": "arn:aws:iam::123456789012:user/test",
            "Policies": [],
            "Tier": "Standard",
            "Type": "String"
        },
        {
            "Name": "foo",
            "DataType": "text",
            "LastModifiedDate": 1607380763.11,
```

```
            "Labels": [
                "l1"
            ],
            "Value": "test",
            "Version": 2,
            "LastModifiedUser": "arn:aws:iam::123456789012:user/test",
            "Policies": [],
            "Tier": "Standard",
            "Type": "String"
        }
    ]
}
```

Anote a versão do parâmetro à qual você deseja anexar um rótulo.

2. Execute o comando a seguir para excluir os rótulos escolhidos desse parâmetro. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

```
aws ssm unlabel-parameter-version \
    --name parameter name \
    --parameter-version version \
    --labels label 1,label 2,label 3
```

O sistema retorna informações como estas.

```
{
    "InvalidLabels": ["invalid"],
    "DeletedLabels" : ["Prod"]
 }
```

Como trabalhar com versões de parâmetros no Parameter Store

Toda vez que você edita o valor de um parâmetro, o Parameter Store, uma ferramenta do AWS Systems Manager, cria uma nova versão do parâmetro e retém as versões anteriores. Quando você cria um parâmetro, o Parameter Store atribui a versão 1 a esse parâmetro. Quando você altera o valor do parâmetro, o Parameter Store incrementa automaticamente em um o número da versão. Você pode visualizar os detalhes, incluindo os valores, de todas as versões no histórico de um parâmetro.

Você também pode especificar a versão de um parâmetro a ser usado em comandos da API e documentos do SSM; por exemplo: `ssm:MyParameter:3`. Você pode especificar o nome de um parâmetro e um número de versão específico em chamadas de API e em documentos do SSM. Se não especificar um número de versão, o sistema usará automaticamente a versão mais recente. Se você especificar o número para uma versão que não existe, o sistema retornará um erro em vez de retornar para a versão mais recente ou padrão do parâmetro.

Também é possível usar versões de parâmetros para ver quantas vezes um parâmetro mudou ao longo de um período. As versões de parâmetros também fornecem uma camada de proteção quando o valor de um parâmetro é alterado acidentalmente.

Você pode criar e manter um máximo de 100 versões de um parâmetro. Depois de criar 100 versões de um parâmetro, cada vez que você criar uma nova versão, a versão mais antiga do parâmetro será removida do histórico para abrir espaço para a nova versão.

Uma exceção a isso é quando já existem 100 versões de parâmetro no histórico e um rótulo de parâmetro é atribuído à versão mais antiga de um parâmetro. Nesse caso, essa versão não é removida do histórico e a solicitação para criar uma nova versão do parâmetro falha. Essa proteção é para evitar que versões de parâmetros com rótulos de missão crítica atribuídos a eles sejam excluídas. Para continuar criando novos parâmetros, primeiro mova o rótulo da versão mais antiga do parâmetro para um mais recente para uso em suas operações. Para obter informações sobre como mover rótulos de parâmetros, consulte Mover um rótulo de parâmetro usando o console e Mover um rótulo de parâmetro usando a AWS CLI.

Os procedimentos a seguir mostram como editar um parâmetro e, em seguida, verificar se uma nova versão foi criada. Você pode usar o`get-parameter`e`get-parameters`para visualizar versões de parâmetros. Para obter exemplos sobre como usar esses comandos, consulte GetParameter e GetParameters na Referência da API do AWS Systems Manager

Tópicos

- Criar uma nova versão de um parâmetro usando o console
- Fazer referência à versão de um parâmetro

Criar uma nova versão de um parâmetro usando o console

É possível usar o console do Systems Manager para criar uma versão de um parâmetro e visualizar o histórico de versões de um parâmetro.

Para criar uma versão de uma parâmetro

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Parameter Store.
3. Escolha o nome do parâmetro que você criou anteriormente. Para obter mais informações sobre como criar um novo parâmetro, consulte Criar parâmetros do Parameter Store no Systems Manager.
4. Escolha Editar.
5. Na caixa Value (Valor), digite um novo valor e escolha Save changes (Salvar alterações).
6. Escolha o nome do parâmetro que você acabou de atualizar. Na guia Overview (Visão geral), verifique se o número da versão aumentou com um incremento de 1 e examine o novo valor.
7. Para exibir o histórico de todas as versões de um parâmetro, escolha a guia History (Histórico).

Fazer referência à versão de um parâmetro

É possível fazer referência a versões de parâmetros específicos em comandos, chamadas de API e documentos do SSM usando o seguinte formato: ssm:*parameter-name*:*version-number*.

No exemplo a seguir, o Amazon Elastic Compute Cloud (Amazon EC2)`run-instances command`O usa a versão 3 do parâmetro`golden-ami`.

Linux & macOS

```
aws ec2 run-instances \
    --image-id resolve:ssm:/golden-ami:3 \
    --count 1 \
    --instance-type t2.micro \
    --key-name my-key-pair \
    --security-groups my-security-group
```

Windows

```
aws ec2 run-instances ^
    --image-id resolve:ssm:/golden-ami:3 ^
    --count 1 ^
    --instance-type t2.micro ^
    --key-name my-key-pair ^
    --security-groups my-security-group
```

> **Note**
>
> O uso de `resolve` e um valor de parâmetro funciona somente com a opção `--image-id` e um parâmetro que contém uma Amazon Machine Image (AMI) como seu valor. Para ter mais informações, consulte Como usar o suporte a parâmetros nativos no Parameter Store para IDs de imagem de máquina da Amazon.

Veja a seguir um exemplo para especificar a versão 2 de um parâmetro nomeado `MyRunCommandParameter` em um documento SSM.

YAML

```
---
schemaVersion: '2.2'
description: Run a shell script or specify the commands to run.
parameters:
  commands:
    type: String
    description: "(Required) Specify a shell script or a command to run."
    displayType: textarea
    default: "{{ssm:MyRunCommandParameter:2}}"
mainSteps:
- action: aws:runShellScript
  name: RunScript
  inputs:
    runCommand:
    - "{{commands}}"
```

JSON

```
{
    "schemaVersion": "2.2",
    "description": "Run a shell script or specify the commands to run.",
    "parameters": {
        "commands": {
            "type": "String",
            "description": "(Required) Specify a shell script or a command to run.",
            "displayType": "textarea",
            "default": "{{ssm:MyRunCommandParameter:2}}"
        }
    },
```

```
    "mainSteps": [
        {
            "action": "aws:runShellScript",
            "name": "RunScript",
            "inputs": {
                "runCommand": [
                    "{{commands}}"
                ]
            }
        }
    ]
}
```

Trabalhar com parâmetros compartilhados no Parameter Store

O compartilhamento de parâmetros avançados simplifica o gerenciamento de dados de configuração em um ambiente com várias contas. É possível armazenar e gerenciar os parâmetros da AMI de forma centralizada e compartilhá-los com outras Contas da AWS que precisam referenciá-los.

O Parameter Store se integra ao AWS Resource Access Manager (AWS RAM) para permitir o compartilhamento avançado de parâmetros. O AWS RAM é um serviço que permite compartilhar recursos com outras Contas da AWS ou via AWS Organizations.

Com o AWS RAM, você compartilha recursos que possui criando um compartilhamento de recursos. Um compartilhamento de atributos especifica os atributos a serem compartilhados, as permissões a serem concedidas e os consumidores com quem o compartilhamento será feito. Os consumidores podem incluir:

- Contas específicas da Contas da AWS dentro ou fora de sua organização no AWS Organizations
- Uma unidade organizacional dentro da organização no AWS Organizations
- Toda a organização no AWS Organizations

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

Este tópico explica como compartilhar parâmetros que você possui e como usar os parâmetros que são compartilhados com você.

Conteúdo

- Pré-requisitos para compartilhar parâmetros

- Compartilhar um parâmetro
- Parar de compartilhar um parâmetro
- Identificar parâmetros compartilhados
- Acessar parâmetros compartilhados
- Conjuntos de permissões para compartilhamento de parâmetros
- Throughput máximo para parâmetros compartilhados
- Preços para parâmetros compartilhados
- Acesso entre contas para Contas da AWS encerradas

Pré-requisitos para compartilhar parâmetros

Os pré-requisitos a seguir devem ser atendidos para que você possa compartilhar parâmetros da sua conta:

- Para compartilhar um recurso, é necessário ser seu proprietário em sua Conta da AWS. Não é possível compartilhar um parâmetro que foi compartilhado com você.
- Para compartilhar um parâmetro, ele deve estar no nível de parâmetros avançados. Para obter informações sobre níveis de parâmetros, consulte Gerenciar camadas de parâmetros. Para obter informações sobre como alterar um parâmetro padrão existente para um parâmetro avançado, consulte Alterar um parâmetro padrão para um parâmetro avançado.
- Para compartilhar um parâmetro `SecureString`, ele deve ser criptografado com uma chave gerenciada pelo cliente e você deve compartilhar a chave separadamente via AWS Key Management Service. Não é possível compartilhar Chaves gerenciadas pela AWS. Parâmetros criptografados com a Chave gerenciada pela AWS padrão podem ser atualizados para usar uma chave gerenciada pelo cliente. Para obter as definições de chaves do AWS KMS, consulte Conceitos do AWS KMS no Guia do desenvolvedor do AWS Key Management Service.
- Para compartilhar um parâmetro com a sua organização ou com uma unidade organizacional no AWS Organizations, é necessário habilitar o compartilhamento com o AWS Organizations. Para obter mais informações, consulte Habilitar o compartilhamento com o AWS Organizations no Guia do usuário do AWS RAM.

Compartilhar um parâmetro

Para compartilhar um parâmetro, é necessário adicioná-lo a um compartilhamento de recursos. Um compartilhamento de recursos é um recurso do AWS RAM que permite que você compartilhe seus

recursos entre Contas da AWS. Um compartilhamento de recursos especifica os recursos a serem compartilhados, e os clientes com os quais compartilhá-los.

Ao compartilhar um parâmetro pertencente a você com outra Contas da AWS, você pode escolher entre duas permissões gerenciadas pela AWS para conceder aos consumidores. Para obter mais informações, consulte Conjuntos de permissões para compartilhamento de parâmetros.

Se você fizer parte de uma organização no AWS Organizations e o compartilhamento estiver habilitado na organização, será possível conceder aos consumidores da organização o acesso a partir do console do AWS RAM para o parâmetro compartilhado. Caso contrário, os consumidores receberão um convite para participar do compartilhamento de recursos e acesso ao parâmetro compartilhado depois de aceitar o convite.

É possível compartilhar um parâmetro pertencente a você usando o console do AWS RAM ou a AWS CLI.

> **Note**
>
> Embora você possa compartilhar um parâmetro usando a operação da API PutResourcePolicy do Systems Manager, recomendamos usar o AWS Resource Access Manager (AWS RAM). Isso ocorre porque o uso de `PutResourcePolicy` exige a etapa extra de promover o parâmetro para um compartilhamento de recursos padrão usando a operação da API PromoteResourceShareCreatedFromPolicy do AWS RAM. Caso contrário, o parâmetro não será retornado pela operação da API DescribeParameters do Systems Manager usando a opção `--shared`.

Para compartilhar um parâmetro pertencente a você usando o console do AWS RAM

Consulte Criar um compartilhamento de recursos no AWS RAM no Guia do usuário do AWS RAM.

Faça as seguintes seleções ao concluir o procedimento:

- Na página Etapa 1, em Recursos, selecione `Parameter Store Advanced Parameter` e, em seguida, selecione a caixa de cada parâmetro na camada de parâmetros avançados que deseja compartilhar.
- Na página Etapa 2, em Permissões gerenciadas, escolha a permissão a ser concedida aos consumidores, conforme descrito em Conjuntos de permissões para compartilhamento de parâmetros mais adiante neste tópico.

Escolha outras opções com base em seus objetivos de compartilhamento de parâmetros.

Para compartilhar um parâmetro pertencente a você usando o console do AWS CLI

Use o comando create-resource-share para adicionar parâmetros a um novo compartilhamento de recursos.

Use o comando associate-resource-share para adicionar parâmetros a um compartilhamento de recursos existente.

O exemplo a seguir cria um novo compartilhamento de recursos para compartilhar parâmetros com consumidores em uma organização e em uma conta individual.

```
aws ram create-resource-share \
    --name "MyParameter" \
    --resource-arns "arn:aws:ssm:us-east-2:123456789012:parameter/MyParameter" \
    --principals "arn:aws:organizations::123456789012:ou/o-63bEXAMPLE/ou-46xi-rEXAMPLE"
 "987654321098"
```

Parar de compartilhar um parâmetro

Quando você para de compartilhar um parâmetro, a conta do consumidor não pode mais acessar o parâmetro.

Para interromper o compartilhamento de um parâmetro pertencente a você, remova-o do compartilhamento de recursos. Isso pode ser feito usando o console do Systems Manager, o console do AWS RAM ou a AWS CLI.

Para parar de compartilhar um parâmetro pertencente a você usando o console do AWS RAM

Consulte Atualizar um compartilhamento de recursos no AWS RAM no Guia do usuário do AWS RAM.

Para parar de compartilhar um parâmetro pertencente a você usando a AWS CLI

Use o comando disassociate-resource-share.

Identificar parâmetros compartilhados

Os proprietários e consumidores podem identificar parâmetros compartilhados usando a AWS CLI.

Para identificar parâmetros compartilhados usando a AWS CLI

Para identificar parâmetros compartilhados usando a AWS CLI, você pode escolher entre o comando `describe-parameters` do Systems Manager e o comando `list-resources` do AWS RAM.

Quando você usa a opção `--shared` com `describe-parameters`, o comando retorna os parâmetros que são compartilhados com você.

Veja um exemplo a seguir:

```
aws ssm describe-parameters --shared
```

Acessar parâmetros compartilhados

Os consumidores podem acessar parâmetros compartilhados usando as ferramentas de linha de comando da AWS e os AWS SDKs. Para contas de consumidores, os parâmetros compartilhados com essa conta não são incluídos na página Meus parâmetros.

Exemplo de CLI: acessar detalhes de parâmetros compartilhados usando a AWS CLI

Para acessar detalhes de parâmetros compartilhados usando a AWS CLI, é possível usar os comandos get-parameter ou get-parameters. É necessário especificar o ARN completo do parâmetro como `--name` para recuperar o parâmetro de outra conta.

Veja um exemplo a seguir.

```
aws ssm get-parameter \
    --name arn:aws:ssm:us-east-2:123456789012:parameter/MySharedParameter
```

Integrações com ou sem suporte para parâmetros compartilhados

No momento, é possível usar parâmetros compartilhados nos seguintes cenários de integração:

- Parâmetros de modelos do AWS CloudFormation
- A extensão do Lambda AWS Parameters and Secrets
- Modelos de execução do Amazon Elastic Compute Cloud (EC2)
- Valores para `ImageID` com o comando RunInstances do EC2 para criar instâncias com base em uma Amazon Machine Image (AMI).
- Recuperar valores de parâmetros em runbooks para o Automation, uma ferramenta do Systems Manager

No momento, os cenários e serviços integrados a seguir não oferecem suporte ao uso de parâmetros compartilhados:

- Parâmetros em comandos em Run Command, uma ferramenta do Systems Manager
- Referências dinâmicas do AWS CloudFormation
- Os valores das variáveis de ambiente no AWS CodeBuild
- Os valores das variáveis de ambiente no AWS App Runner
- O valor de um segredo no Amazon Elastic Continer Service

Conjuntos de permissões para compartilhamento de parâmetros

As contas de consumidor recebem acesso somente leitura aos parâmetros que você compartilha com elas. O consumidor não pode atualizar nem excluir o parâmetro. O consumidor não pode compartilhar o parâmetro com uma terceira conta.

Ao criar um compartilhamento de recursos no AWS Resource Access Manager para compartilhar seus parâmetros, é possível escolher entre dois conjuntos de permissões gerenciados pela AWS para conceder esse acesso somente leitura:

AWSRAMDefaultPermissionSSMParameterReadOnly

Ações permitidas: `DescribeParameters`, `GetParameter`, `GetParameters`

AWSRAMPermissionSSMParameterReadOnlyWithHistory

Ações permitidas: `DescribeParameters`, `GetParameter`, `GetParameters`, `GetParameterHistory`

Ao seguir as etapas em Criar um compartilhamento de recursos no AWS RAM no Guia do usuário do AWS RAM, escolha `Parameter Store Advanced Parameters` como o tipo de recurso e qualquer uma dessas permissões gerenciadas, dependendo se você deseja que os usuários visualizem o histórico de parâmetros ou não.

> **Note**
>
> Se você estiver recuperando parâmetros compartilhados programaticamente (por exemplo, usando o AWS Lambda), talvez seja necessário adicionar as permissões

`ssm:GetResourcePolicies` e `ssm:PutResourcePolicy` a qualquer perfil do IAM que chame ações de API do AWS Resource Access Manager.

Throughput máximo para parâmetros compartilhados

O Systems Manager limita o throughput máximo (transações por segundo) para as operações GetParameter e GetParameters. O throughput é aplicado em nível de conta individual. Portanto, cada conta que consome um parâmetro compartilhado pode usar seu throughput máximo permitido sem ser afetada por outras contas. Para obter mais informações sobre o throughput máximo para parâmetros, consulte os tópicos a seguir:

- Aumentar o throughput do Parameter Store
- Cotas do Systems Manager Service no Referência geral da Amazon Web Services.

Preços para parâmetros compartilhados

O compartilhamento entre contas só está disponível no nível de parâmetros avançados. Para parâmetros avançados, as cobranças são cobradas de acordo com o preço atual do armazenamento e do uso da API para cada parâmetro avançado. A conta proprietária é cobrada pelo armazenamento do parâmetro avançado. Qualquer conta consumidora que faça uma chamada de API para um parâmetro avançado compartilhado será cobrada pelo uso do parâmetro.

Por exemplo, se a Conta A criar um parâmetro avançado, `MyAdvancedParameter`, ela será cobrada em USD 0,05 por mês para armazenar o parâmetro.

A Conta A então compartilha `MyAdvancedParameter` com a Conta B e a Conta C. Durante um mês, as três contas fazem chamadas para `MyAdvancedParameter`. A tabela a seguir ilustra as cobranças em que eles incorreriam pelo número de chamadas feitas por cada um.

> **Note**
>
> As cobranças na tabela a seguir são apenas para fins ilustrativos. Para verificar os preços atuais, consulte Preços do AWS Systems Manager para Parameter Store.

Conta	Número de chamadas	Cobranças
Conta A (conta proprietária)	10.000 chamadas	• Armazenamento avançado de parâmetros em um mês: USD 0,05 • 10.000 chamadas para `MyAdvancedParameter` : USD 0,05 • Total: USD 0,10
Conta B (conta consumidora)	20.000 chamadas	• 20.000 chamadas para `MyAdvancedParameter` : USD 0,10 • Total: USD 0,10
Conta C (conta consumidora)	30.000 chamadas	• 30.000 chamadas para `MyAdvancedParameter` : USD 0,15 • Total: USD 0,15

Acesso entre contas para Contas da AWS encerradas

Se a Conta da AWS proprietária de um parâmetro compartilhado for encerrada, todas as contas consumidoras perderão o acesso ao parâmetro compartilhado. Se a conta proprietária for reaberta dentro de 90 dias após seu encerramento, as contas consumidoras recuperarão o acesso aos parâmetros compartilhados anteriormente. Para obter mais informações sobre a reabertura de uma conta durante o período pós-encerramento, consulte Acessar sua Conta da AWS após encerrá-la no Guia de referência do AWS Gerenciamento de contas.

Trabalhar com parâmetros no Parameter Store usando comandos do Run Command

É possível trabalhar com parâmetros no Run Command, uma ferramenta do AWS Systems Manager. Para obter mais informações, consulte AWS Systems Manager Run Command.

Executar um parâmetro String usando o console

O procedimento a seguir explica o processo de criação de um parâmetro no `String` e, subsequentemente, a execução de um comando que usa esse parâmetro.

Para criar um parâmetro String usando o Parameter Store

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Run Command.
3. Selecione Run command.
4. Na lista Command document (Documento de comandos), escolha `AWS-RunPowerShellScript` (Windows) ou `AWS-RunShellScript` (Linux).
5. Em Command parameters (Parâmetros de comando), insira **`echo {{ssm:`*`parameter-name`*`}}`**. Por exemplo: **`echo {{ssm:/Test/helloWorld}}`**.
6. Na seção Targets (Destinos), escolha os nós gerenciados nos quais você quer executar essa operação, especificando as etiquetas, selecionando as instâncias ou dispositivos de borda manualmente ou especificando um grupo de recursos.

   > **Tip**
   >
   > Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

7. Para Other parameters (Outros parâmetros):
   - Em Comment (Comentário), digite as informações sobre esse comando.
   - Em Timeout (seconds) (Tempo limite [segundos]), especifique o número de segundos para o sistema aguardar até a falha de execução do comando total.
8. Para Rate control (Controle de taxa):
   - Em Concurrency (Concorrência), especifique um número ou uma porcentagem de nós gerenciados nos quais executar o comando ao mesmo tempo.

     > **Note**
     >
     > Se você selecionou destinos especificando tags aplicadas a instâncias a nós gerenciados ou especificando grupos de recursos da AWS, e não tiver certeza de quantas instâncias são direcionadas, restrinja o número de instâncias que poderão executar o documento ao mesmo tempo, especificando uma porcentagem.

- Em Error threshold (Limite de erro), especifique quando parar de executar o comando em outros nós depois de falhar em alguns ou em uma porcentagem de nós. Por exemplo, se você especificar três erros, o Systems Manager deixará de enviar o comando quando o 4° erro for recebido. Os nós gerenciados que continuam processando o comando também podem enviar erros.

9. (Opcional) Em Output options (Opções de saída), para salvar a saída do comando em um arquivo, selecione a caixa Write command output to an S3 bucket (Gravar saída do comando em um bucket do S3). Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

   > **Note**
   >
   > As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil de instância (para instâncias do EC2) ou perfil de serviço do IAM (máquinas ativadas para ambientes híbridos) atribuído à instância, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço do IAM associado ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

10. Na seção SNS notifications (Notificações do SNS), se quiser enviar notificações sobre o status da execução do comando, marque a caixa de seleção Enable SNS notifications (Habilitar notificações do SNS).

    Para obter mais informações sobre a configuração de notificações do Amazon SNS para o Run Command, consulte Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS.

11. Escolha Executar.
12. Na página Command ID (ID do comando), na área Targets and outputs (Destinos e saídas), selecione o botão ao lado do ID de um nó em que você executou o comando e escolha View output (Exibir saída). Verifique se a saída do comando é o valor fornecido para o parâmetro, como **`This is my first parameter`**.

Executar um parâmetro usando a AWS CLI

Exemplo 1: comando simples

O comando de exemplo a seguir inclui um parâmetro do Systems Manager chamado `DNS-IP`. O valor desse parâmetro é simplesmente o endereço IP de um nó. Esse exemplo usa um comando da AWS Command Line Interface (AWS CLI) para fazer ecoar o valor do parâmetro.

Linux & macOS

```
aws ssm send-command \
    --document-name "AWS-RunShellScript" \
    --document-version "1" \
    --targets "Key=instanceids,Values=i-02573cafcfEXAMPLE" \
    --parameters "commands='echo {{ssm:DNS-IP}}'" \
    --timeout-seconds 600 \
    --max-concurrency "50" \
    --max-errors "0" \
    --region us-east-2
```

Windows

```
aws ssm send-command ^
    --document-name "AWS-RunPowerShellScript" ^
    --document-version "1" ^
    --targets "Key=instanceids,Values=i-02573cafcfEXAMPLE" ^
    --parameters "commands='echo {{ssm:DNS-IP}}'" ^
    --timeout-seconds 600 ^
    --max-concurrency "50" ^
    --max-errors "0" ^
    --region us-east-2
```

O comando retorna informações como as seguintes.

```
{
    "Command": {
        "CommandId": "c70a4671-8098-42da-b885-89716EXAMPLE",
        "DocumentName": "AWS-RunShellScript",
        "DocumentVersion": "1",
        "Comment": "",
        "ExpiresAfter": "2023-12-26T15:19:17.771000-05:00",
        "Parameters": {
            "commands": [
                "echo {{ssm:DNS-IP}}"
            ]
```

```
        },
        "InstanceIds": [],
        "Targets": [
            {
                "Key": "instanceids",
                "Values": [
                    "i-02573cafcfEXAMPLE"
                ]
            }
        ],
        "RequestedDateTime": "2023-12-26T14:09:17.771000-05:00",
        "Status": "Pending",
        "StatusDetails": "Pending",
        "OutputS3Region": "us-east-2",
        "OutputS3BucketName": "",
        "OutputS3KeyPrefix": "",
        "MaxConcurrency": "50",
        "MaxErrors": "0",
        "TargetCount": 0,
        "CompletedCount": 0,
        "ErrorCount": 0,
        "DeliveryTimedOutCount": 0,
        "ServiceRole": "",
        "NotificationConfig": {
            "NotificationArn": "",
            "NotificationEvents": [],
            "NotificationType": ""
        },
        "CloudWatchOutputConfig": {
            "CloudWatchLogGroupName": "",
            "CloudWatchOutputEnabled": false
        },
        "TimeoutSeconds": 600,
        "AlarmConfiguration": {
            "IgnorePollAlarmFailure": false,
            "Alarms": []
        },
        "TriggeredAlarms": []
    }
}
```

Depois que a execução de um comando for concluída, você poderá ver mais informações sobre ele usando os seguintes comandos:

- get-command-invocation: visualize informações detalhadas sobre a execução do comando.
- list-command-invocations: visualize o status de execução do comando em um nó gerenciado específico.
- list-commands: visualize o status de execução do comando nos nós gerenciados.

Exemplo 2: descriptografar um valor de parâmetro **`SecureString`**

O próximo comando de exemplo usa um parâmetro `SecureString` chamado SecurePassword. O comando usado no campo `parameters` recupera e descriptografa o valor do parâmetro `SecureString` e redefine a senha do administrador local sem a necessidade de transmitir a senha em texto simples.

Linux

```
aws ssm send-command \
        --document-name "AWS-RunShellScript" \
        --document-version "1" \
        --targets "Key=instanceids,Values=i-02573cafcfEXAMPLE" \
        --parameters '{"commands":["secure=$(aws ssm get-parameters --names
 SecurePassword --with-decryption --query Parameters[0].Value --output text --region
 us-east-2)","echo $secure | passwd myuser --stdin"]}' \
        --timeout-seconds 600 \
        --max-concurrency "50" \
        --max-errors "0" \
        --region us-east-2
```

Windows

```
aws ssm send-command ^
        --document-name "AWS-RunPowerShellScript" ^
        --document-version "1" ^
        --targets "Key=instanceids,Values=i-02573cafcfEXAMPLE" ^
        --parameters "commands=['$secure = (Get-SSMParameterValue -Names
 SecurePassword -WithDecryption $True).Parameters[0].Value','net user administrator
 $secure']" ^
        --timeout-seconds 600 ^
        --max-concurrency "50" ^
        --max-errors "0" ^
        --region us-east-2
```

Exemplo 3: referenciar um parâmetro em um documento do SSM

Você também pode fazer referência a parâmetros do Systems Manager na seção Parameters de um documento do SSM, conforme mostrado no exemplo a seguir.

```
{
   "schemaVersion":"2.0",
   "description":"Sample version 2.0 document v2",
   "parameters":{
      "commands" : {
        "type": "StringList",
        "default": ["{{ssm:parameter-name}}"]
      }
    },
    "mainSteps":[
       {
          "action":"aws:runShellScript",
          "name":"runShellScript",
          "inputs":{
             "runCommand": "{{commands}}"
          }
       }
    ]
}
```

Não confunda a sintaxe semelhante de parâmetros locais usada na seção `runtimeConfig` de documentos do SSM com parâmetros do Parameter Store. Um parâmetro local não é o mesmo que um parâmetro do Systems Manager. Você pode distinguir parâmetros locais de parâmetros do Systems Manager pela ausência do prefixo `ssm:`.

```
"runtimeConfig":{
        "aws:runShellScript":{
            "properties":[
                {
                    "id":"0.aws:runShellScript",
                    "runCommand":"{{ commands }}",
                    "workingDirectory":"{{ workingDirectory }}",
                    "timeoutSeconds":"{{ executionTimeout }}"
```

Note

No momento, os documentos do SSM não são compatíveis com referências a parâmetros de `SecureString`. Isso significa que, para usar parâmetros de `SecureString` com, por exemplo, o Run Command, você precisa recuperar o valor do parâmetro antes de transmiti-lo para Run Command, como mostram os exemplos a seguir:

Linux & macOS

```
value=$(aws ssm get-parameters --names parameter-name --with-decryption)
```

```
aws ssm send-command \
    --name AWS-JoinDomain \
    --parameters password=$value \
    --instance-id instance-id
```

Windows

```
aws ssm send-command ^
    --name AWS-JoinDomain ^
    --parameters password=$value ^
    --instance-id instance-id
```

Powershell

```
$secure = (Get-SSMParameterValue -Names parameter-name -WithDecryption
 $True).Parameters[0].Value | ConvertTo-SecureString -AsPlainText -Force
```

```
$cred = New-Object System.Management.Automation.PSCredential -
argumentlist user-name,$secure
```

Como usar o suporte a parâmetros nativos no Parameter Store para IDs de imagem de máquina da Amazon

Ao criar um parâmetro `String`, você pode especificar um tipo de dado, como `aws:ec2:image`, para garantir que o valor do parâmetro inserido seja um formato válido de ID da Amazon Machine Image (AMI).

O suporte para formatos de ID de AMI permite que você evite atualizar todos os scripts e modelos com um novo ID sempre que a AMI que deseja usar em seus processos for alterada. Você pode criar um parâmetro com o tipo de dados `aws:ec2:image` e, em seu valor, inserir o ID de uma AMI. Esta é a AMI a partir da qual você deseja que novas instâncias sejam criadas. Depois, você faz referência a esse parâmetro em seus modelos, comandos e scripts.

Por exemplo, é possível especificar o parâmetro que contém o ID da AMI de sua preferência, quando você executar o comando `run-instances` do Amazon Elastic Compute Cloud (Amazon EC2).

> **Note**
>
> O usuário que executa esse comando deve ter permissões do AWS Identity and Access Management (IAM) que incluam a operação da API `ssm:GetParameters` para que o valor do parâmetro seja validado. Caso contrário, o processo de criação do parâmetro falhará.

Linux & macOS

```
aws ec2 run-instances \
    --image-id resolve:ssm:/golden-ami \
    --count 1 \
    --instance-type t2.micro \
    --key-name my-key-pair \
    --security-groups my-security-group
```

Windows

```
aws ec2 run-instances ^
    --image-id resolve:ssm:/golden-ami ^
    --count 1 ^
    --instance-type t2.micro ^
    --key-name my-key-pair ^
    --security-groups my-security-group
```

Você também pode escolher a AMI preferida ao criar uma instância usando o console do Amazon EC2. Para obter mais informações, consulte Usar um parâmetro do Systems Manager para localizar uma AMI no Guia do usuário do Amazon EC2.

Quando chegar a hora de usar uma AMI diferente em seu fluxo de trabalho de criação de instância, você só precisará atualizar o parâmetro com o novo valor da AMI, e o Parameter Store validará novamente que você inseriu um ID no formato adequado.

Conceder permissões para criar um parâmetro do tipo de dados **`aws:ec2:image`**

Usando políticas do AWS Identity and Access Management (IAM), você pode fornecer ou restringir o acesso do usuário a operações e conteúdos da API do Parameter Store.

Para criar um parâmetro de tipo de dados `aws:ec2:image`, o usuário deve ter ambas as permissões `ssm:PutParameter` e `ec2:DescribeImages`.

O exemplo de política a seguir concede permissão aos usuários para chamar o`PutParameter`Operação da API para`aws:ec2:image`. Isso significa que o usuário pode adicionar um parâmetro do tipo de dados`aws:ec2:image`Para o sistema.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ssm:PutParameter",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeImages",
            "Resource": "*"
        }
    ]
}
```

Como funciona a validação de formato da AMI

Quando você especifica `aws:ec2:image` como o tipo de dados para um parâmetro, o Systems Manager não cria o parâmetro imediatamente. Em vez disso, ele executa uma operação de validação assíncrona para garantir que o valor do parâmetro atenda aos requisitos de formatação para um ID de AMI e que a AMI especificada esteja disponível em sua Conta da AWS.

Um número de versão de parâmetro pode ser gerado antes que a operação de validação seja concluída. A operação pode não estar concluída mesmo que um número de versão do parâmetro seja gerado.

Para monitorar se seus parâmetros foram criados com êxito, recomendamos usar o Amazon EventBridge para enviar notificações sobre as operações dos parâmetros `create` e `update`. Essas notificações relatam se uma operação de parâmetro foi bem-sucedida ou não. Se uma operação falhar, a notificação incluirá uma mensagem de erro que indica o motivo da falha.

```
{
    "version": "0",
    "id": "eed4a719-0fa4-6a49-80d8-8ac65EXAMPLE",
    "detail-type": "Parameter Store Change",
    "source": "aws.ssm",
    "account": "111122223333",
    "time": "2020-05-26T22:04:42Z",
    "region": "us-east-2",
    "resources": [
        "arn:aws:ssm:us-east-2:111122223333:parameter/golden-ami"
    ],
    "detail": {
        "exception": "Unable to Describe Resource",
        "dataType": "aws:ec2:image",
        "name": "golden-ami",
        "type": "String",
        "operation": "Create"
    }
}
```

Para obter informações sobre como se inscrever em eventos do Parameter Store no EventBridge, consulte Configurar notificações ou acionar ações com base nos eventos do Parameter Store.

Excluir parâmetros do Parameter Store

Este tópico descreve como excluir parâmetros que você criou no Parameter Store, uma ferramenta do AWS Systems Manager.

> **Warning**
>
> A exclusão de um parâmetro remove todas as versões dele. Depois de excluído, o parâmetro e suas versões não poderão ser restaurados.

Para excluir um parâmetro usando o console

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Parameter Store.
3. Na guiaMy parameters (Meus parâmetros), marque a caixa de seleção ao lado de cada parâmetro a ser excluído.
4. Escolha Excluir.
5. Na caixa de diálogo de confirmação, selecione Delete parameters (Excluir parâmetros).

Para excluir um parâmetro usando a AWS CLI

- Execute o seguinte comando:

  ```
  aws ssm delete-parameter --name "my-parameter"
  ```

  Substitua *my-parameter* pelo nome do seu parâmetro a ser excluído.

  Para obter informações sobre outras opções que podem ser usadas com o comando `delete-parameter`, consulte delete-parameter na seção AWS Systems Manager da Referência de comandos da AWS CLI.

Trabalhar com parâmetros públicos no Parameter Store

Alguns Serviços da AWS publicam informações sobre artefatos comuns como parâmetros do AWS Systems Manager públicos. Por exemplo, o serviço Amazon Elastic Compute Cloud (Amazon EC2) publica informações sobre o Amazon Machine Images (AMIs) como parâmetros públicos.

Tópicos neste guia

- Descobrir parâmetros públicos no Parameter Store
- Chamar parâmetros públicos da AMI no Parameter Store
- Chamar o parâmetro público da AMI otimizada para ECS no Parameter Store
- Chamar o parâmetro público da AMI otimizada para EKS no Parameter Store
- Chamar parâmetros públicos para Serviços da AWS, regiões, endpoints, zonas de disponibilidade, zonas locais e zonas do Wavelength no Parameter Store

Publicações relacionadas em blogs da AWS

- Consultar regiões da Regiões da AWS, endpoints e muito mais usando o AWS Systems ManagerParameter Store
- Consultar os IDs de AMI do Amazon Linux mais recentes usando o AWS Systems Manager Parameter Store
- Consultar a AMI do Windows mais recente usando o Parameter Store do AWS Systems Manager

Descobrir parâmetros públicos no Parameter Store

É possível pesquisar parâmetros públicos usando o console do Parameter Store ou a AWS Command Line Interface.

Um nome de parâmetro público começa com `aws/service/list`. A próxima parte do nome corresponde ao serviço que possui esse parâmetro.

A lista a seguir é uma lista parcial de Serviços da AWS e outras ferramentas que fornecem parâmetros públicos:

- `ami-amazon-linux-latest`
- `ami-windows-latest`
- `ec2-macos`
- `appmesh`
- `aws-for-fluent-bit`
- `aws-sdk-pandas`
- `bottlerocket`
- `canonical`
- `cloud9`
- `datasync`
- `debian`
- `deeplearning`
- `ecs`
- `eks`

- `fis`
- `freebsd`
- `global-infrastructure`
- `marketplace`
- `neuron`
- `powertools`
- `sagemaker-distribution`
- `storagegateway`

Nem todos os parâmetros públicos são publicados em cada Região da AWS.

Localizar parâmetros públicos usando o console do Parameter Store

Você deve ter pelo menos um parâmetro no Conta da AWS e Região da AWS antes que possa pesquisar parâmetros públicos usando o console.

Como localizar parâmetros públicos usando o console

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Parameter Store.
3. Selecione a guia Public parameters (Parâmetros públicos).
4. Selecione a lista suspensa Select a service (Selecionar um serviço). Escolha o serviço cujos parâmetros você deseja usar.
5. (Opcional) Filtre os parâmetros pertencentes ao serviço selecionado inserindo mais informações na barra de pesquisa.
6. Escolha o parâmetro público que você quiser usar.

Localizar parâmetros públicos usando a AWS CLI

Use o `describe-parameters` para a descoberta de parâmetros públicos.

Usar o`get-parameters-by-path`para obter o caminho real para um serviço listado em`/aws/service/list`. Para obter o caminho do serviço, remova`/list`do caminho. Por exemplo, `/aws/service/list/ecs` torna-se `/aws/service/ecs`.

Para recuperar uma lista de parâmetros públicos de propriedade de diferentes serviços noParameter Store, execute o comando a seguir.

```
aws ssm get-parameters-by-path --path /aws/service/list
```

O comando retorna informações como as seguintes. Este exemplo foi truncado por questões de espaço.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/list/ami-al-latest",
            "Type": "String",
            "Value": "/aws/service/ami-al-latest/",
            "Version": 1,
            "LastModifiedDate": "2021-01-29T10:25:10.902000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/list/ami-al-latest",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/list/ami-windows-latest",
            "Type": "String",
            "Value": "/aws/service/ami-windows-latest/",
            "Version": 1,
            "LastModifiedDate": "2021-01-29T10:25:12.567000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/list/ami-windows-
latest",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/list/aws-storage-gateway-latest",
            "Type": "String",
            "Value": "/aws/service/aws-storage-gateway-latest/",
            "Version": 1,
            "LastModifiedDate": "2021-01-29T10:25:09.903000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/list/aws-storage-
gateway-latest",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/list/global-infrastructure",
            "Type": "String",
            "Value": "/aws/service/global-infrastructure/",
```

```
            "Version": 1,
            "LastModifiedDate": "2021-01-29T10:25:11.901000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/list/global-
infrastructure",
            "DataType": "text"
        }
    ]
}
```

Se você quiser exibir parâmetros de propriedade de um serviço específico, escolha o serviço na lista que foi produzido após a execução do comando anterior. Depois, crie uma chamada de `get-parameters-by-path` usando o nome do serviço desejado.

Por exemplo, `/aws/service/global-infrastructure`. O caminho pode ser de um nível (chama apenas parâmetros que correspondem aos valores exatos fornecidos) ou recursivo (contém elementos no caminho além do que você forneceu).

> **Note**
>
> O caminho `/aws/service/global-infrastructure` não é compatível com consultas em todas as regiões. Para ter mais informações, consulte Chamar parâmetros públicos para Serviços da AWS, regiões, endpoints, zonas de disponibilidade, zonas locais e zonas do Wavelength no Parameter Store.

Se nenhum resultado for retornado para o serviço especificado, adicione um sinalizador `--recursive` e execute o comando novamente.

```
aws ssm get-parameters-by-path --path /aws/service/global-infrastructure
```

Isso retorna todos os parâmetros de propriedade de`global-infrastructure`. Veja um exemplo a seguir.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/global-infrastructure/current-region",
            "Type": "String",
            "LastModifiedDate": "2019-06-21T05:15:34.252000-07:00",
            "Version": 1,
            "Tier": "Standard",
```

```
            "Policies": [],
            "DataType": "text"
        },
        {
            "Name": "/aws/service/global-infrastructure/version",
            "Type": "String",
            "LastModifiedDate": "2019-02-04T06:59:32.875000-08:00",
            "Version": 1,
            "Tier": "Standard",
            "Policies": [],
            "DataType": "text"
        }
    ]
}
```

Você também pode exibir parâmetros de propriedade de um serviço específico usando o`Option:BeginsWith`Filter.

```
aws ssm describe-parameters --parameter-filters "Key=Name, Option=BeginsWith, Values=/
aws/service/ami-amazon-linux-latest"
```

O comando retorna informações como as seguintes. Este exemplo de saída foi truncado por questões de espaço.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-ebs",
            "Type": "String",
            "LastModifiedDate": "2021-01-26T13:39:40.686000-08:00",
            "Version": 25,
            "Tier": "Standard",
            "Policies": [],
            "DataType": "text"
        },
        {
            "Name": "/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-gp2",
            "Type": "String",
            "LastModifiedDate": "2021-01-26T13:39:40.807000-08:00",
            "Version": 25,
            "Tier": "Standard",
            "Policies": [],
            "DataType": "text"
```

```
        },
        {
            "Name": "/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-s3",
            "Type": "String",
            "LastModifiedDate": "2021-01-26T13:39:40.920000-08:00",
            "Version": 25,
            "Tier": "Standard",
            "Policies": [],
            "DataType": "text"
        }
    ]
}
```

> **Note**
>
> Os parâmetros retornados podem ser diferentes quando você usa`Option=BeginsWith`porque ele usa um padrão de pesquisa diferente.

Chamar parâmetros públicos da AMI no Parameter Store

Os parâmetros públicos da Amazon Machine Image (AMI) do Amazon Elastic Compute Cloud (Amazon EC2) estão disponíveis para o Amazon Linux 1, Amazon Linux 2, Amazon Linux 2023 (AL2023), macOS e Windows Server nos seguintes caminhos:

- Amazon Linux 1, Amazon Linux 2 e Amazon Linux 2023: `/aws/service/ami-amazon-linux-latest`
- macOS: `/aws/service/ec2-macos`
- Windows Server: `/aws/service/ami-windows-latest`

Chamar parâmetros públicos da AMI para o Amazon Linux1, Amazon Linux 2 e Amazon Linux 2023

Você pode visualizar uma lista de todas as AMIs do Amazon Linux 1, Amazon Linux 2 e Amazon Linux 2023 (AL2023) na atual Região da AWS usando o comando a seguir na AWS Command Line Interface (AWS CLI).

Linux & macOS

```
aws ssm get-parameters-by-path \
```

```
    --path /aws/service/ami-amazon-linux-latest \
    --query 'Parameters[].Name'
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/ami-amazon-linux-latest ^
    --query Parameters[].Name
```

O comando retorna informações como as seguintes.

```
[
    "/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-arm64",
    "/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-x86_64",
    "/aws/service/ami-amazon-linux-latest/al2023-ami-minimal-kernel-6.1-arm64",
    "/aws/service/ami-amazon-linux-latest/al2023-ami-minimal-kernel-6.1-x86_64",
    "/aws/service/ami-amazon-linux-latest/al2023-ami-minimal-kernel-default-arm64",
    "/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-gp2",
    "/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-s3",
    "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-ebs",
    "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2",
    "/aws/service/ami-amazon-linux-latest/amzn2-ami-kernel-5.10-hvm-x86_64-ebs",
    "/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-default-arm64",
    "/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-default-x86_64",
    "/aws/service/ami-amazon-linux-latest/al2023-ami-minimal-kernel-default-x86_64",
    "/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-ebs",
    "/aws/service/ami-amazon-linux-latest/amzn-ami-minimal-hvm-x86_64-ebs",
    "/aws/service/ami-amazon-linux-latest/amzn-ami-minimal-hvm-x86_64-s3",
    "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-arm64-gp2",
    "/aws/service/ami-amazon-linux-latest/amzn2-ami-kernel-5.10-hvm-arm64-gp2",
    "/aws/service/ami-amazon-linux-latest/amzn2-ami-kernel-5.10-hvm-x86_64-gp2",
    "/aws/service/ami-amazon-linux-latest/amzn2-ami-minimal-hvm-arm64-ebs",
    "/aws/service/ami-amazon-linux-latest/amzn2-ami-minimal-hvm-x86_64-ebs"
]
```

Você pode visualizar detalhes sobre essas AMIs, incluindo os IDs da AMI e ARNs (nomes de recursos da Amazon), usando o comando a seguir.

Linux & macOS

```
aws ssm get-parameters-by-path \
```

```
    --path "/aws/service/ami-amazon-linux-latest" \
    --region region
```

Windows

```
aws ssm get-parameters-by-path ^
    --path "/aws/service/ami-amazon-linux-latest" ^
    --region region
```

A *região* representa o identificador da região para uma região da Região da AWS compatível com o AWS Systems Manager, como `us-east-2` para a região Leste dos EUA (Ohio). Para ver uma lista dos valores de *região* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

O comando retorna informações como as seguintes. Este exemplo de saída foi truncado por questões de espaço.

```
{
    "Parameters": [
         {
            "Name": "/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-arm64",
            "Type": "String",
            "Value": "ami-0b1b8b24a6c8e5d8b",
            "Version": 69,
            "LastModifiedDate": "2024-03-13T14:05:09.583000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ami-amazon-linux-
latest/al2023-ami-kernel-6.1-arm64",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-
x86_64",
            "Type": "String",
            "Value": "ami-0e0bf53f6def86294",
            "Version": 69,
            "LastModifiedDate": "2024-03-13T14:05:09.890000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ami-amazon-linux-
latest/al2023-ami-kernel-6.1-x86_64",
            "DataType": "text"
        },
        {
```

```
            "Name": "/aws/service/ami-amazon-linux-latest/al2023-ami-minimal-
kernel-6.1-arm64",
            "Type": "String",
            "Value": "ami-09951bb66f9e5b5a5",
            "Version": 69,
            "LastModifiedDate": "2024-03-13T14:05:10.197000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ami-amazon-linux-
latest/al2023-ami-minimal-kernel-6.1-arm64",
            "DataType": "text"
        }
    ]
}
```

Você pode visualizar detalhes de uma AMI específica usando a ação da API GetParameters com o nome completo da AMI, incluindo o caminho. Veja a seguir um exemplo de comando.

Linux & macOS

```
aws ssm get-parameters \
    --names /aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-arm64 \
    --region us-east-2
```

Windows

```
aws ssm get-parameters ^
    --names /aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-arm64 ^
    --region us-east-2
```

O comando retorna as seguintes informações.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-arm64",
            "Type": "String",
            "Value": "ami-0b1b8b24a6c8e5d8b",
            "Version": 69,
            "LastModifiedDate": "2024-03-13T14:05:09.583000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ami-amazon-linux-
latest/al2023-ami-kernel-6.1-arm64",
            "DataType": "text"
```

```
        }
    ],
    "InvalidParameters": []
}
```

Chamar parâmetros públicos da AMI para macOS

Você pode visualizar uma lista de todas as macOS da AMIs na Região da AWS atual,usando o seguinte comando na AWS CLI.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/ec2-macos\
    --query 'Parameters[].Name'
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/ec2-macos ^
    --query Parameters[].Name
```

O comando retorna informações como as seguintes.

```
[
    "/aws/service/ec2-macos/sonoma/x86_64_mac/latest/image_id",
    "/aws/service/ec2-macos/ventura/x86_64_mac/latest/image_id",
    "/aws/service/ec2-macos/monterey/x86_64_mac/latest/image_id",
    "/aws/service/ec2-macos/sonoma/arm64_mac/latest/image_id",
    "/aws/service/ec2-macos/ventura/arm64_mac/latest/image_id",
    "/aws/service/ec2-macos/monterey/arm64_mac/latest/image_id"
]
```

Você pode visualizar detalhes sobre essas AMIs, incluindo os IDs da AMI e ARNs (nomes de recursos da Amazon), usando o comando a seguir.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path "/aws/service/ec2-macos" \
```

```
    --region region
```

Windows

```
aws ssm get-parameters-by-path ^
    --path "/aws/service/ec2-macos" ^
    --region region
```

A *região* representa o identificador da região para uma região da Região da AWS compatível com o AWS Systems Manager, como `us-east-2` para a região Leste dos EUA (Ohio). Para ver uma lista dos valores de *região* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

O comando retorna informações como as seguintes. Este exemplo de saída foi truncado por questões de espaço.

```
{
    "Parameters": [
        ...sample results pending...
    ]
}
```

Você pode visualizar detalhes de uma AMI específica usando a ação da API GetParameters com o nome completo da AMI, incluindo o caminho. Veja a seguir um exemplo de comando.

Linux & macOS

```
aws ssm get-parameters \
    --names /aws/service/ec2-macos/...pending... \
    --region us-east-2
```

Windows

```
aws ssm get-parameters ^
     --names /aws/service/ec2-macos/...pending... ^
    --region us-east-2
```

O comando retorna as seguintes informações.

```
{
    "Parameters": [
         ...sample results pending...
    ],
    "InvalidParameters": []
}
```

Chamar parâmetros públicos da AMI para Windows Server

Você pode visualizar uma lista de todas as Windows Server da AMIs na Região da AWS atual,usando o seguinte comando na AWS CLI.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/ami-windows-latest \
    --query 'Parameters[].Name'
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/ami-windows-latest ^
    --query Parameters[].Name
```

O comando retorna informações como as seguintes. Este exemplo de saída foi truncado por questões de espaço.

```
[
    "/aws/service/ami-windows-latest/EC2LaunchV2-Windows_Server-2016-English-Full-
Base",
    "/aws/service/ami-windows-latest/Windows_Server-2016-English-Full-
SQL_2014_SP3_Enterprise",
    "/aws/service/ami-windows-latest/Windows_Server-2016-German-Full-Base",
    "/aws/service/ami-windows-latest/Windows_Server-2016-Japanese-Full-
SQL_2016_SP3_Standard",
    "/aws/service/ami-windows-latest/Windows_Server-2016-Japanese-Full-SQL_2017_Web",
    "/aws/service/ami-windows-latest/Windows_Server-2019-English-Core-
EKS_Optimized-1.25",
    "/aws/service/ami-windows-latest/Windows_Server-2019-Italian-Full-Base",
    "/aws/service/ami-windows-latest/Windows_Server-2022-Japanese-Full-
SQL_2019_Enterprise",
```

```
    "/aws/service/ami-windows-latest/Windows_Server-2022-Portuguese_Brazil-Full-Base",
    "/aws/service/ami-windows-latest/amzn2-ami-hvm-2.0.20191217.0-x86_64-gp2-mono",
    "/aws/service/ami-windows-latest/Windows_Server-2016-English-Deep-Learning",
    "/aws/service/ami-windows-latest/Windows_Server-2016-Japanese-Full-
SQL_2016_SP3_Web",
    "/aws/service/ami-windows-latest/Windows_Server-2016-Korean-Full-Base",
    "/aws/service/ami-windows-latest/Windows_Server-2019-English-STIG-Core",
    "/aws/service/ami-windows-latest/Windows_Server-2019-French-Full-Base",
    "/aws/service/ami-windows-latest/Windows_Server-2019-Japanese-Full-
SQL_2017_Enterprise",
    "/aws/service/ami-windows-latest/Windows_Server-2019-Korean-Full-Base",
    "/aws/service/ami-windows-latest/Windows_Server-2022-English-Full-SQL_2022_Web",
    "/aws/service/ami-windows-latest/Windows_Server-2022-Italian-Full-Base",
    "/aws/service/ami-windows-latest/amzn2-x86_64-SQL_2019_Express",
    "/aws/service/ami-windows-latest/EC2LaunchV2-Windows_Server-2016-English-Core-
Base",
    "/aws/service/ami-windows-latest/Windows_Server-2016-English-Full-
SQL_2019_Enterprise",
    "/aws/service/ami-windows-latest/Windows_Server-2016-English-Full-
SQL_2019_Standard",
    "/aws/service/ami-windows-latest/Windows_Server-2016-Portuguese_Portugal-Full-
Base",
    "/aws/service/ami-windows-latest/Windows_Server-2019-English-Core-
EKS_Optimized-1.24",
    "/aws/service/ami-windows-latest/Windows_Server-2019-English-Deep-Learning",
    "/aws/service/ami-windows-latest/Windows_Server-2019-English-Full-SQL_2017_Web",
    "/aws/service/ami-windows-latest/Windows_Server-2019-Hungarian-Full-Base
]
```

Você pode visualizar detalhes sobre essas AMIs, incluindo os IDs da AMI e ARNs (nomes de recursos da Amazon), usando o comando a seguir.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path "/aws/service/ami-windows-latest" \
    --region region
```

Windows

```
aws ssm get-parameters-by-path ^
    --path "/aws/service/ami-windows-latest" ^
```

```
    --region region
```

A *região* representa o identificador da região para uma região da Região da AWS compatível com o AWS Systems Manager, como `us-east-2` para a região Leste dos EUA (Ohio). Para ver uma lista dos valores de *região* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

O comando retorna informações como as seguintes. Este exemplo de saída foi truncado por questões de espaço.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/ami-windows-latest/EC2LaunchV2-Windows_Server-2016-
English-Full-Base",
            "Type": "String",
            "Value": "ami-0a30b2e65863e2d16",
            "Version": 36,
            "LastModifiedDate": "2024-03-15T15:58:37.976000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ami-windows-latest/
EC2LaunchV2-Windows_Server-2016-English-Full-Base",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/ami-windows-latest/Windows_Server-2016-English-Full-
SQL_2014_SP3_Enterprise",
            "Type": "String",
            "Value": "ami-001f20c053dd120ce",
            "Version": 69,
            "LastModifiedDate": "2024-03-15T15:53:58.905000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ami-windows-latest/
Windows_Server-2016-English-Full-SQL_2014_SP3_Enterprise",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/ami-windows-latest/Windows_Server-2016-German-Full-
Base",
            "Type": "String",
            "Value": "ami-063be4935453e94e9",
            "Version": 102,
            "LastModifiedDate": "2024-03-15T15:51:12.003000-04:00",
```

```
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ami-windows-latest/
Windows_Server-2016-German-Full-Base",
            "DataType": "text"
        }
    ]
}
```

Você pode visualizar detalhes de uma AMI específica usando a ação da API GetParameters com o nome completo da AMI, incluindo o caminho. Veja a seguir um exemplo de comando.

Linux & macOS

```
aws ssm get-parameters \
    --names /aws/service/ami-windows-latest/EC2LaunchV2-Windows_Server-2016-English-
Full-Base \
    --region us-east-2
```

Windows

```
aws ssm get-parameters ^
    --names /aws/service/ami-windows-latest/EC2LaunchV2-Windows_Server-2016-English-
Full-Base ^
    --region us-east-2
```

O comando retorna as seguintes informações.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/ami-windows-latest/EC2LaunchV2-Windows_Server-2016-
English-Full-Base",
            "Type": "String",
            "Value": "ami-0a30b2e65863e2d16",
            "Version": 36,
            "LastModifiedDate": "2024-03-15T15:58:37.976000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ami-windows-latest/
EC2LaunchV2-Windows_Server-2016-English-Full-Base",
            "DataType": "text"
        }
    ],
    "InvalidParameters": []
```

```
}
```

Chamar o parâmetro público da AMI otimizada para ECS no Parameter Store

O serviço Amazon Elastic Container Service (Amazon ECS) publica o nome da versão mais recente da Amazon Machine Images (AMIs) otimizada para Amazon ECS, como parâmetros públicos. Os usuários são encorajados a usar esta AMI ao criar um novo cluster do Amazon Elastic Compute Cloud (Amazon EC2) para o Amazon ECS porque a AMIs otimizada inclui correções de bugs e atualizações de recursos.

Use o comando a seguir para visualizar o nome da AMI otimizada para Amazon ECS mais recente do Amazon Linux 2. Para ver comandos de outros sistemas operacionais, consulte Recuperar metadados de AMIs otimizadas para o Amazon ECS no Guia do desenvolvedor do Amazon Elastic Container Service.

Linux & macOS

```
aws ssm get-parameters \
    --names /aws/service/ecs/optimized-ami/amazon-linux-2/recommended
```

Windows

```
aws ssm get-parameters ^
    --names /aws/service/ecs/optimized-ami/amazon-linux-2/recommended
```

O comando retorna informações como as seguintes.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/recommended",
            "Type": "String",
            "Value": "{\"schema_version\":1,\"image_name\":\"amzn2-ami-ecs-
hvm-2.0.20210929-x86_64-ebs\",\"image_id\":\"ami-0c38a2329ed4dae9a\",\"os\":\"Amazon
 Linux 2\",\"ecs_runtime_version\":\"Docker version 20.10.7\",\"ecs_agent_version\":
\"1.55.4\"}",
            "Version": 73,
            "LastModifiedDate": "2021-10-06T16:35:10.004000-07:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/recommended",
            "DataType": "text"
```

```
        }
    ],
    "InvalidParameters": []
}
```

Chamar o parâmetro público da AMI otimizada para EKS no Parameter Store

O serviço Amazon Elastic Kubernetes Service (Amazon EKS) publica o nome da nova Amazon Machine Image (AMI) otimizada para Amazon EKS como um parâmetro público. Recomendamos o uso dessa AMI ao adicionar nós a um cluster do Amazon EKS, já que as novas versões incluem patches do Kubernetes e atualizações de segurança. Antes, para garantir o uso da AMI mais recente, era necessário verificar a documentação do Amazon EKS e atualizar manualmente os modelos ou recursos de implantação com o novo ID da AMI.

Use o comando a seguir para visualizar o nome da AMI otimizada para Amazon EKS mais recente do Amazon Linux 2.

Linux & macOS

```
aws ssm get-parameters \
    --names /aws/service/eks/optimized-ami/1.14/amazon-linux-2/recommended
```

Windows

```
aws ssm get-parameters ^
    --names /aws/service/eks/optimized-ami/1.14/amazon-linux-2/recommended
```

O comando retorna informações como as seguintes.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/eks/optimized-ami/1.14/amazon-linux-2/recommended",
            "Type": "String",
            "Value": "{\"schema_version\":\"2\",\"image_id\":\"ami-08984d8491de17ca0\",
\"image_name\":\"amazon-eks-node-1.14-v20201007\",\"release_version\":
\"1.14.9-20201007\"}",
            "Version": 24,
            "LastModifiedDate": "2020-11-17T10:16:09.971000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/eks/optimized-
ami/1.14/amazon-linux-2/recommended",
```

```
            "DataType": "text"
        }
    ],
    "InvalidParameters": []
}
```

Chamar parâmetros públicos para Serviços da AWS, regiões, endpoints, zonas de disponibilidade, zonas locais e zonas do Wavelength no Parameter Store

Você pode chamar a Região da AWS, o serviço, o endpoint, a disponibilidade e as zonas do Wavelength de parâmetros públicos usando o caminho a seguir.

`/aws/service/global-infrastructure`

> **Note**
>
> Atualmente, o caminho `/aws/service/global-infrastructure` é compatível com consultas apenas nas seguintes Regiões da AWS:
>
> - Leste dos EUA (Norte da Virgínia) (us-east-1)
> - Leste dos EUA (Ohio) (us-east-2)
> - Oeste dos EUA (Norte da Califórnia) (us-west-1)
> - Oeste dos EUA (Oregon) (us-west-2)
> - Ásia-Pacífico (Hong Kong) (ap-east-1)
> - Ásia-Pacífico (Mumbai) (ap-south-1)
> - Ásia-Pacífico (Seul) (ap-northeast-2)
> - Ásia-Pacífico (Singapura) (ap-southeast-1)
> - Ásia-Pacífico (Sydney) (ap-southeast-2)
> - Ásia Pacific (Tóquio) (ap-northeast-1)
> - Canadá (Central) (ca-central-1)
> - Europa (Frankfurt) (eu-central-1)
> - Europa (Irlanda) (eu-west-1)
> - Europa (Londres) (eu-west-2)
> - Europa (Paris) (eu-west-3)
> - UE (Estocolmo) (eu-north-1)
> - América do Sul (São Paulo) (sa-east-1)

Se você estiver trabalhando em uma região comercial diferente, poderá especificar uma região compatível em sua consulta para ver os resultados. Por exemplo, se você estiver trabalhando na região Oeste do Canadá (Calgary) (ca-west-1), você pode especificar Canadá (Central) (ca-central-1) em sua consulta:

```
aws ssm get-parameters-by-path \
    --path /aws/service/global-infrastructure/regions \
    --region ca-central-1
```

Visualizar as Regiões da AWS ativas

É possível visualizar uma lista de todas as Regiões da AWS ativas usando o seguinte comando na AWS Command Line Interface (AWS CLI).

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/global-infrastructure/regions \
    --query 'Parameters[].Name'
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/global-infrastructure/regions ^
    --query Parameters[].Name
```

O comando retorna informações como as seguintes.

```
[
    "/aws/service/global-infrastructure/regions/af-south-1",
    "/aws/service/global-infrastructure/regions/ap-east-1",
    "/aws/service/global-infrastructure/regions/ap-northeast-3",
    "/aws/service/global-infrastructure/regions/ap-south-2",
    "/aws/service/global-infrastructure/regions/ca-central-1",
    "/aws/service/global-infrastructure/regions/eu-central-2",
    "/aws/service/global-infrastructure/regions/eu-west-2",
    "/aws/service/global-infrastructure/regions/eu-west-3",
```

```
    "/aws/service/global-infrastructure/regions/us-east-1",
    "/aws/service/global-infrastructure/regions/us-gov-west-1",
    "/aws/service/global-infrastructure/regions/ap-northeast-2",
    "/aws/service/global-infrastructure/regions/ap-southeast-1",
    "/aws/service/global-infrastructure/regions/ap-southeast-2",
    "/aws/service/global-infrastructure/regions/ap-southeast-3",
    "/aws/service/global-infrastructure/regions/cn-north-1",
    "/aws/service/global-infrastructure/regions/cn-northwest-1",
    "/aws/service/global-infrastructure/regions/eu-south-1",
    "/aws/service/global-infrastructure/regions/eu-south-2",
    "/aws/service/global-infrastructure/regions/us-east-2",
    "/aws/service/global-infrastructure/regions/us-west-1",
    "/aws/service/global-infrastructure/regions/ap-northeast-1",
    "/aws/service/global-infrastructure/regions/ap-south-1",
    "/aws/service/global-infrastructure/regions/ap-southeast-4",
    "/aws/service/global-infrastructure/regions/ca-west-1",
    "/aws/service/global-infrastructure/regions/eu-central-1",
    "/aws/service/global-infrastructure/regions/il-central-1",
    "/aws/service/global-infrastructure/regions/me-central-1",
    "/aws/service/global-infrastructure/regions/me-south-1",
    "/aws/service/global-infrastructure/regions/sa-east-1",
    "/aws/service/global-infrastructure/regions/us-gov-east-1",
    "/aws/service/global-infrastructure/regions/eu-north-1",
    "/aws/service/global-infrastructure/regions/eu-west-1",
    "/aws/service/global-infrastructure/regions/us-west-2"
]
```

Visualizar Serviços da AWS disponíveis

É possível visualizar uma lista completa de todos os Serviços da AWS disponíveis e classificá-los em ordem alfabética usando o comando a seguir. Este exemplo de saída foi truncado por questões de espaço.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/global-infrastructure/services \
    --query 'Parameters[].Name | sort(@)'
```

Windows

```
aws ssm get-parameters-by-path ^
```

```
    --path /aws/service/global-infrastructure/services ^
    --query "Parameters[].Name | sort(@)"
```

O comando retorna informações como as seguintes. Este exemplo foi truncado por questões de espaço.

```
[
     "/aws/service/global-infrastructure/services/accessanalyzer",
    "/aws/service/global-infrastructure/services/account",
    "/aws/service/global-infrastructure/services/acm",
    "/aws/service/global-infrastructure/services/acm-pca",
    "/aws/service/global-infrastructure/services/ahl",
    "/aws/service/global-infrastructure/services/aiq",
    "/aws/service/global-infrastructure/services/amazonlocationservice",
    "/aws/service/global-infrastructure/services/amplify",
    "/aws/service/global-infrastructure/services/amplifybackend",
    "/aws/service/global-infrastructure/services/apigateway",
    "/aws/service/global-infrastructure/services/apigatewaymanagementapi",
    "/aws/service/global-infrastructure/services/apigatewayv2",
    "/aws/service/global-infrastructure/services/appconfig",
    "/aws/service/global-infrastructure/services/appconfigdata",
    "/aws/service/global-infrastructure/services/appflow",
    "/aws/service/global-infrastructure/services/appintegrations",
    "/aws/service/global-infrastructure/services/application-autoscaling",
    "/aws/service/global-infrastructure/services/application-insights",
    "/aws/service/global-infrastructure/services/applicationcostprofiler",
    "/aws/service/global-infrastructure/services/appmesh",
    "/aws/service/global-infrastructure/services/apprunner",
    "/aws/service/global-infrastructure/services/appstream",
    "/aws/service/global-infrastructure/services/appsync",
    "/aws/service/global-infrastructure/services/aps",
    "/aws/service/global-infrastructure/services/arc-zonal-shift",
    "/aws/service/global-infrastructure/services/artifact",
    "/aws/service/global-infrastructure/services/athena",
    "/aws/service/global-infrastructure/services/auditmanager",
    "/aws/service/global-infrastructure/services/augmentedairuntime",
    "/aws/service/global-infrastructure/services/aurora",
    "/aws/service/global-infrastructure/services/autoscaling",
    "/aws/service/global-infrastructure/services/aws-appfabric",
    "/aws/service/global-infrastructure/services/awshealthdashboard",
```

Visualizar regiões compatíveis para um AWS service (Serviço da AWS)

É possível visualizar uma lista de Regiões da AWS em que um serviço está disponível. Este exemplo usa o AWS Systems Manager (`ssm`).

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/global-infrastructure/services/ssm/regions \
    --query 'Parameters[].Value'
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/global-infrastructure/services/ssm/regions ^
    --query Parameters[].Value
```

O comando retorna informações como as seguintes.

```
[
    "ap-south-1",
    "eu-central-1",
    "eu-central-2",
    "eu-west-1",
    "eu-west-2",
    "eu-west-3",
    "il-central-1",
    "me-south-1",
    "us-east-2",
    "us-gov-west-1",
    "af-south-1",
    "ap-northeast-3",
    "ap-southeast-1",
    "ap-southeast-4",
    "ca-central-1",
    "ca-west-1",
    "cn-north-1",
    "eu-north-1",
    "eu-south-2",
    "us-west-1",
    "ap-east-1",
    "ap-northeast-1",
    "ap-northeast-2",
    "ap-southeast-2",
```

```
    "ap-southeast-3",
    "cn-northwest-1",
    "eu-south-1",
    "me-central-1",
    "us-gov-east-1",
    "us-west-2",
    "ap-south-2",
    "sa-east-1",
    "us-east-1"
]
```

Visualizar o endpoint regional de um serviço

É possível visualizar um endpoint regional para um serviço usando o comando a seguir. Esse comando consulta a região Leste dos EUA (Ohio) (us-east-2).

Linux & macOS

```
aws ssm get-parameter \
    --name /aws/service/global-infrastructure/regions/us-east-2/services/ssm/
endpoint \
    --query 'Parameter.Value'
```

Windows

```
aws ssm get-parameter ^
    --name /aws/service/global-infrastructure/regions/us-east-2/services/ssm/
endpoint ^
    --query Parameter.Value
```

O comando retorna informações como as seguintes.

```
"ssm.us-east-2.amazonaws.com"
```

Visualizar detalhes completos da zona de disponibilidade

Você pode visualizar zonas de disponibilidade usando o seguinte comando:

Linux & macOS

```
aws ssm get-parameters-by-path \
```

```
    --path /aws/service/global-infrastructure/availability-zones/
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/global-infrastructure/availability-zones/
```

O comando retorna informações como as seguintes. Este exemplo foi truncado por questões de espaço.


```
{
    "Parameters": [
        {
            "Name": "/aws/service/global-infrastructure/availability-zones/afs1-az3",
            "Type": "String",
            "Value": "afs1-az3",
            "Version": 1,
            "LastModifiedDate": "2020-04-21T12:05:35.375000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
availability-zones/afs1-az3",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/global-infrastructure/availability-zones/aps1-az2",
            "Type": "String",
            "Value": "aps1-az2",
            "Version": 1,
            "LastModifiedDate": "2020-04-03T16:13:57.351000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
availability-zones/aps1-az2",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/global-infrastructure/availability-zones/apse3-az1",
            "Type": "String",
            "Value": "apse3-az1",
            "Version": 1,
            "LastModifiedDate": "2021-12-13T08:51:38.983000-05:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
availability-zones/apse3-az1",
            "DataType": "text"
        }
```

```
    ]
}
```

Visualizar somente nomes da zona de disponibilidade

Você só pode visualizar os nomes das zonas de disponibilidade usando o comando a seguir.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/global-infrastructure/availability-zones \
    --query 'Parameters[].Name | sort(@)'
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/global-infrastructure/availability-zones ^
    --query "Parameters[].Name | sort(@)"
```

O comando retorna informações como as seguintes. Este exemplo foi truncado por questões de espaço.

```
[
    "/aws/service/global-infrastructure/availability-zones/afs1-az1",
    "/aws/service/global-infrastructure/availability-zones/afs1-az2",
    "/aws/service/global-infrastructure/availability-zones/afs1-az3",
    "/aws/service/global-infrastructure/availability-zones/ape1-az1",
    "/aws/service/global-infrastructure/availability-zones/ape1-az2",
    "/aws/service/global-infrastructure/availability-zones/ape1-az3",
    "/aws/service/global-infrastructure/availability-zones/apne1-az1",
    "/aws/service/global-infrastructure/availability-zones/apne1-az2",
    "/aws/service/global-infrastructure/availability-zones/apne1-az3",
    "/aws/service/global-infrastructure/availability-zones/apne1-az4"
```

Visualizar nomes de zonas de disponibilidade em uma única região

Você pode visualizar os nomes das zonas de disponibilidade em uma região (`us-east-2`, neste exemplo) usando o comando a seguir.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/global-infrastructure/regions/us-east-2/availability-zones \
    --query 'Parameters[].Name | sort(@)'
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/global-infrastructure/regions/us-east-2/availability-zones ^
    --query "Parameters[].Name | sort(@)"
```

O comando retorna informações como as seguintes.

```
[
    "/aws/service/global-infrastructure/regions/us-east-2/availability-zones/use2-az1",
    "/aws/service/global-infrastructure/regions/us-east-2/availability-zones/use2-az2",
    "/aws/service/global-infrastructure/regions/us-east-2/availability-zones/use2-az3"
```

Visualizar somente ARNs da zona de disponibilidade

Você só pode visualizar os nomes do recurso da Amazon (ARNs) das zonas de disponibilidade, usando o comando a seguir.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/global-infrastructure/availability-zones \
    --query 'Parameters[].ARN | sort(@)'
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/global-infrastructure/availability-zones ^
    --query "Parameters[].ARN | sort(@)"
```

O comando retorna informações como as seguintes. Este exemplo foi truncado por questões de espaço.

```
[
    "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/availability-
zones/afs1-az1",
    "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/availability-
zones/afs1-az2",
    "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/availability-
zones/afs1-az3",
    "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/availability-
zones/ape1-az1",
    "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/availability-
zones/ape1-az2",
    "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/availability-
zones/ape1-az3",
    "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/availability-
zones/apne1-az1",
```

Visualizar detalhes da zona local

Você pode visualizar zonas locais usando o seguinte comando.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/global-infrastructure/local-zones
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/global-infrastructure/local-zones
```

O comando retorna informações como as seguintes. Este exemplo foi truncado por questões de espaço.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/global-infrastructure/local-zones/afs1-los1-az1",
            "Type": "String",
            "Value": "afs1-los1-az1",
            "Version": 1,
```

```
            "LastModifiedDate": "2023-01-25T11:53:11.690000-05:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
local-zones/afs1-los1-az1",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/global-infrastructure/local-zones/apne1-tpe1-az1",
            "Type": "String",
            "Value": "apne1-tpe1-az1",
            "Version": 1,
            "LastModifiedDate": "2024-03-15T12:35:41.076000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
local-zones/apne1-tpe1-az1",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/global-infrastructure/local-zones/aps1-ccu1-az1",
            "Type": "String",
            "Value": "aps1-ccu1-az1",
            "Version": 1,
            "LastModifiedDate": "2022-12-19T11:34:43.351000-05:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
local-zones/aps1-ccu1-az1",
            "DataType": "text"
        }
    ]
}
```

Visualizar os detalhes da zona de Wavelength

Você pode visualizar zonas do Wavelength usando o seguinte comando:

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/global-infrastructure/wavelength-zones
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/global-infrastructure/wavelength-zones
```

O comando retorna informações como as seguintes. Este exemplo foi truncado por questões de espaço.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/global-infrastructure/wavelength-zones/apne1-wl1-nrt-
wlz1",
            "Type": "String",
            "Value": "apne1-wl1-nrt-wlz1",
            "Version": 3,
            "LastModifiedDate": "2020-12-15T17:16:04.715000-05:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
wavelength-zones/apne1-wl1-nrt-wlz1",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/global-infrastructure/wavelength-zones/apne2-wl1-sel-
wlz1",
            "Type": "String",
            "Value": "apne2-wl1-sel-wlz1",
            "Version": 1,
            "LastModifiedDate": "2022-05-25T12:29:13.862000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
wavelength-zones/apne2-wl1-sel-wlz1",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/global-infrastructure/wavelength-zones/cac1-wl1-yto-
wlz1",
            "Type": "String",
            "Value": "cac1-wl1-yto-wlz1",
            "Version": 1,
            "LastModifiedDate": "2022-04-26T09:57:44.495000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
wavelength-zones/cac1-wl1-yto-wlz1",
            "DataType": "text"
        }
    ]
}
```

Visualizar todos os parâmetros e valores em uma região local

Você pode visualizar todos os dados de parâmetros de uma região local usando o seguinte comando.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path "/aws/service/global-infrastructure/local-zones/usw2-lax1-az1/"
```

Windows

```
aws ssm get-parameters-by-path ^
    --path "/aws/service/global-infrastructure/local-zones/use1-bos1-az1"
```

O comando retorna informações como as seguintes. Este exemplo foi truncado por questões de espaço.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/
geolocationCountry",
            "Type": "String",
            "Value": "US",
            "Version": 3,
            "LastModifiedDate": "2020-12-15T14:16:17.641000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
local-zones/use1-bos1-az1/geolocationCountry",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/
geolocationRegion",
            "Type": "String",
            "Value": "US-MA",
            "Version": 3,
            "LastModifiedDate": "2020-12-15T14:16:17.794000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
local-zones/use1-bos1-az1/geolocationRegion",
            "DataType": "text"
        },
        {
```

```
            "Name": "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/
location",
            "Type": "String",
            "Value": "US East (Boston)",
            "Version": 1,
            "LastModifiedDate": "2021-01-11T10:53:24.634000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
local-zones/use1-bos1-az1/location",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/
network-border-group",
            "Type": "String",
            "Value": "us-east-1-bos-1",
            "Version": 3,
            "LastModifiedDate": "2020-12-15T14:16:20.641000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
local-zones/use1-bos1-az1/network-border-group",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/
parent-availability-zone",
            "Type": "String",
            "Value": "use1-az4",
            "Version": 3,
            "LastModifiedDate": "2020-12-15T14:16:20.834000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
local-zones/use1-bos1-az1/parent-availability-zone",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/
parent-region",
            "Type": "String",
            "Value": "us-east-1",
            "Version": 3,
            "LastModifiedDate": "2020-12-15T14:16:20.721000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
local-zones/use1-bos1-az1/parent-region",
            "DataType": "text"
        },
        {
```

```
            "Name": "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/zone-
group",
            "Type": "String",
            "Value": "us-east-1-bos-1",
            "Version": 3,
            "LastModifiedDate": "2020-12-15T14:16:17.983000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
local-zones/use1-bos1-az1/zone-group",
            "DataType": "text"
        }
    ]
}
```

Visualizar somente nomes de parâmetros de zona local

Você pode visualizar somente os nomes dos parâmetros de região local usando o seguinte comando.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/global-infrastructure/local-zones/usw2-lax1-az1 \
    --query 'Parameters[].Name | sort(@)'
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/global-infrastructure/local-zones/use1-bos1-az1 ^
    --query "Parameters[].Name | sort(@)"
```

O comando retorna informações como as seguintes.

```
[
    "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/geolocationCountry",
    "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/geolocationRegion",
    "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/location",
    "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/network-border-
group",
    "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/parent-availability-
zone",
    "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/parent-region",
```

```
    "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/zone-group"
]
```

Demonstrações do Parameter Store

A demonstração nesta seção mostra como criar, armazenar e executar parâmetros com o Parameter Store, uma ferramenta do AWS Systems Manager, em um ambiente de teste. Essas demonstrações mostram como usar o Parameter Store com outras ferramentas do Systems Manager. Você pode usar o Parameter Store também com outros Serviços da AWS. Para obter mais informações, consulte O que é um parâmetro ?.

Conteúdo

- Criar um parâmetro SecureString no Parameter Store e integrar um nó a um domínio (PowerShell)

Criar um parâmetro SecureString no Parameter Store e integrar um nó a um domínio (PowerShell)

Esta demonstração mostra como associar um nó do Windows Server a um domínio usando os parâmetros AWS Systems Manager `SecureString` e Run Command. A demonstração usa parâmetros de domínio típicos, como o nome do domínio e um nome do usuário do domínio. Esses valores são transmitidos como valores de string não criptografados. A senha do domínio é criptografada usando uma chave mestra de cliente (CMK) gerenciada pela Chave gerenciada pela AWS e transmitida como uma string criptografada.

Pré-requisitos

Esta demonstração pressupõe que você já tenha especificado seu nome de domínio e o endereço IP do servidor DNS no conjunto de opções de DHCP associado à sua Amazon VPC. Para obter mais informações, consulte Working with DHCP Options no Guia do usuário da Amazon VPC.

Para criar um parâmetro **`SecureString`** e ingressar um nó em um domínio

1. Insira parâmetros no sistema usando o AWS Tools for Windows PowerShell.

   Nos comandos a seguir, substitua cada *`espaço reservado para entrada do usuário`* por suas próprias informações.

   ```
   Write-SSMParameter -Name "domainName" -Value "DOMAIN-NAME" -Type String
   Write-SSMParameter -Name "domainJoinUserName" -Value "DOMAIN\USERNAME" -Type String
   Write-SSMParameter -Name "domainJoinPassword" -Value "PASSWORD" -Type SecureString
   ```

> **Important**
>
> Somente o valor de um parâmetro `SecureString` é criptografado. O nome do parâmetro, a descrição e outras propriedades não são criptografados.

2. Anexe as seguintes políticas do AWS Identity and Access Management (IAM) às permissões de função do IAM para o nó:

   - AmazonSSMManagedInstanceCore – necessário. Essa política gerenciada da AWS permite que um nó gerenciado use a funcionalidade básica do serviço Systems Manager.
   - AmazonSSMDirectoryServiceAccess – necessário. Essa política gerenciada da AWS permite que o SSM Agent acesse o AWS Directory Service em seu nome para solicitações para ingressar no domínio pelo nó gerenciado.
   - Uma política personalizada para acesso ao bucket do S3: obrigatória. O SSM Agent, localizado em um nó, executa tarefas do Systems Manager e requer acesso a buckets do Amazon Simple Storage Service (Amazon S3) específicos, de propriedade da Amazon. Na política de bucket do S3 personalizada que você cria, também é possível fornecer acesso aos buckets do S3 de sua propriedade que sejam necessários para as operações do Systems Manager.

     Exemplos: você pode gravar a saída para comandos do Run Command ou sessões do Session Manager para um bucket do S3 e usar essa saída posteriormente para a auditoria ou para solução de problemas. Você armazena scripts de acesso ou listas de linha de base de patches personalizados em um bucket do S3 e faz referência ao script ou à lista quando executa um comando, ou quando uma linha de base de patch é aplicada.

     Para obter informações sobre como criar uma política personalizada para acesso ao bucket do Amazon S3, consulte Criar uma política personalizada de bucket do S3 para um perfil de instância

     > **Note**
     >
     > Salvar os dados de log de saída em um bucket do S3 é opcional, mas recomendamos configurá-lo no início do seu processo de configuração do Systems Manager caso tenha decidido usá-lo. Para obter mais informações, consulte Criar um bucket no Guia do usuário do Amazon Simple Storage Service.

- CloudWatchAgentServerPolicy – opcional. Essa política gerenciada da AWS permite que você execute o agente do CloudWatch em nós gerenciados. Essa política possibilita ler informações sobre um nó e gravá-las no Amazon CloudWatch. Seu perfil de instância precisará dessa política somente se você pretende usar recursos do Amazon EventBridge ou o CloudWatch Logs.

> **Note**
>
> Usar recursos do CloudWatch e do EventBridge é opcional, mas recomendamos configurá-los no início do seu processo de configuração do Systems Manager se você decidiu usá-los. Para obter mais informações, consulte o Manual do usuário do Amazon EventBridge e o Manual do usuário do Amazon CloudWatch Logs.

3. Edite a função do IAM anexada ao nó e adicione a seguinte política: Essa política fornece as permissões do nó para chamar o `kms:Decrypt` e a API `ssm:CreateDocument`.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "ssm:CreateDocument"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/kms-key-id"
            ]
        }
    ]
}
```

4. Copie e cole o seguinte texto do json em um editor de texto e salve o arquivo como `JoinInstanceToDomain.json` no seguinte local: `c:\temp\JoinInstanceToDomain.json`.

```
{
    "schemaVersion": "2.2",
```

```
   "description": "Run a PowerShell script to securely join a Windows Server
 instance to a domain",
   "mainSteps": [
       {
           "action": "aws:runPowerShellScript",
           "name": "runPowerShellWithSecureString",
           "precondition": {
               "StringEquals": [
                   "platformType",
                   "Windows"
               ]
           },
           "inputs": {
               "runCommand": [
                   "$domain = (Get-SSMParameterValue -Name
 domainName).Parameters[0].Value",
                   "if ((gwmi Win32_ComputerSystem).domain -eq $domain){write-host
 \"Computer is part of $domain, exiting\"; exit 0}",
                   "$username = (Get-SSMParameterValue -Name
 domainJoinUserName).Parameters[0].Value",
                   "$password = (Get-SSMParameterValue -Name domainJoinPassword -
WithDecryption $True).Parameters[0].Value | ConvertTo-SecureString -asPlainText -
Force",
                   "$credential = New-Object
 System.Management.Automation.PSCredential($username,$password)",
                   "Add-Computer -DomainName $domain -Credential $credential -
ErrorAction SilentlyContinue -ErrorVariable domainjoinerror",
                   "if($?){Write-Host \"Instance joined to domain successfully.
 Restarting\"; exit 3010}else{Write-Host \"Instance failed to join domain with
 error:\" $domainjoinerror; exit 1 }"
               ]
           }
       }
   ]
}
```

5. Execute o comando a seguir no Tools for Windows PowerShell para criar um novo documento do SSM.

```
$json = Get-Content C:\temp\JoinInstanceToDomain | Out-String
New-SSMDocument -Name JoinInstanceToDomain -Content $json -DocumentType Command
```

6. Execute o comando a seguir no Tools for Windows PowerShell para associar o nó ao domínio.

```
Send-SSMCommand -InstanceId instance-id -DocumentName JoinInstanceToDomain
```

Se houver êxito, o comando retornará informações semelhantes às seguintes:

```
WARNING: The changes will take effect after you restart the computer EC2ABCD-
EXAMPLE.
Domain join succeeded, restarting
Computer is part of example.local, exiting
```

Se houver êxito, o comando retornará informações semelhantes às seguintes:

```
Failed to join domain with error:
Computer 'EC2ABCD-EXAMPLE' failed to join domain 'example.local'
from its current workgroup 'WORKGROUP' with following error message:
The specified domain either does not exist or could not be contacted.
```

Auditar e registrar atividades do Parameter Store em log

O AWS CloudTrail captura chamadas à API feitas no console do AWS Systems Manager, na AWS Command Line Interface (AWS CLI) e no SDK do Systems Manager. Você pode visualizar as informações no console do CloudTrail ou em um bucket do Amazon Simple Storage Service (Amazon S3). Todos os logs do CloudTrail para sua conta usam um bucket. Para obter mais informações sobre como visualizar e utilizar os logs do CloudTrail de atividades do Systems Manager, consulte Registrar em log chamadas de API do AWS Systems Manager com o AWS CloudTrail. Para obter mais informações sobre opções de auditoria e registro para o Systems Manager, consulte Registrar em log e monitorar no AWS Systems Manager.

Solução de problemas do Parameter Store

Use as informações a seguir para obter ajuda para solucionar problemas com o Parameter Store, uma ferramenta do AWS Systems Manager.

Solução de problemas de criação de parâmetros **`aws:ec2:image`**

Use as seguintes informações para ajudar a solucionar problemas com a criação de parâmetros de tipo de dados `aws:ec2:image`.

Sem permissão para criar uma instância

Problema: você tenta criar uma instância usando um parâmetro `aws:ec2:image`, mas recebe uma mensagem de erro semelhante a "Você não está autorizado a executar esta operação".

- Solução: você não tem todas as permissões necessárias para criar uma instância do EC2 usando um valor de parâmetro, como permissões para `ec2:RunInstances`, `ec2:DescribeImages` e `ssm:GetParameter`, entre outras. Entre em contato com um usuário com permissões de administrador em sua organização para solicitar as permissões necessárias.

O EventBridge relata a mensagem de falha “Não é possível descrever o recurso”

Problema: você executou um comando para criar um parâmetro `aws:ec2:image`, mas a criação do parâmetro falhou. Você recebe uma notificação do Amazon EventBridge que relata a exceção "Unable to Describe Resource" (“Não é possível descrever o recurso”).

Solução: essa mensagem pode indicar o seguinte:

- Você não tem todas as permissões necessárias para a operação de API `ec2:DescribeImages` ou não tem permissões para acessar a imagem específica referenciada no parâmetro. Entre em contato com um usuário que tem permissões de administrador em sua organização para solicitar as permissões necessárias.
- O ID da Amazon Machine Image (AMI) que você inseriu como um valor de parâmetro não é válido. Insira o ID de uma AMI que esteja disponível na Região da AWS e conta atuais em que você está trabalhando.

O novo parâmetro **`aws:ec2:image`** não está disponível

Problema: você acabou de executar um comando para criar um parâmetro `aws:ec2:image` e um número de versão foi relatado, mas o parâmetro não está disponível.

- Solução: quando você executa o comando para criar um parâmetro que usa o tipo de dados `aws:ec2:image`, um número de versão é gerado para o parâmetro imediatamente, mas o formato do parâmetro deve ser validado antes que o parâmetro esteja disponível. Esse processo pode levar até alguns minutos. Para monitorar o processo de criação e validação de parâmetros, você pode fazer o seguinte:
  - Use o EventBridge para enviar notificações sobre as operações dos parâmetros `create` e `update`. Essas notificações relatam se uma operação de parâmetro foi bem-sucedida ou

não. Para obter informações sobre como se inscrever em eventos do Parameter Store no EventBridge, consulte Configurar notificações ou acionar ações com base nos eventos do Parameter Store.

- Na seção Parameter Store do console do Systems Manager, atualize a lista de parâmetros periodicamente para verificar os detalhes do parâmetro novo ou atualizado.
- Use o comando GetParameter para verificar o parâmetro novo ou atualizado. Por exemplo, usando a AWS Command Line Interface (AWS CLI):

```
aws ssm get-parameter name MyParameter
```

Para um novo parâmetro, uma mensagem `ParameterNotFound` é retornada até que o parâmetro seja validado. Para um parâmetro existente que você estiver atualizando, as informações sobre a nova versão não serão incluídas até que o parâmetro seja validado.

Se você tentar criar ou atualizar o parâmetro novamente antes que o processo de validação seja concluído, o sistema informará que a validação ainda está em andamento. Se o parâmetro não for criado ou atualizado com êxito, você poderá tentar novamente após cinco minutos da tentativa original.

Ferramentas de operações do AWS Systems Manager

As Ferramentas de operações são um conjunto de recursos que ajudam você a gerenciar seus recursos da AWS.

- Os painéis do Amazon CloudWatch hospedados pelo Systems Manager são páginas iniciais personalizáveis no console do CloudWatch que você pode usar para monitorar seus recursos em Regiões da AWS em uma única visualização.
- O Explorer é um painel de operações personalizável que relata informações sobre seus recursos da AWS. O Explorer exibe uma visualização agregada dos dados de operações (OpsData) para suas Contas da AWS e em todas as Regiões da AWS.
- O Incident Manager ajuda você a mitigar e se recuperar de incidentes que afetam suas aplicações hospedadas na AWS.
- O OpsCenter fornece um local central onde engenheiros de operações e profissionais de TI podem visualizar, investigar e resolver itens de trabalho operacionais (OpsItems) relacionados a recursos da AWS.

Tópicos

- AWS Systems Manager Incident Manager
- AWS Systems Manager Explorer
- AWS Systems Manager OpsCenter
- Usar painéis do Amazon CloudWatch hospedados pelo Systems Manager

AWS Systems Manager Incident Manager

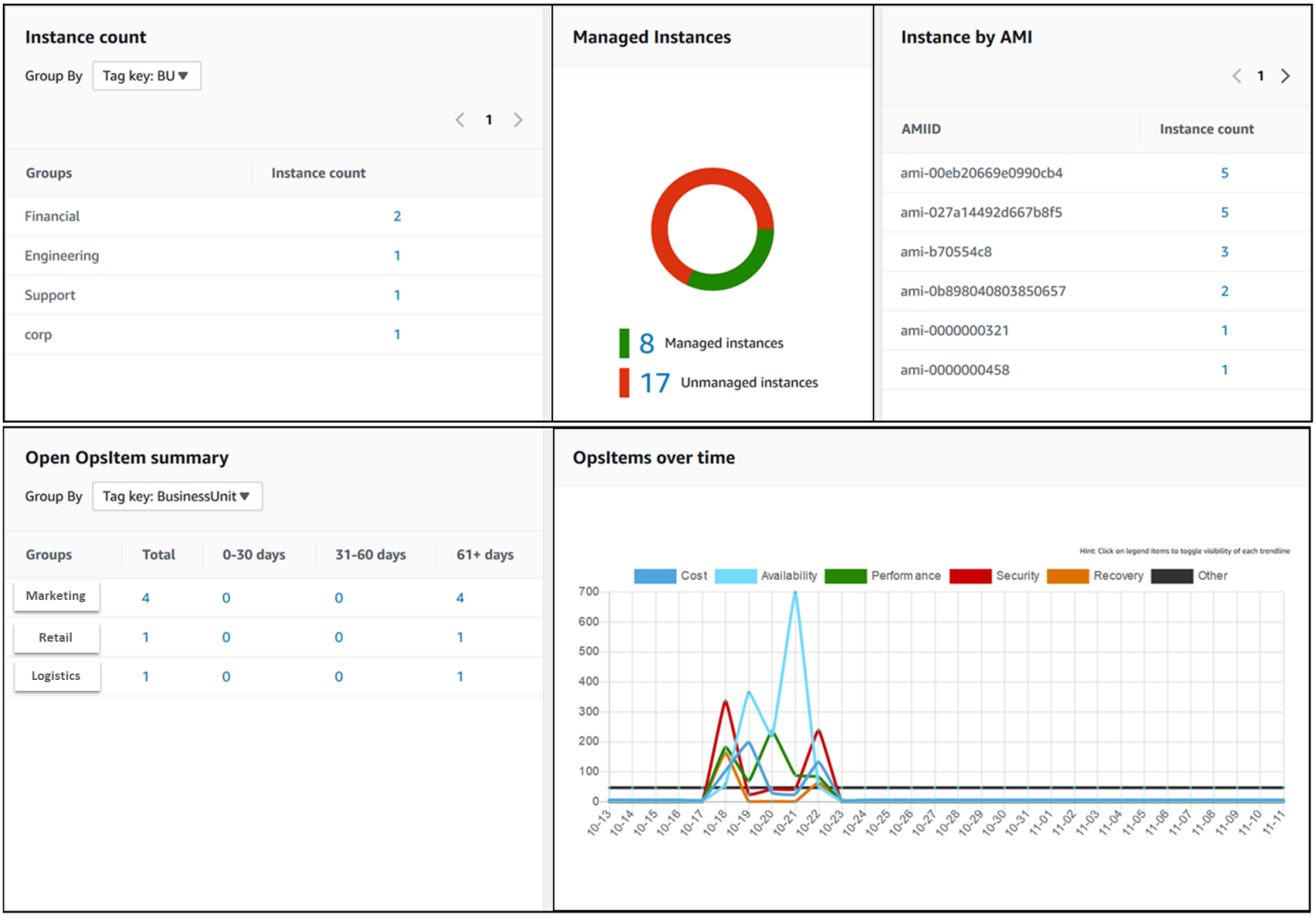
Use o Incident Manager, uma ferramenta do AWS Systems Manager, para gerenciar incidentes que ocorrerem nas aplicações hospedadas pela AWS. O Incident Manager combina engajamentos de usuários, escalonamento, runbooks, planos de resposta, canais de conversa e análise pós-incidente para ajudar sua equipe a fazer a triagem de incidentes mais rapidamente e retornar suas aplicações ao normal. Para saber mais sobre o Incident Manager, consulte o Manual do usuário do Incident.

AWS Systems Manager Explorer

O AWS Systems Manager Explorer é um painel de operações personalizável que informa sobre seus recursos da AWS. O Explorer exibe uma visualização agregada dos dados de operações (OpsData) para as Contas da AWS em todas as Regiões da AWS. No Explorer, o OpsData inclui metadados sobre os nós gerenciados no seu ambiente híbrido e multinuvem. O OpsData também inclui informações fornecidas por outras ferramentas do Systems Manager, incluindo detalhes de conformidade de patches do Patch Manager e de conformidade de associações do State Manager. Para simplificar ainda mais a forma como você acessa o OpsData, o Explorer exibe informações de serviços de suporte da AWS como AWS Config, AWS Trusted Advisor, AWS Compute Optimizer e AWS Support (casos de suporte).

Para aumentar a conscientização operacional, o Explorer também exibe itens de trabalho operacionais (OpsItems). O Explorer fornece contexto sobre como os OpsItems são distribuídos em suas unidades de negócios ou aplicações, a tendência ao longo do tempo e como elas variam de acordo com a categoria. Você pode agrupar e filtrar informações no Explorer para se concentrar em itens que são relevantes para você e que exigem ação. Ao identificar problemas de alta prioridade, você pode usar o Systems Manager OpsCenter para executar runbooks do Automation e resolver rapidamente esses problemas. Para começar a usar o Explorer, abra o Systems Manager console (Console do gerenciador de sistemas). No painel de navegação, escolha Explorer.

A imagem a seguir mostra algumas das caixas de relatório individuais, chamadas widgets, que estão disponíveis no Explorer.



Quais são os recursos do Explorer?

O Explorer inclui os seguintes recursos:

- Exibição personalizável de informações práticas: o Explorer inclui widgets de arrastar e soltar que exibem automaticamente informações práticas sobre os recursos da AWS. O Explorer exibe informações em dois tipos de widget.
  - Widgets informativos: esses widgets resumem dados do Amazon EC2, do Patch Manager, do State Manager e de Serviços da AWS de suporte, como o AWS Trusted Advisor, o AWS Compute Optimizer e o Suporte. Esses widgets fornecem contexto importante para ajudar você a entender o estado e os riscos operacionais de seus recursos da AWS. Exemplos de widgets informativos incluem Instance count (Contagem de instâncias), Instance by AMI (Instância por AMI), Total noncompliant nodes (Total de nós sem conformidade), Non-compliant associations (Associações sem conformidade) e Support Center cases (Casos da Central de suporte).

  - Widgets do OpsItem: Um OpsItem do Systems Manager é um item de trabalho operacional relacionado a um ou mais recursos da AWS. OpsItems são recursos do Systems Manager OpsCenter. O OpsItems pode exigir que os engenheiros de DevOps investiguem e potencialmente corrijam um problema. Os exemplos de possíveis OpsItems incluem alta utilização de CPU pela instância do EC2, volumes do Amazon Elastic Block Store (Amazon EBS) desanexados, falha de implantação do AWS CodeDeploy ou falha na execução do Systems Manager Automation. Exemplos de widgets do OpsItem incluem Open OpsItem summary (Abrir o resumo do item operacional), OpsItem by status (OpsItem por status), e OpsItems over time (OpsItem no decorrer do tempo).
- Filtros: cada widget oferece a opção de filtrar informações com base na Conta da AWS, Região da AWS e tag. Os filtros ajudam você a refinar rapidamente as informações exibidas no Explorer.
- Links diretos para telas de serviço: para ajudar você a investigar problemas com recursos da AWS, os widgets do Explorer contêm links diretos para telas de serviço relacionadas. Os filtros aplicados a um widget permanecerão em vigor se você navegar para uma tela de serviço relacionada.
- Grupos: para ajudar você a entender os tipos de problemas operacionais em toda a organização, alguns widgets permitem agrupar dados com base em conta, região e tag.
- Chaves de tag de relatório: ao configurar o Explorer, você pode especificar até cinco chaves de tag. Essas chaves ajudam você a agrupar e filtrar dados no Explorer. Se uma chave especificada corresponder a uma chave em um recurso que gera um OpsItem, a chave e o valor serão incluídos no OpsItems.
- Três modos de exibição da Conta da AWS e Região da AWS: o Explorer inclui os seguintes modos de exibição para OpsData e OpsItems em Contas da AWS e Regiões da AWS:
  - Conta única/região única: esta é a visualização padrão. Esse modo permite que os usuários visualizem dados e OpsItems em sua própria conta e na região atual.
  - Conta única/várias regiões: este modo requer que você crie uma ou mais sincronizações de dados de recursos usando a página Settings (Configurações) do Explorer. Uma sincronização de dados de recurso agrega OpsData de uma ou mais regiões. Depois de criar uma sincronização de dados de recurso, você pode alternar a sincronização a ser usada no painel do Explorer. Em seguida, você pode filtrar e agrupar os dados com base na região.
  - Várias contas/várias regiões: esse modo requer que sua organização ou empresa use o AWS Organizations com a opção All features (Todos os recursos) habilitada. Depois de configurar o AWS Organizations em seu ambiente de computação, você poderá agregar todos os dados da conta em uma conta mestra. Em seguida, você pode criar sincronizações de dados de recursos para poder filtrar e agrupar os dados com base na região. Para obter mais informações sobre o modo Todos os recursos na organização, consulte Ativar todos os recursos na organização.

- Relatórios: você pode exportar relatórios do Explorer, como arquivos de valores separados por vírgula (.csv), para um bucket do Amazon Simple Storage Service (Amazon S3). Você recebe um alerta do Amazon Simple Notification Service (Amazon SNS) quando uma exportação for concluída.

Como o Explorer está relacionado ao OpsCenter?

O Systems Manager OpsCenter fornece um local central onde os engenheiros de operações e profissionais de TI visualizam, investigam e resolvem OpsItems relacionados aos recursos da AWS. O Explorer é um hub de relatórios no qual os gerentes de DevOps visualizam resumos agregados de seus dados de operações, incluindo OpsItems, em regiões e contas da Regiões da AWS. O Explorer ajuda os usuários a descobrir tendências e padrões e, se necessário, resolver rapidamente problemas usando runbooks do Systems Manager Automation.

Agora, a configuração do OpsCenter está integrada à configuração do Explorer. Se você já tiver configurado o OpsCenter, o Explorer exibirá automaticamente os dados de operações, incluindo informações agregadas sobre OpsItems. Se você não tiver configurado o OpsCenter, poderá usar a instalação do Explorer para começar a usar ambas as ferramentas. Para ter mais informações, consulte Conceitos básicos do Systems Manager Explorer e do OpsCenter.

O que é OpsData?

OpsData são todos os dados de operações exibidos no painel do Systems Manager. O Explorer recupera OpsData das seguintes fontes:

- Amazon Elastic Compute Cloud (Amazon EC2)

  Os dados exibidos no Explorer incluem: número total de instâncias, número total de nós gerenciados e não gerenciados e uma contagem de nós usando uma Amazon Machine Image (AMI) específica.
- Systems Manager OpsCenter

  Os dados exibidos no Explorer incluem: uma contagem de OpsItems por status, uma contagem de OpsItems por gravidade, uma contagem de aberturas de OpsItems entre grupos e períodos de 30 dias e dados históricos de OpsItems ao longo do tempo.
- Patch Manager do Systems Manager

  Os dados exibidos em Explorer incluem uma contagem de nós sem conformidade e sem conformidade críticos.

- AWS Trusted Advisor

  Os dados exibidos no Explorer incluem: status das verificações de práticas recomendadas para instâncias reservadas do EC2 nas áreas de otimização de custos, segurança, tolerância a falhas, performance e limites de serviço.

- AWS Compute Optimizer

  Os dados exibidos no Explorer incluem: uma contagem de instâncias do EC2, descobertas de otimização, detalhes de preço sob demanda e recomendações para preço e tipo de instância Subprovisionados e Superprovisionados.

- Casos do Suporte Center

  Dados exibidos no Explorer incluem: ID do caso, gravidade, status, hora de criação, assunto, serviço e categoria.

- AWS Config

  Os dados exibidos no Explorer incluem: um resumo geral das regras de conformidade e não-conformidade do AWS Config, o número de recursos compatíveis e não compatíveis e detalhes específicos sobre cada um (quando você aprofunda em uma regra ou recurso não compatível).

- AWS Security Hub

  Dados exibidos no Explorer incluem: resumo geral das descobertas do Security Hub, o número de cada descoberta agrupada por gravidade e detalhes específicos sobre a descoberta.

> **Note**
>
> Para visualizar casos do AWS Trusted Advisor e do Suporte Center no Explorer, você deve ter uma conta Enterprise ou Business configurada com o AWS Support.

Você pode exibir e gerenciar fontes de OpsData na página Settings (Configurações) do Explorer. Para obter informações sobre como configurar serviços que preenchem widgets do Explorer com OpsData, consulte Configurar serviços relacionados para o Explorer.

Há cobrança pelo uso do Explorer?

Sim. Ao ativar as regras padrão para a criação de OpsItems durante a instalação integrada, você inicia um processo que cria automaticamente OpsItems. Sua conta é cobrada com base no número

de OpsItems criados por mês. Sua conta também é cobrada com base no número de chamadas das APIs `GetOpsItem`, `DescribeOpsItem`, `UpdateOpsItem` e `GetOpsSummary` feitas por mês. Além disso, você pode ser cobrado por chamadas de API públicas para outros serviços que expõem informações de diagnóstico relevantes. Para obter mais informações, consulte Preços do AWS Systems Manager.

Tópicos

- Conceitos básicos do Systems Manager Explorer e do OpsCenter
- Usar o Explorer
- Exportar OpsData do Systems Manager Explorer
- Solução de problemas do Systems Manager Explorer

Conceitos básicos do Systems Manager Explorer e do OpsCenter

O AWS Systems Manager usa uma experiência de configuração integrada para ajudar você a começar a usar o Systems Manager Explorer e o Systems Manager OpsCenter. Nesta documentação, a configuração do Explorer e do OpsCenter é chamada de configuração integrada. Se você já tiver configurado o OpsCenter, ainda precisará concluir a configuração integrada para verificar as definições e opções. Se você não tiver configurado o OpsCenter, poderá usar a configuração integrada para começar a usar ambas as ferramentas.

> **Note**
>
> A configuração integrada está disponível somente no console do Systems Manager. Você não pode configurar o Explorer ou o OpsCenter programaticamente.

A configuração integrada executa as seguintes tarefas:

- Configure funções e permissões: a configuração integrada cria uma função do AWS Identity and Access Management (IAM) que permite ao Amazon EventBridge criar automaticamente um OpsItems com base em regras padrão. Após a configuração, você deve configurar as permissões de usuário, grupo ou perfil para o OpsCenter, conforme descrito nesta seção.
- Permite regras padrão para a criação de OpsItem: a configuração integrada cria regras padrão no EventBridge. Essas regras criam automaticamente OpsItems em resposta a eventos. Alguns exemplos desses eventos são: alteração do estado de um recurso da AWS, alteração nas configurações de segurança ou um serviço que fica indisponível.

- Ativa as fontes de OpsData: a configuração integrada as fontes de dados a seguir preenchem os widgets do Explorer.
  - Suporte Center (Você deve ter um plano Business ou Enterprise Support para ativar essa fonte).
  - AWS Compute Optimizer (Você deve ter um plano Business ou Enterprise Support para ativar essa fonte).
  - Conformidade de associações do Systems Manager State Manager
  - AWS Config Compatibilidade
  - Systems Manager OpsCenter
  - Conformidade de patches do Systems Manager Patch Manager
  - Amazon Elastic Compute Cloud (Amazon EC2)
  - Inventário do Systems Manager
  - AWS Trusted Advisor (Você deve ter um plano Business ou Enterprise Support para ativar essa fonte).
  - AWS Security Hub

> **Note**
>
> Você pode alterar as opções de configuração a qualquer momento na página Settings (Configurações).

Depois de concluir a instalação integrada, recomendamos que você Set up (Configurar) o Explorer para exibir dados de várias regiões e contas. O Explorer e o OpsCenter sincronizam automaticamente OpsData e OpsItems para a Conta da AWS e a Região da AWS que você usou quando concluiu a configuração integrada. Você pode agregar OpsData e OpsItems de outras contas e regiões criando uma sincronização de dados de recurso.

Configurar serviços relacionados para o Explorer

O AWS Systems Manager Explorer e o AWS Systems Manager OpsCenter coletam informações ou interagem com outros Serviços da AWS e ferramentas do Systems Manager. Recomendamos instalar e configurar esses outros serviços ou ferramentas antes de usar a configuração integrada.

A tabela a seguir inclui tarefas que permitem que o Explorer e o OpsCenter coletem informações ou interajam com outros Serviços da AWS e ferramentas do Systems Manager.

Tarefa	Informações
Verifique as permissões no Systems Manager Automation	O Explorer e o OpsCenter permitem corrigir problemas em recursos da AWS usando documentos do Systems Manager Automatio n (runbooks). Para usar essa ferramenta de correção, é necessário ter permissão para executar runbooks do Systems Manager Automation. Para ter mais informações, consulte Configurar a automação.
Instale e configure o Systems Manager Patch Manager	O Explorer inclui um widget que fornece informações sobre a conformidade de patches. Para visualizar esses dados no Explorer, é necessário configurar a aplicação de patches. Para ter mais informações, consulte AWS Systems Manager Patch Manager.
Instale e configure o Systems Manager State Manager	O Explorer inclui um widget que fornece informações sobre o a conformidade de associações do State Manager no Systems Manager. Para visualizar esses dados no Explorer, é necessário configurar o State Manager. Para ter mais informações, consulte AWS Systems Manager State Manager.
Ative o Gravador de configurações do AWS Config	O Explorer usa dados fornecidos pelo gravador de configuração do AWS Config para preencher widgets com informações sobre suas instâncias do EC2. Para visualizar esses dados no Explorer, ative o gravador de configurações do AWS Config. Para obter mais informações, consulte Gerenciar o gravador de configuração.

Tarefa	Informações
	Note Depois que você permitir o gravador de configuração, o Systems Manager poderá levar até seis horas para exibir dados nos widgets do Explorer que exibirá informações sobre suas instâncias do EC2.
Ativar o AWS Trusted Advisor	O Explorer usa os dados fornecidos pelo Trusted Advisor para exibir um status de verificações das práticas recomendadas para instâncias reservadas do Amazon EC2 nas áreas de otimização de custos, segurança , tolerância a falhas, performance e limites de serviço. Para visualizar esses dados no Explorer, é necessário ter um plano de suporte comercial ou empresarial. Para ter mais informações, consulte Suporte.
Ativar o AWS Compute Optimizer	O Explorer usa os dados fornecidos pelo Compute Optimizer para exibir detalhes de uma contagem de instâncias do EC2 Under provisioned (Subprovisionadas) e Over provisioned (Superprovisionadas), descobert as de otimização, detalhes de preço sob demanda e recomendações para tipo e preço de instâncias. Para visualizar esses dados no Explorer, ative o Compute Optimizer. Para obter mais informações, consulte Conceitos básicos do AWS Compute Optimizer.

Tarefa	Informações
Ativar o AWS Security Hub	O Explorer usa dados fornecidos pelo Security Hub para preencher widgets com informações sobre suas descobertas de segurança. Para visualizar esses dados no Explorer, ative a integração do Security Hub. Para obter mais informações, consulte O que é o AWS Security Hub.

Configurar funções e permissões para o Systems Manager Explorer

A configuração integrada cria e configura automaticamente perfis do AWS Identity and Access Management (IAM) para o AWS Systems Manager Explorer e o AWS Systems Manager OpsCenter. Se você concluiu a configuração integrada, não precisará executar nenhuma tarefa adicional para configurar funções e permissões para o Explorer. No entanto, você deve configurar a permissão para o OpsCenter, conforme descrito mais adiante neste tópico.

A configuração integrada cria e configura as seguintes funções para trabalhar com o Explorer e o OpsCenter.

- `AWSServiceRoleForAmazonSSM`: fornece acesso a recursos da AWS gerenciados ou usados pelo Systems Manager.
- `OpsItem-CWE-Role`: permite que o CloudWatch Events e o EventBridge criem OpsItems em resposta aos eventos comuns.
- `AWSServiceRoleForAmazonSSM_AccountDiscovery`: permite que o Systems Manager chame outros Serviços da AWS para descobrir informações da Conta da AWS ao sincronizar os dados. Para obter mais informações sobre essa função, consulte Usar perfis para coletar informações da Conta da AWS para o OpsCenter e o Explorer.
- `AmazonSSMExplorerExport`: permite que o Explorer exporte OpsData para um arquivo de valores separados por vírgula (CSV).

Se você configurar o Explorer para exibir dados de várias contas e regiões usando o AWS Organizations e uma sincronização de dados do recurso, o Systems Manager criará um perfil `AWSServiceRoleForAmazonSSM_AccountDiscovery` vinculado ao serviço. O Systems Manager

usa essa função para obter informações sobre sua Contas da AWS no AWS Organizations. A função utiliza a seguinte política de permissões.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "organizations:DescribeAccount",
            "organizations:DescribeOrganization",
            "organizations:ListAccounts",
            "organizations:ListAWSServiceAccessForOrganization",
            "organizations:ListChildren",
            "organizations:ListParents"
         ],
         "Resource":"*"
      }
   ]
}
```

Para obter mais informações sobre a função `AWSServiceRoleForAmazonSSM_AccountDiscovery`, consulte Usar perfis para coletar informações da Conta da AWS para o OpsCenter e o Explorer.

Configurar permissões para o Systems Manager OpsCenter

Após concluir a configuração integrada, você deve configurar as permissões de usuário, grupo ou perfil para que os usuários possam executar ações no OpsCenter.

Antes de começar

É possível configurar o OpsCenter para criar e gerenciar OpsItems em uma única conta ou em diversas contas. Se você configurar o OpsCenter para criar e gerenciar OpsItems em diversas contas, poderá utilizar a conta do administrador delegado do Systems Manager ou a conta de gerenciamento do AWS Organizations para criar, visualizar ou editar OpsItems manualmente em outras contas. Para obter mais informações sobre a conta de administrador delegado do Systems Manager, consulte Configuração de um administrador delegado para Explorer.

Se você configurar o OpsCenter para uma única conta, só poderá visualizar ou editar OpsItems na conta em que os OpsItems foram criados. Você não pode compartilhar nem transferir OpsItems entre contas da Contas da AWS. Por esse motivo, recomendamos configurar as permissões para OpsCenter na Conta da AWS usada para executar as cargas de trabalho da AWS. Assim, você pode criar usuários ou grupos do nessa conta. Dessa maneira, vários engenheiros de operações ou profissionais de TI podem criar, visualizar e editar o OpsItems na mesma conta da Conta da AWS.

Explorer e OpsCenter usam as seguintes operações da API: Você poderá usar todos os recursos do Explorer e do OpsCenter se seu usuário, grupo ou perfil tiver acesso a essas ações. Você também pode criar acesso mais restritivo, conforme descrito posteriormente nesta seção.

- CreateOpsItem
- CreateResourceDataSync
- DescribeOpsItems
- DeleteResourceDataSync
- GetOpsItem
- GetOpsSummary
- ListResourceDataSync
- UpdateOpsItem
- UpdateResourceDataSync

Se preferir, é possível especificar a permissão de somente leitura ao adicionar a política em linha a seguir à sua conta, ao seu grupo ou ao seu perfil.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetOpsItem",
        "ssm:GetOpsSummary",
        "ssm:DescribeOpsItems",
        "ssm:GetServiceSetting",
```

```
        "ssm:ListResourceDataSync"
      ],
      "Resource": "*"
    }
  ]
}
```

Para obter mais informações sobre como criar e editar políticas de usuários do IAM, consulte Criar políticas do IAM, no Manual do usuário do IAM. Para obter informações sobre como atribuir essa política a um grupo do IAM, consulte Anexar uma política a um grupo do IAM.

Crie uma permissão usando o seguinte e adicione-a aos seus usuários, grupos ou perfis:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetOpsItem",
        "ssm:UpdateOpsItem",
        "ssm:DescribeOpsItems",
        "ssm:CreateOpsItem",
        "ssm:CreateResourceDataSync",
        "ssm:DeleteResourceDataSync",
        "ssm:ListResourceDataSync",
        "ssm:UpdateResourceDataSync"

      ],
      "Resource": "*"
    }
  ]
}
```

Dependendo da aplicação de identidade que você usa em sua organização, é possível selecionar qualquer uma das opções a seguir para configurar o acesso do usuário.

Para conceder acesso, adicione as permissões aos seus usuários, grupos ou perfis:

- Usuários e grupos no AWS IAM Identity Center:

  Crie um conjunto de permissões. Siga as instruções em Criação de um conjunto de permissões no Guia do usuário do AWS IAM Identity Center.
- Usuários gerenciados no IAM com provedor de identidades:

  Crie um perfil para a federação de identidades. Siga as instruções em Criando um perfil para um provedor de identidades de terceiros (federação) no Guia do Usuário do IAM.
- Usuários do IAM:
  - Crie um perfil que seu usuário possa assumir. Siga as instruções em Criação de um perfil para um usuário do IAM no Guia do usuário do IAM.
  - (Não recomendado) Vincule uma política diretamente a um usuário ou adicione um usuário a um grupo de usuários. Siga as instruções em Adição de permissões a um usuário (console) no Guia do usuário do IAM.

Restringir o acesso aos OpsItems usando tags

Você também pode restringir o acesso aos OpsItems usando uma política do IAM em linha que especifique tags. Veja um exemplo que especifica uma chave de tag de Departamento e um valor de tag de Finanças. Com essa política, o usuário só pode chamar a operação de API GetOpsItem para visualizar os OpsItems que foram marcados anteriormente com Key=Department and Value=Finance. Os usuários não podem visualizar nenhum outro OpsItems.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetOpsItem"
             ],
      "Resource": "*"
      ,
      "Condition": { "StringEquals": { "ssm:resourceTag/Department":
 "Finance" } }
    }
  ]
```

```
}
```

Veja a seguir um exemplo que especifica operações de API para visualizar e atualizar OpsItems. Essa política também especifica dois conjuntos de pares de chave-valor da tag: Departamento-Finança e Projeto-Unidade.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "ssm:GetOpsItem",
            "ssm:UpdateOpsItem"
         ],
         "Resource":"*",
         "Condition":{
            "StringEquals":{
               "ssm:resourceTag/Department":"Finance",
               "ssm:resourceTag/Project":"Unity"
            }
         }
      }
   ]
}
```

Para obter informações sobre como adicionar tags a um OpsItem, consulte Criar OpsItems manualmente.

Noções básicas sobre regras do EventBridge criadas pela configuração integrada

Durante o processo de configuração integrada do Explorer e do OpsCenter, é possível escolher habilitar várias regras padrão baseadas em eventos detectados pelo Amazon EventBridge. Quando esses eventos são detectados, o sistema automaticamente cria OpsItems em AWS Systems Manager OpsCenter.

Por exemplo, a regra `SSMOpsItems-Autoscaling-instance-termination-failure` resulta na criação de um OpsItem quando o término da instância de ajuste de escala automático do EC2 falhar.

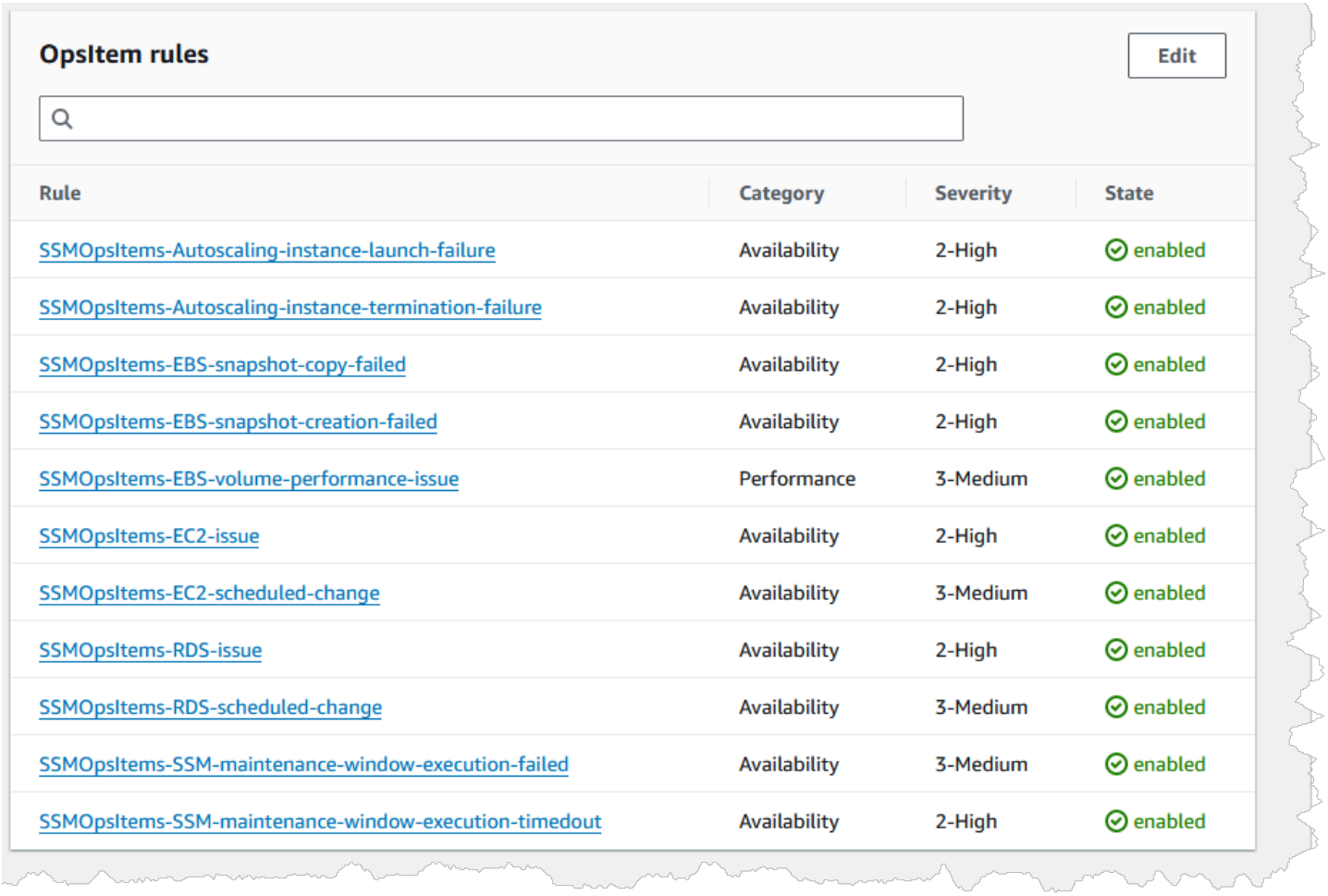
A regra `SSMOpsItems-SSM-maintenance-window-execution-failed` resulta na criação de um OpsItem quando uma janela de manutenção do Systems Manager não for concluída com êxito.

Para obter instruções de configuração e descrições de todas as regras do EventBridge que você pode habilitar durante o processo de configuração, consulte Configurar o OpsCenter.

Se você não quiser que o EventBridge crie OpsItems para os eventos a seguir, é possível escolher não habilitar esta opção na configuração integrada. Se preferir, você poderá especificar o OpsCenter como o destino de eventos específicos do EventBridge. Para ter mais informações, consulte Configurar regras do EventBridge para criar OpsItems.

Você pode desabilitar uma regra padrão ou alterar sua categoria e nível de severidade na página Configurações do OpsCenter escolhendo OpsCenter, Configurações e, em seguida, Editar na área de Regras do OpsItem.

Você também pode editar a categoria ou a severidade atribuída a um único OpsItem criado com essas regras no console do Systems Manager. Para ter mais informações, consulte Editar um OpsItem.



Configuração de um administrador delegado para Explorer

Se você agregar dados do AWS Systems Manager Explorer de várias contas e Regiões da AWS usando a sincronização de dados de recursos com o AWS Organizations, é recomendável configurar um administrador delegado para o Explorer. Um administrador delegado melhora a segurança do Explorer das seguintes maneiras.

- Limite o número de administradores do Explorer que podem criar ou excluir sincronizações de dados de recurso de várias contas e de região para uma Conta da AWS individual.
- Não é mais necessário estar conectado à conta de gerenciamento do AWS Organizations para administrar as sincronizações de dados de recursos no Explorer.

Um administrador delegado pode usar as seguintes APIs de sincronização de dados de recursos do Explorer usando o console, o SDK, a AWS Command Line Interface (AWS CLI) ou o AWS Tools for Windows PowerShell:

- CreateResourceDataSync
- DeleteResourceDataSync
- ListResourceDataSync
- UpdateResourceDataSync

Um administrador delegado pode pesquisar, filtrar e agregar dados do Explorer no console ou usando ferramentas programáticas, como o SDK, a AWS CLI ou o AWS Tools for Windows PowerShell. Para pesquisar, filtrar e agregar dados, use a operação de API GetOpsSummary.

Um administrador delegado pode criar, no máximo, cinco sincronizações de dados de recursos para uma organização inteira ou um subconjunto de unidades organizacionais. As sincronizações de dados de recursos criadas por um administrador delegado só estão disponíveis na conta de administrador delegado. Não é possível visualizar as sincronizações ou os dados agregados na conta de gerenciamento do AWS Organizations.

> **Note**
>
> Não é possível usar uma conta de administrador delegado para criar uma sincronização de dados de recursos em Regiões da AWS de adesão. Você deve usar uma conta gerencial do AWS Organizations.

Para obter mais informações sobre a sincronização de dados de recursos, consulte Configurar o Systems Manager Explorer para exibir dados de várias contas e regiões. Para obter mais informações sobre o AWS Organizations, consulte O que é AWS Organizations? no Guia do Usuário doAWS Organizations.

Tópicos

- Antes de começar
- Configurar um administrador delegado do Explorer
- Cancelar o registro de um administrador delegado do Explorer

Antes de começar

A lista a seguir inclui informações importantes sobre a administração delegada do Explorer.

- É possível delegar apenas uma conta para a administração do Explorer.

- O ID da conta especificada como administrador delegado do Explorer deve ser listado como uma conta-membro no AWS Organizations. Para obter informações, consulte Criar uma Conta da AWS na sua organização no Manual do usuário do AWS Organizations.
- Um administrador delegado pode usar todas as operações da API de sincronização de dados de recursos do Explorer no console ou usando ferramentas programáticas, como o SDK, a AWS Command Line Interface (AWS CLI) ou o AWS Tools for Windows PowerShell. As operações da API de sincronização de dados de recursos incluem o seguinte: CreateResourceDataSync, DeleteResourceDataSync, ListResourceDataSync e UpdateResourceDataSync.
- Um administrador delegado pode pesquisar, filtrar e agregar dados do Explorer no console ou usando ferramentas programáticas, como o SDK, a AWS CLI ou o AWS Tools for Windows PowerShell. Para pesquisar, filtrar e agregar dados, use a operação de API GetOpsSummary.
- As sincronizações de dados de recursos criadas por um administrador delegado só estão disponíveis na conta de administrador delegado. Não é possível visualizar as sincronizações ou os dados agregados na conta de gerenciamento do AWS Organizations.
- Um administrador delegado pode criar no máximo cinco sincronizações de dados de recursos.
- Um administrador delegado pode criar uma sincronização de dados de recursos para uma organização inteira no AWS Organizations ou um subconjunto de unidades organizacionais.

Configurar um administrador delegado do Explorer

Use o procedimento a seguir para registrar um administrador delegado do Explorer.

Como registrar um administrador delegado do Explorer

1. Faça login na conta de gerenciamento do AWS Organizations.
2. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
3. No painel de navegação, escolha Explorer.
4. Escolha Settings.
5. Na seção Administrador delegado para o Explorer, verifique se você configurou a função vinculada ao serviço e as opções de acesso ao serviço necessárias. Se for necessário, selecione os botões Create role (Criar regra) e Enable access (Ativar acesso) para configurar essas opções.
6. Em Account ID (ID da conta), insira o ID da Conta da AWS. Essa conta deve ser uma conta-membro no AWS Organizations.
7. Selecione Registrar administrador delegado.

O administrador delegado agora tem acesso às opções Include all accounts from my AWS Organizations configuration (Incluir todas as contas da minha configuração do) e Select organization units in (Selecionar unidades organizacionais no AWS Organizations) na página Create resource data sync (Criar sincronização de dados dos recursos).

Cancelar o registro de um administrador delegado do Explorer

Use o procedimento a seguir para cancelar o registro de um administrador delegado do Explorer. Só é possível cancelar o registro de uma conta de administrador delegado pela conta de gerenciamento do AWS Organizations. Quando o registro de uma conta de administrador delegado é cancelado, o sistema exclui todas as sincronizações de dados de recursos do AWS Organizations criadas pelo administrador delegado.

Como cancelar o registro de um administrador delegado do Explorer

1. Faça login na conta de gerenciamento do AWS Organizations.
2. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
3. No painel de navegação, escolha Explorer.
4. Escolha Settings.
5. Na seção Administrador delegado do Explorer, selecione Cancelar registro. O sistema exibe um aviso.
6. Insira o ID da conta e selecione Remove (Remover).

A conta não tem mais acesso às operações da API de sincronização de dados de recursos do AWS Organizations. O sistema exclui todas as sincronizações de dados de recursos do AWS Organizations criadas pela conta.

Configurar o Systems Manager Explorer para exibir dados de várias contas e regiões

O AWS Systems Manager usa uma experiência de configuração integrada para ajudar você a começar a usar o AWS Systems Manager Explorer e o AWS Systems Manager OpsCenter. Depois de concluir a Configuração Integrada, o Explorer e o OpsCenter sincronizam os dados automaticamente. Mais especificamente, essas ferramentas sincronizam o OpsData e o OpsItems para a Conta da AWS e a Região da AWS que você usou quando concluiu a Configuração integrada. Para agregar o OpsData e o OpsItems em outras contas e regiões, crie uma sincronização de dados de recursos, conforme descrito neste tópico.

> **Note**
>
> Para obter mais informações sobre a configuração integrada, consulte Conceitos básicos do Systems Manager Explorer e do OpsCenter.

Tópicos

- Noções básicas da sincronização de recursos para o Explorer
- Noções básicas sobre sincronizações de dados de recursos de várias contas e regiões
- Criar uma sincronização de dados de recursos

Noções básicas da sincronização de recursos para o Explorer

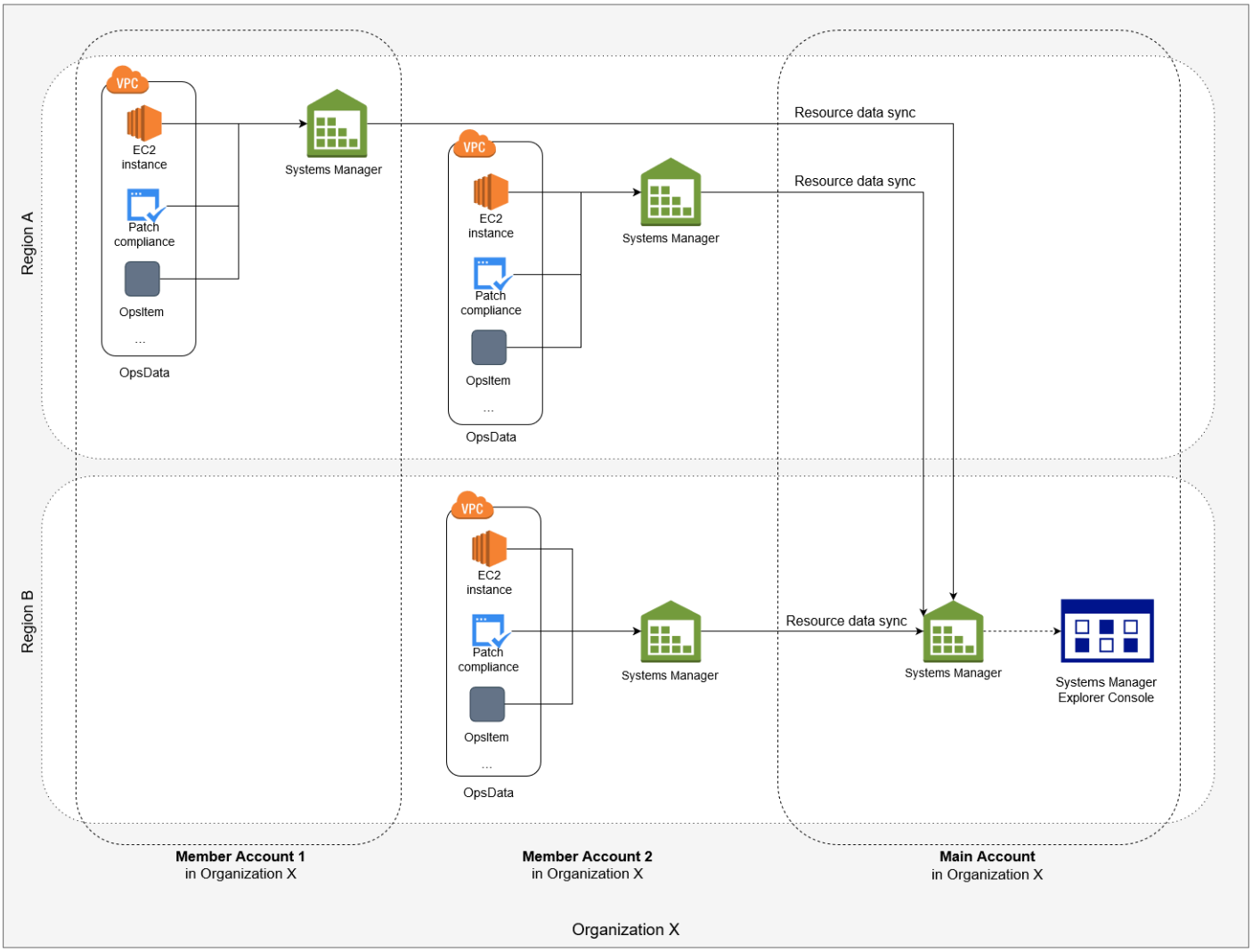
A sincronização de dados de recursos do Explorer oferece duas opções de agregação:

- Conta única/várias regiões: você pode configurar o Explorer para agregar OpsItems e OpsData de várias Regiões da AWS, mas o conjunto de dados é limitado à Conta da AWS atual.
- Várias contas/várias regiões: você pode configurar o Explorer para agregar dados de várias regiões e contas da Regiões da AWS. Essa opção requer a instalação e a configuração do AWS Organizations. Se você instalar e configurar o AWS Organizations, poderá agregar dados no Explorer por unidade organizacional (UO) ou para uma organização inteira. O Systems Manager agrega os dados na conta de gerenciamento do AWS Organizations antes de exibi-los em Explorer. Para obter mais informações, consulte O que é o AWS Organizations? no Guia do usuário do AWS Organizations.

> **Warning**
>
> Se você configurar Explorer para agregar dados de uma organização emAWS Organizations, o sistema habilita o OpsData em todas as contas-membro da organização. Habilitar fontes OpsData em todas as contas de membros aumenta o número de chamadas para OpsCenter APIs como CreateOpsItem e GetOpsSummary. Você é cobrado pelas chamadas para essas ações da API.

O diagrama a seguir mostra uma sincronização de dados de recursos configurada para trabalhar com o AWS Organizations. Nesse cenário, o usuário tem duas contas definidas no AWS

Organizations. A sincronização de dados de recursos agrega dados de ambas as contas e de várias Regiões da AWS na conta de gerenciamento do AWS Organizations onde são então exibidos no Explorer.



Noções básicas sobre sincronizações de dados de recursos de várias contas e regiões

Esta seção descreve detalhes importantes sobre as sincronizações de dados de recursos de várias contas e regiões que usam o AWS Organizations. As informações nesta seção se aplicam especificamente se você escolher uma das seguintes opções na página Create resource data sync (Criar sincronização de dados de recursos).

- Incluir todas as contas das minhas configurações do AWS Organizations
- Selecionar unidades organizacionais em AWS Organizations

Se você não planeja usar uma dessas opções, ignore esta seção.

Ao criar uma sincronização de dados do recurso no console do SSM, se você escolher uma das opções do AWS Organizations, o Systems Manager permitirá automaticamente todas as fontes do OpsData em todas as regiões selecionadas para todas as Contas da AWS de sua organização (ou das unidades organizacionais selecionadas). Por exemplo, mesmo que você não tenha ativado o Explorer em uma região, se você selecionar uma opção do AWS Organizations para a sincronização de dados de recursos, o Systems Manager coletará automaticamente o OpsData dessa região. Para criar uma sincronização de dados de recursos sem permitir fontes de OpsData, especifique EnableAllOpsDataSources como false ao criar a sincronização de dados. Para obter mais informações, consulte os detalhes do parâmetro `EnableAllOpsDataSources` do tipo de dados ResourceDataSyncSource na Referência da API do Amazon EC2 Systems Manager.

Se você não selecionar uma das opções do AWS Organizations para uma sincronização de dados de recurso, você deverá concluir a configuração integrada em cada conta e região nas quais quiser que o Explorer acesse os dados. Caso contrário, o Explorer não exibirá OpsData e OpsItems para as contas e regiões nas quais você não concluiu a configuração integrada.

Se você adicionar uma conta filho à sua organização, o Explorer permitirá automaticamente todas as origens do OpsData para a conta. Se, posteriormente, você remover a conta de criança da sua organização, o Explorer continuará a coletar o OpsData da conta.

Se você atualizar uma sincronização de dados de recursos existentes que usa uma das opções do AWS Organizations, o sistema solicitará que você aprove a coleta de todas as origens do OpsData para todas as contas e regiões afetadas pela alteração.

Se você adicionar um novo serviço à sua Conta da AWS, e se o Explorer coleta o OpsData para esse serviço, o Systems Manager configura automaticamente o Explorer para coletar esse OpsData. Por exemplo, se sua organização não usou o AWS Trusted Advisor anteriormente quando você criou uma sincronização de dados de recursos, mas está inscrita nesse serviço, o Explorer atualiza automaticamente suas sincronizações de dados de recursos para coletar o OpsData.

Important

Observe as seguintes informações importantes sobre as sincronizações de dados de recursos de várias contas e regiões:

- A exclusão de uma sincronização de dados do recurso não desativa uma fonte OpsData no Explorer.

- Para visualizar o OpsData e o OpsItems de várias contas, você deve ter o modo All features (Todos os recursos) do AWS Organizations ativado e estar conectado à conta de gerenciamento do AWS Organizations.
- A maioria das Regiões da AWS está ativa por padrão para a Conta da AWS, mas algumas regiões são ativadas somente quando você as seleciona manualmente. Essas regiões são conhecidas como regiões de adesão. Por padrão, as sincronizações de dados de recursos do Explorer entre contas e regiões não oferecem suporte à agregação de dados em regiões de adesão. Suporte foi adicionado às seguintes regiões de adesão em 30 de junho de 2025.
  - Europa (Milão)
  - África (Cidade do Cabo)
  - Oriente Médio (Bahrein)
  - Ásia-Pacífico (Hong Kong)

  Observe que não é possível usar uma conta de administrador delegado para criar uma sincronização de dados de recursos em regiões de adesão. Você deve usar uma conta gerencial do AWS Organizations.

Criar uma sincronização de dados de recursos

Antes de configurar uma sincronização de dados de recursos para o Explorer, observe os detalhes a seguir.

- O Explorer permite no máximo cinco sincronizações de dados de recursos.
- Depois de criar uma sincronização de dados de recurso para uma região, não é possível alterar as opções de conta para essa sincronização. Por exemplo, se você criar uma sincronização na região us-east-2 (Ohio) e escolher a opção Include only the current account (Incluir somente a conta atual), você não poderá editar essa sincronização posteriormente e escolher a opção Include all accounts from my AWS Organizations configuration (Incluir todas as contas da minha configuração do). Em vez disso, você deve excluir a primeira sincronização de dados de recurso e criar uma nova.
- OpsData visualizado no Explorer é somente leitura.

Use o procedimento a seguir para criar uma sincronização de dados de recurso para o Explorer.

Para criar uma sincronização de dados de recurso

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Explorer.
3. Escolha Settings.
4. Na seção Configure resource data sync (Configurar sincronização de dados de recurso), escolha Create resource data sync (Criar sincronização de dados de recurso).
5. Em Resource data sync name (Nome de sincronização de dados do recurso), insira um nome.
6. Na seção Add accounts (Adicionar contas), escolha uma opção.

> **Note**
>
> Para usar qualquer uma das opções do AWS Organizations, é necessário estar conectado à conta de gerenciamento do AWS Organizations ou estar conectado a uma conta de administrador delegado do Explorer. Para obter mais informações sobre a conta de administrador delegado, consulte Configuração de um administrador delegado para Explorer.

7. Na seção Regions to include (Regiões a incluir), escolha uma das seguintes opções.
   - Escolha All current and future regions (Todas as regiões atuais e futuras) para sincronizar automaticamente os dados de todas as Regiões da AWS atuais e de quaisquer novas regiões que fiquem online no futuro.
   - Escolha All regions (Todas as regiões) para sincronizar automaticamente os dados de todas as Regiões da AWS atuais.
   - Escolha individualmente as regiões que você deseja incluir.
8. Escolha Create resource data sync (Criar sincronização de dados de recurso).

O sistema pode levar vários minutos para preencher o Explorer com dados depois de você criar uma sincronização de dados de recurso. Você pode exibir a sincronização selecionando-a na lista Select a resource data sync (Selecionar uma sincronização de dados de recurso) no Explorer.

Usar o Explorer

Esta seção inclui informações sobre como personalizar o AWS Systems Manager Explorer, alterando o layout do widget e os dados exibidos no painel.

Conteúdo

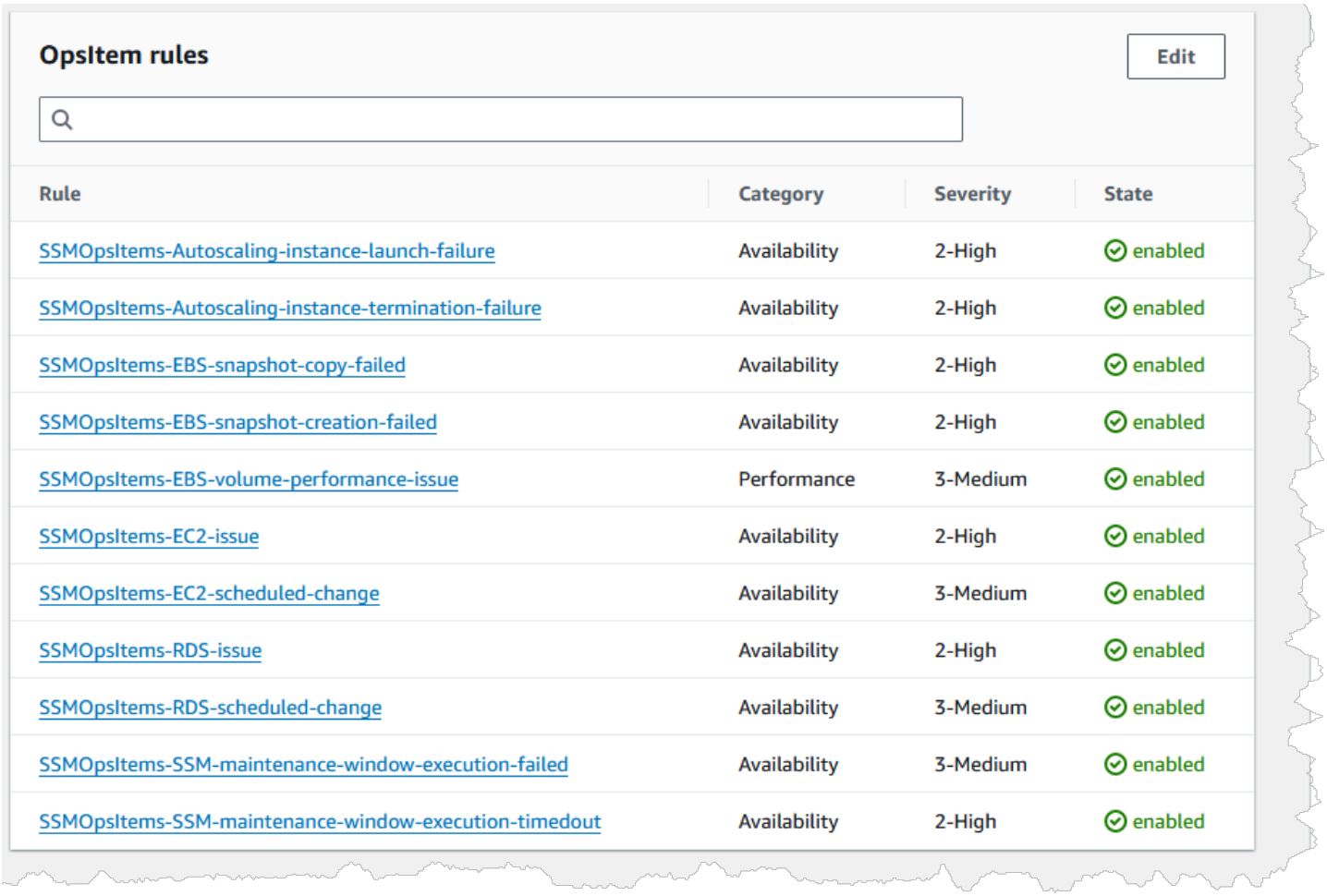
- Editar regras do EventBridge criadas para o Explorer
- Edite as fontes de dados do Systems Manager Explorer
- Personalizar a exibição do Explorer
- Receber descobertas do AWS Security Hub no Explorer
- Excluir a sincronização de dados de recursos do Systems Manager Explorer

Editar regras do EventBridge criadas para o Explorer

Quando você concluir a configuração integrada, o sistema permitirá mais de uma dúzia de regras no Amazon EventBridge. Essas regras criam automaticamente OpsItems no AWS Systems Manager OpsCenter. O AWS Systems Manager Explorer exibe informações agregadas sobre OpsItems.

Cada regra inclui um valor predefinido de Category (Categoria) e Severity (Gravidade). Quando o sistema cria OpsItems a partir de um evento, ele atribui automaticamente os valores predefinidos de Category (Categoria) e Severity (Gravidade).

> **Important**
>
> Você não pode editar os valores Category (Categoria) e Severity (Gravidade) para regras padrão, mas pode editar esses valores no OpsItems criado com as regras padrão.



Como editar regras padrão para criar OpsItems

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Explorer.
3. Escolha Configurações.
4. Na seção OpsItems rules (Regras do OpsItems), selecione Edit (Editar).
5. Expanda CWE rules (Regras do CWE).
6. Desmarque a caixa de seleção ao lado das regras que você não deseja usar.
7. Use as listas Category (Categoria) e Severity (Gravidade) para alterar essas informações de uma regra.
8. Escolha Salvar.

Suas alterações entrarão em vigor na próxima vez que o sistema criar um OpsItem.

Edite as fontes de dados do Systems Manager Explorer

Para Regiões da AWS que estão disponíveis por padrão, o AWS Systems Manager Explorer exibe dados das fontes a seguir. Você pode editar as configurações do Explorer para adicionar ou remover fontes de dados:

- Amazon Elastic Compute Cloud (Amazon EC2)
- Inventário do AWS Systems Manager
- AWS Systems Manager OpsCenter OpsItems
- Conformidade de patches do AWS Systems Manager Patch Manager
- Conformidade das associações do AWS Systems Manager State Manager
- AWS Trusted Advisor
- AWS Compute Optimizer
- Casos do AWS Support Center
- Conformidade de regras e recursos do AWS Config
- Descobertas do AWS Security Hub

> **Note**
>
> Para a região Ásia-Pacífico (Osaka), o Explorer não exibe dados de descobertas do AWS Compute Optimizer e AWS Security Hub.

Para as Regiões da AWS de aceitação, o Explorer exibe dados das seguintes fontes:

- Amazon Elastic Compute Cloud (Amazon EC2)
- Inventário do AWS Systems Manager
- AWS Systems Manager OpsCenter OpsItems
- Conformidade de patches do AWS Systems Manager Patch Manager
- Conformidade das associações do AWS Systems Manager State Manager
- AWS Trusted Advisor
- Casos do AWS Support Center
- Conformidade de regras e recursos do AWS Config

> **Note**
>
> - Para visualizar casos do Suporte Center no Explorer, você deve ter uma conta Enterprise ou Business configurada com o Suporte Support.
> - Não é possível configurar o Explorer para parar de exibir dados de OpsItem do OpsCenter.

Antes de começar

Verifique se você configurou os serviços que preenchem os widgets do Explorer com dados. Para obter mais informações, consulte Configurar serviços relacionados para o Explorer.

Como editar fontes de dados

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Explorer.
3. Escolha Configurações e, em seguida, escolha a guia Configurar painel.
4. Na seção Fontes de OpsData, na coluna Status, ative ou desative as fontes de acordo com os dados que você deseja visualizar.

Personalizar a exibição do Explorer

Você pode personalizar o layout do widget no AWS Systems Manager Explorer usando um recurso de arrastar e soltar. Você também pode personalizar os OpsData e OpsItems exibidos no Explorer usando filtros, conforme descrito neste tópico.

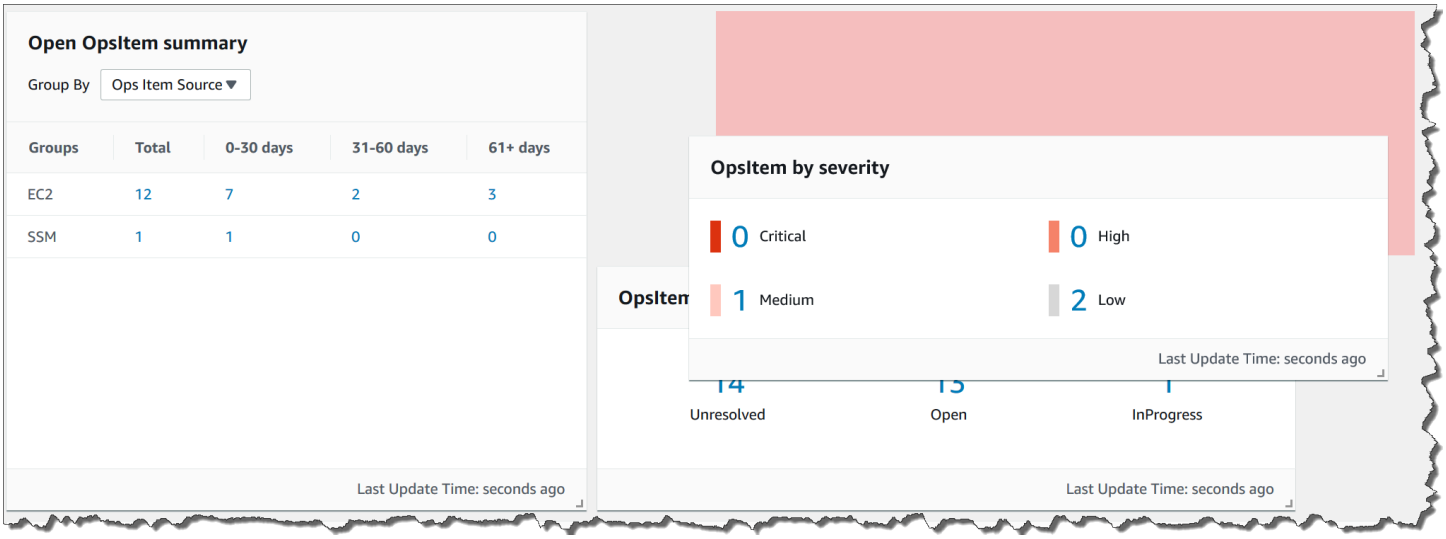
Antes de personalizar o layout do widget, verifique se os widgets que você deseja visualizar estão atualmente exibidos no Explorer. Para visualizar alguns widgets no Explorer (como o widget de conformidade do AWS Config), você deve habilitá-los na página Configure dashboard (Configurar painel).

Para permitir que os widgets sejam exibidos no Explorer

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Explorer.
3. Escolha Dashboard actions (Ações do painel), Configure dashboard (Configurar painel).
4. Escolha a guia Configure Dashboard (Configurar painel).

5. Escolha Enable all (Habilitar tudo) ou ative um widget ou uma fonte de dados individual.
6. Escolha Explorer para visualizar suas alterações.

Para personalizar o layout do widget no Explorer, escolha um widget que você deseja mover. Clique e mantenha pressionado o nome do widget e arraste-o até o novo local.



Repita este processo para cada widget que você deseja reposicionar.

Se você decidir que não gosta do novo layout, escolha Reset layout (Redefinir layout) para mover todos os widgets de volta para o local original.

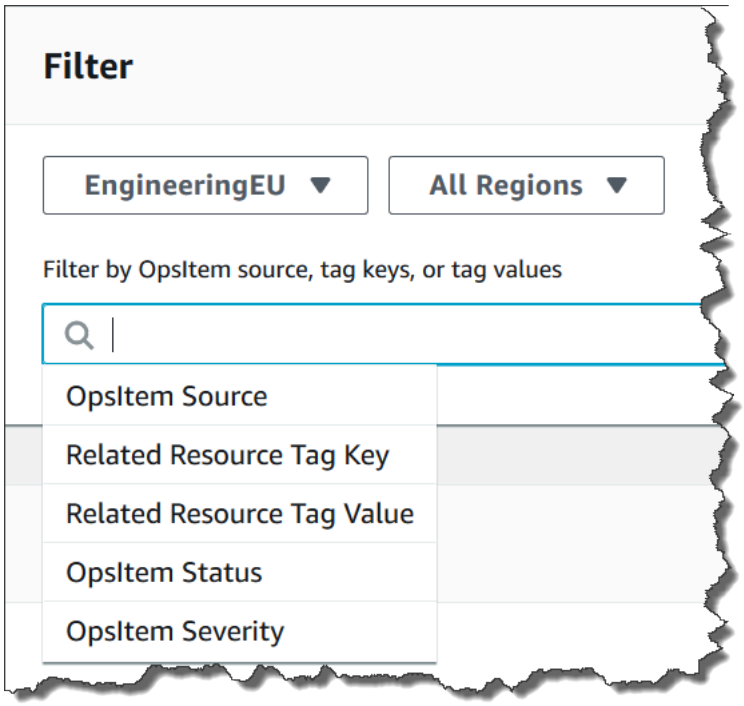
Usar filtros para alterar os dados exibidos no Explorer

Por padrão, o Explorer exibe dados da Conta da AWS e da região atual. Se você criar uma ou mais sincronizações de dados de recurso, poderá usar filtros para alterar qual sincronização está ativa. Depois, você pode optar por exibir dados para uma região específica ou para todas as regiões. Você também pode usar a barra de pesquisa para filtrar diferentes critérios de OpsItem e chave-tag.

Como alterar os dados exibidos no Explorer usando filtros

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Explorer.
3. Na seção Filter (Filtro), use a lista Select a resource data sync (Selecionar uma sincronização de dados de recurso) para escolher uma sincronização.
4. Use a lista Regions (Regiões) para escolher uma região específica da Região da AWS ou escolha All Regions (Todas as regiões).

5. Escolha a barra de pesquisa e os critérios para filtrar os dados.



6. Pressione Enter.

O Explorer manterá as opções de filtro selecionadas se você fechar e reabrir a página.

Receber descobertas do AWS Security Hub no Explorer

O AWS Security Hub fornece uma visualização abrangente de seu estado de segurança na AWS. O serviço coleta dados de segurança, chamados de descobertas, de várias Contas da AWS, serviços e produtos de terceiros compatíveis. As descobertas do Security Hub podem ajudar a verificar o ambiente conforme os padrões e as práticas recomendadas do setor de segurança, a analisar as tendências de segurança e a identificar os problemas de segurança de maior prioridade.

O Security Hub envia descobertas ao Amazon EventBridge, que usa uma regra de evento para enviar as descobertas ao Explorer. Após habilitar a integração, conforme descrito aqui, você poderá ver as descobertas do Security Hub em um widget do Explorer e ver os detalhes da descoberta em OpsCenter OpsItems. O widget fornece um resumo de todas as descobertas do Security Hub com base na gravidade. Novas descobertas no Security Hub geralmente são visíveis no Explorer segundos após serem criadas.

Warning

Observe as seguintes informações importantes:

- O Explorer é integrado ao OpsCenter, uma ferramenta do Systems Manager. Depois que você habilitar a integração do Explorer com o Security Hub, o OpsCenter criará automaticamente OpsItems para as descobertas do Security Hub. Dependendo do ambiente da AWS, habilitar a integração poderá resultar em um grande número de OpsItems, que incorrerá em custos.

  Antes de continuar, leia sobre a integração do OpsCenter com o Security Hub. O tópico contém detalhes específicos sobre como as alterações e atualizações nas descobertas e nos OpsItems são cobradas em sua conta. Para ter mais informações, consulte Noções básicas sobre a integração do OpsCenter com o AWS Security Hub. Para obter informações sobre preços do OpsCenter, consulte Preços do AWS Systems Manager.
- Se você criar uma sincronização de dados de recursos no Explorer enquanto estiver conectado à conta do administrador, a integração com o Security Hub será habilitada automaticamente para o administrador e para todas as contas de membro na sincronização. Depois de habilitado, o OpsCenter cria automaticamente OpsItems para descobertas do Security Hub, incorrendo em custos. Para obter mais informações sobre como criar uma sincronização de dados de recursos, consulte Configurar o Systems Manager Explorer para exibir dados de várias contas e regiões.

Tipos de descobertas que o Explorer recebe

O Explorer recebe todas as descobertas Security Hub. Você pode ver todas as descobertas com base na gravidade no widget do Explorer quando ativar as configurações padrão do Security Hub. Por padrão, o Explorer cria OpsItems para descobertas criticas e de alta gravidade. Você pode configurar o Explorer manualmente para criar OpsItems para descobertas de gravidade média e baixa.

Embora o Explorer não crie OpsItems para descobertas informativas, é possível visualizar dados de operações informativas (OpsData) no widget de resumo das descobertas do Security Hub. O Explorer cria OpsData para todas as descobertas, de qualquer gravidade. Para obter mais informações sobre níveis de gravidade do Security Hub, consulte Severity na AWS Security Hub API Reference.

Habilitar a integração

Esta seção descreve como habilitar configurar o Explorer para começar a receber descobertas do Security Hub.

Antes de começar

Conclua as seguintes tarefas antes de configurar o Explorer para começar a receber descobertas do Security Hub.

- Habilite e configure o Security Hub. Para obter mais informações , consulte Configurar o Security Hub no Manual do usuário do AWS Security Hub.
- Faça login na conta de gerenciamento do AWS Organizations. O Systems Manager requer acesso ao AWS Organizations para criar OpsItems das descobertas do Security Hub. Depois de se registrar na conta de gerenciamento, você deverá selecionar o botão Enable access (Habilitar acesso) na guia Explorer Configure dashboard (Configurar painel do Explorer), conforme descrito no procedimento a seguir. Se você não fizer login na conta de gerenciamento do AWS Organizations, você não poderá permitir acesso e o Explorer não poderá criar OpsItems com base nas descobertas do Security Hub.

Para começar a receber as descobertas do Security Hub

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Explorer.
3. Selecione Settings.
4. Selecione a guia Configure dashboard (Configurar painel).
5. Selecione AWS Security Hub.
6. Selecione o controle deslizante Disabled (Desabilitado) para ativar o AWS Security Hub.

   As descobertas críticas e de alta gravidade são exibidas por padrão. Para exibir descobertas de gravidade média e baixa, selecione o botão deslizante Desabilitado, ao lado de Média, Baixa.
7. Na seção OpsItems created by Security Hub findings (OpsItems criados pelas descobertas do Security Hub), selecione Enable access (Habilitar acesso). Se esse botão não for exibido, faça login na conta de gerenciamento do AWS Organizations e retorne a esta página para selecionar o botão.

Como examinar as descobertas do Security Hub.

O procedimento a seguir descreve como visualizar as descobertas do Security Hub.

Para visualizar descobertas do Security Hub

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Explorer.
3. Localize o widget de resumo das descobertas do AWS Security Hub. Isso exibe suas descobertas do Security Hub. Você pode selecionar um nível de gravidade para exibir uma descrição detalhada do OpsItem correspondente.

Como parar de receber descobertas

O procedimento a seguir descreve como parar de receber descobertas do Security Hub.

Para interromper o recebimento das descobertas do Security Hub

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Explorer.
3. Selecione Settings.
4. Selecione a guia Configure dashboard (Configurar painel).
5. Selecione o controle deslizante Enabled (Habilitado) para desativar o AWS Security Hub.

> **Important**
>
> Se a opção de desabilitar as descobertas do Security Hub estiver esmaecida no console, você poderá desabilitar essa configuração executando o seguinte comando na AWS CLI. O comando deverá ser executado enquanto você estiver conectado à conta de gerenciamento do AWS Organizations ou à conta de administrador delegado do Systems Manager. Para o parâmetro `region`, especifique a Região da AWS na qual você deseja parar de receber as descobertas do Security Hub no Explorer.
>
> ```
> aws ssm update-service-setting --setting-id /ssm/opsdata/SecurityHub --setting-value Disabled --region Região da AWS
> ```

Aqui está um exemplo.

```
aws ssm update-service-setting --setting-id /ssm/opsdata/SecurityHub --setting-
value Disabled --region us-east-1
```

Excluir a sincronização de dados de recursos do Systems Manager Explorer

No AWS Systems Manager Explorer, você pode agregar OpsData e OpsItems de outras contas e regiões criando uma sincronização de dados de recurso.

Não é possível alterar as opções de conta para uma sincronização de dados de recurso. Por exemplo, se você criou uma sincronização na região us-east-2 (Ohio) e escolheu a opção Include only the current account (Incluir somente a conta atual), você não poderá editar essa sincronização posteriormente e escolher a opção Include all accounts from my AWS Organizations configuration (Incluir todas as contas da minha configuração do). Em vez disso, você deve excluir a sincronização de dados de recurso e criar uma nova, conforme descrito no procedimento a seguir.

Como excluir uma sincronização de dados de recurso

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Explorer.
3. Escolha Configurações.
4. Na seção Configure resource data sync (Configurar sincronização de dados de recurso), escolha a sincronização de dados de recurso que deseja excluir.
5. Escolha Excluir.

Exportar OpsData do Systems Manager Explorer

É possível exportar cinco mil relatórios de OpsData, como um arquivo de valores separados por vírgula (.csv), para um bucket do Amazon Simple Storage Service (Amazon S3) pelo Explorer do AWS Systems Manager. O Explorer usa o runbook de automação AWS-ExportOpsDataToS3 para exportar OpsData. Quando você exporta OpsData, o sistema exibe a página do runbook de automação, na qual é possível especificar detalhes, como assumeRole, nome do bucket do Amazon S3, ARN do tópico do SNS e campos a serem exportados.

Para exportar OpsData:

- Etapa 1: especificar um tópico do SNS
- Etapa 2: (opcional) configurar a exportação de dados
- Etapa 3: exportar OpsData

Etapa 1: especificar um tópico do SNS

Ao configurar a exportação de dados, você deverá especificar um tópico do Amazon Simple Notification Service (Amazon SNS) que existe na mesma Região da AWS para onde você deseja exportar os dados. O Systems Manager envia uma notificação para o tópico do Amazon SNS quando uma exportação é concluída. Para obter informações sobre como criar um tópico do Amazon SNS, consulte Criar um tópico do Amazon SNS.

Etapa 2: (opcional) configurar a exportação de dados

Você pode definir as configurações de exportação de dados na página Configurações ou Exportar OpsData para o bucket do S3.

Para configurar a exportação de dados do Explorer

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Explorer.
3. Escolha Settings.
4. Na seção Configure data export (Configurar exportação de dados), escolha Edit (Editar).
5. Para carregar o arquivo de exportação de dados para um bucket do Amazon S3, escolha Selecionar um bucket existente do S3 e escolha o bucket na lista.

   Para carregar o arquivo de exportação de dados para um novo bucket do Amazon S3, escolha Criar um bucket do S3 e insira o nome que deseja usar para o novo bucket.

> **Note**
>
> É possível editar o nome do bucket do Amazon S3 e o ARN do tópico do Amazon SNS somente na página em que você definiu essas configurações pela primeira vez no Explorer. Se você configurar o bucket do Amazon S3 e o ARN do tópico do Amazon SNS na página Configurações, só poderá modificar essas configurações na página Configurações.

6. Em Selecionar um ARN de tópico do Amazon SNS, escolha o tópico que você quer notificar quando a exportação for concluída.
7. Escolha Criar.

Etapa 3: exportar OpsData

Ao exportar dados do Explorer, o Systems Manager cria um perfil do AWS Identity and Access Management (IAM) chamado `AmazonSSMExplorerExportRole`. Essa função usa a seguinte política do IAM:

JSON

```
{
      "Version": "2012-10-17",
      "Statement": [
          {
              "Sid": "OpsSummaryExportAutomationServiceRoleStatement1",
              "Effect": "Allow",
              "Action": [
                  "s3:PutObject"
              ],
              "Resource": [
                 "arn:aws:s3:::{{ExportDestinationS3BucketName}}/*"
              ]
          },
          {
              "Sid": "OpsSummaryExportAutomationServiceRoleStatement2",
              "Effect": "Allow",
              "Action": [
                  "s3:GetBucketAcl",
                  "s3:GetBucketLocation"
              ],
              "Resource": [
                 "arn:aws:s3:::{{ExportDestinationS3BucketName}}"
              ]
          },
          {
              "Sid": "OpsSummaryExportAutomationServiceRoleStatement3",
              "Effect": "Allow",
              "Action": [
                  "sns:Publish"
```

```
            ],
            "Resource": [
                "arn:aws:sns:*:*:{{SNSTopicName}}"
            ]
        },
        {
            "Sid": "OpsSummaryExportAutomationServiceRoleStatement4",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "OpsSummaryExportAutomationServiceRoleStatement5",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:PutLogEvents",
                "logs:CreateLogStream"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "OpsSummaryExportAutomationServiceRoleStatement6",
            "Effect": "Allow",
            "Action": [
                "ssm:GetOpsSummary"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

A função inclui a seguinte entidade de confiança.

JSON

```
{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Sid": "OpsSummaryExportAutomationServiceRoleTrustPolicy",
          "Effect": "Allow",
          "Principal": {
            "Service": "ssm.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }
```

Como exportar OpsData do Explorer

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Explorer.
3. Escolha um link de dados em um widget do Explorer. Por exemplo, escolha Não resolvidos ou Problemas abertos no widget OpsItems por status. Ou escolha o link de dados Crítica ou Alta no widget OpsItem por severidade. O Explorer abre a página OpsData para o conjunto de dados selecionado.
4. Escolha Exportar tabela.

   > **Note**
   >
   > Quando você exporta OpsData pela primeira vez, o sistema cria um perfil de admissão para a exportação. Não é possível modificar o perfil de admissão padrão.

5. Em Nome do bucket do S3, escolha um bucket existente. É possível escolher Criar para criar um bucket do Amazon S3, se necessário.

   Caso não consiga alterar o nome do bucket do S3, significa que você configurou o nome do bucket na página Configurações. Só é possível alterar o nome do bucket na página Configurações.

> **Note**
>
> É possível editar o nome do bucket do Amazon S3 e o ARN do tópico do Amazon SNS somente na página em que você definiu essas configurações pela primeira vez no Explorer.

6. Em ARN do tópico do SNS, escolha um ARN de tópico atual do Amazon SNS para notificar quando o download for concluído.

   Se não for possível alterar o ARN do tópico do Amazon SNS, significa que você configurou o ARN do tópico do Amazon SNS na página Configurações. Só é possível alterar o ARN do tópico na página Configurações.
7. (Opcional) Em Mensagem de sucesso do SNS, especifique uma mensagem de sucesso que você deseja exibir quando a exportação for concluída com êxito.
8. Selecione Enviar. O sistema navega até a página anterior e exibe a mensagem Clique para visualizar o status do processo de exportação. Visualizar detalhes.

   Você pode escolher Visualizar detalhes para ver o status do runbook e o progresso no Systems Manager Automation.

Agora você pode exportar OpsData do Explorer para o bucket do Amazon S3 especificado.

Se você não puder exportar dados usando este procedimento, verifique se seu usuário, grupo ou perfil inclui as ações `iam:CreatePolicyVersion` e `iam:DeletePolicyVersion`. Para obter informações sobre como adicionar essas ações ao seu usuário, grupo ou perfil, consulte Edição de políticas do IAM no Guia do usuário do IAM.

Solução de problemas do Systems Manager Explorer

Este tópico inclui informações sobre como resolver problemas comuns com o AWS Systems Manager Explorer.

A sincronização de dados de recursos parou de funcionar

Se sua sincronização de dados de recursos parou de funcionar para uma Região da AWS de adesão, verifique se houve a adesão da região tanto para a conta gerencial do AWS Organizations quanto para a conta de membro para a qual a sincronização está configurada.

A maioria das Regiões da AWS está ativa por padrão para a Conta da AWS, mas algumas regiões são ativadas somente quando você as seleciona manualmente. Essas regiões são conhecidas como regiões de adesão. Por padrão, as sincronizações de dados de recursos do Explorer entre contas e regiões não oferecem suporte à agregação de dados em regiões de adesão. Suporte foi adicionado às seguintes regiões de adesão em 30 de junho de 2025.

- Europa (Milão)
- África (Cidade do Cabo)
- Oriente Médio (Bahrein)
- Ásia-Pacífico (Hong Kong)

Observe que não é possível usar uma conta de administrador delegado para criar uma sincronização de dados de recursos em regiões de adesão. Você deve usar uma conta gerencial do AWS Organizations.

Não é possível filtrar recursos da AWS no Explorer depois de atualizar as etiquetas na página Settings (Configurações)

Se você atualizar chaves de etiquetas ou outras configurações de dados no Explorer, o sistema poderá levar até seis horas para sincronizar dados com base em suas alterações.

As opções do AWS Organizations na página Criar sincronização de dados de recursos estão desabilitadas

As opções Include all accounts from my AWS Organizations configuration (Incluir todas as contas da minha configuração) e Select organization units in AWS Organizations (Selecionar unidades organizacionais na) página Create resource data sync (Criar sincronização de dados de recursos) só estarão disponíveis se você configurar e configurar AWS Organizations. Se você configurar o AWS Organizations, a conta de gerenciamento do AWS Organizations ou um administrador delegado do Explorer, poderá criar sincronizações de dados de recursos que usam essas opções.

Para obter mais informações, consulte Configurar o Systems Manager Explorer para exibir dados de várias contas e regiões e Configuração de um administrador delegado para Explorer.

O Explorer não exibe nenhum dado

- Verifique se você concluiu a configuração integrada em cada conta e região onde o Explorer deseja acessar e exibir dados. Caso contrário, o Explorer não exibirá OpsData e OpsItems para

as contas e regiões nas quais você não concluiu a configuração integrada. Para obter mais informações, consulte Conceitos básicos do Systems Manager Explorer e do OpsCenter.

- Ao usar o Explorer para exibir dados de várias contas e regiões, verifique se você está conectado à conta de gerenciamento do AWS Organizations ou a conta do administrador delegado do Explorer. Para exibir OpsData e OpsItems de várias contas e regiões, você deve estar conectado a essa conta.

Widgets sobre instâncias do Amazon EC2 não exibem dados

Se os widgets sobre instâncias do Amazon Elastic Compute Cloud (Amazon EC2), como Instance count (Contagem de instâncias), Managed instances (Instâncias gerenciadas) e Instance by AMI (Instância por AMI), não exibirem dados, verifique o seguinte:

- Verifique se você esperou vários minutos. O OpsData pode levar vários minutos para ser exibido no Explorer após a conclusão da configuração integrada.
- Verifique se você configurou o gravador de configuração do AWS Config. O Explorer usa dados fornecidos pelo gravador de configuração do AWS Config para preencher widgets com informações sobre as instâncias do EC2. Para obter mais informações, consulte Gerenciar o gravador de configuração.
- Verifique se a fonte de OpsData do Amazon EC2 está ativa na página Settings (Configurações). Além disso, verifique se mais de 6 horas se passaram desde que você ativou o gravador de configuração ou desde que fez alterações nas instâncias. O Systems Manager pode levar até seis horas para exibir dados do AWS Config nos widgets do Explorer EC2, depois que você ativar inicialmente o gravador de configurações ou fazer alterações em suas instâncias.
- Lembre-se de que, se uma instância for interrompida ou encerrada, o Explorer deixará de exibir essas instâncias após 24 horas.
- Verifique se você está na Região da AWS correta, na qual você configurou as instâncias do Amazon EC2. O Explorer não exibe dados sobre instâncias on-premises.
- Se você configurou uma sincronização de dados de recurso para várias contas e regiões, verifique se está conectado à conta de gerenciamento do Organizations ou à conta do administrador delegado do Explorer.

O widget de patch não exibe dados

O widget Non-compliant instances for patching (Instâncias não compatíveis para a aplicação de patches) exibe apenas os dados sobre instâncias de patch que não são compatíveis. Esse widget

não exibirá dados se suas instâncias forem compatíveis. Se você suspeitar que tem instâncias não compatíveis, verifique se instalou e configurou patches do Systems Manager e use o AWS Systems Manager Patch Manager para verificar a conformidade do patch. Para obter mais informações, consulte AWS Systems Manager Patch Manager.

Questões diversas

O Explorer não permite que você edite ou corrija OpsItems: OpsItems visualizados em contas ou regiões são somente leitura. Eles só podem ser atualizados e corrigidos de sua conta ou região inicial.

AWS Systems Manager OpsCenter

O OpsCenter, uma ferramenta do AWS Systems Manager, fornece um local central em que engenheiros de operações e profissionais de TI podem visualizar, investigar e resolver itens de trabalho operacionais (OpsItems) relacionados aos recursos da AWS. O OpsCenter é projetado para reduzir o tempo médio até a resolução de problemas que impactam os recursos da AWS. O OpsCenter agrega e padroniza OpsItems em todos os serviços, fornecendo dados de investigação contextual sobre cadaOpsItem, OpsItems relacionados, entre outros recursos associados. O OpsCenter também fornece runbooks de automação do Systems Manager, que você pode usar facilmente para solucionar problemas. Você pode especificar dados personalizados e pesquisáveis para cada OpsItem. Você também pode visualizar relatórios de resumo gerados automaticamente sobre o OpsItems por status e origem. Para começar a usar o OpsCenter, abra o Systems Manager console (Console do gerenciador de sistemas). No painel de navegação, escolha OpsCenter.

O OpsCenter é integrado ao Amazon EventBridge e ao Amazon CloudWatch. Isso significa que você pode configurar esses serviços para criar automaticamente um OpsItem no OpsCenter quando um alarme do CloudWatch entrar no estado `ALARM` ou quando o EventBridge processar um evento de qualquer serviço da AWS que publique eventos. Configurar alarmes do CloudWatch e eventos do EventBridge para criar OpsItems automaticamente permite diagnosticar e corrigir rapidamente problemas com os recursos da AWS em um único console.

Para ajudar você a diagnosticar problemas, cada OpsItem inclui informações contextualmente relevantes como o nome e o ID do recurso da AWS que gerou o OpsItem, detalhes do alarme ou do evento, histórico de alarmes e um gráfico da linha do tempo do alarme.

Para o recurso da AWS, o OpsCenter agrega informações de AWS Config de logs do AWS CloudTrail e do Amazon CloudWatch Events, para que você não precise navegar em várias páginas do console durante sua investigação.

A lista a seguir inclui tipos de recursos e métricas da AWS para os quais os clientes configuram os alarmes do CloudWatch que criam OpsItems.

- Amazon DynamoDB: ações de leitura e gravação de banco de dados atingem um limite
- Amazon EC2: a utilização da CPU atinge um limite
- Faturamento da AWS: cobranças estimadas atingem um limite
- Amazon EC2: falha na verificação de status de uma instância
- Amazon Elastic Block Store (EBS): a utilização do espaço em disco atinge um limite

A lista a seguir inclui tipos de regras do EventBridge configuradas pelo cliente para criar OpsItems.

- AWS Security Hub: alerta de segurança emitido
- DynamoDB: um evento de controle de utilização
- Amazon EC2 Auto Scaling falha ao iniciar uma instância.
- Systems Manager: falha na execução de uma automação
- AWS Health: um alerta para manutenção programada
- EC2: alteração do estado da instância de `Running` para `Stopped`

O OpsCenter também é integrado ao Amazon CloudWatch Application Insights para .NET e SQL Server. Isso significa que você pode criar automaticamente OpsItems para problemas detectados em seus aplicativos. Você também pode integrar o OpsCenter ao AWS Security Hub para agregar e tomar medidas sobre seus problemas operacionais, de segurança e de performance no Systems Manager.

Engenheiros de operações e profissionais de TI podem criar, visualizar e editar OpsItems usando a página OpsCenter no console do AWS Systems Manager, operações de API pública, a AWS Command Line Interface (AWS CLI), o AWS Tools for Windows PowerShell ou os AWS SDKs .As operações da API pública OpsCenter também permitem que você integre o OpsCenter a seus sistemas de gerenciamento de casos e painéis de integridade.

Como o OpsCenter beneficia minha organização?

O OpsCenter possibilita um padrão e uma experiência unificada para visualização, trabalho e correção de problemas relacionados aos recursos da AWS. Um padrão e uma experiência unificada melhoram o tempo necessário para corrigir problemas, investigar problemas relacionados e treinar

novos engenheiros de operações e profissionais de TI. Uma experiência padrão e unificada também reduz o número de erros manuais inseridos no sistema de gerenciamento e correção de problemas.

Mais especificamente, o OpsCenter oferece os seguintes benefícios para engenheiros de operações e organizações:

- Não é mais necessário navegar em várias páginas do console para visualizar, investigar e resolver OpsItems relacionados aos recursos da AWS. Os OpsItems são agregados aos serviços, em um local central.
- Você pode visualizar dados específicos do serviço e relevantes contextualmente para o OpsItems, que são gerados automaticamente pelos alarmes do CloudWatch, eventos do EventBridge e pelo CloudWatch Application Insights para .NET e SQL Server.
- Você pode especificar o nome do recurso da Amazon (ARN) de um recurso relacionado a um OpsItem. Ao especificar recursos relacionados, o OpsCenter usa lógica integrada para ajudar a evitar a criação de OpsItems duplicados.
- Você pode visualizar detalhes e informações de resolução sobre OpsItems semelhantes.
- Você pode visualizar rapidamente informações sobre runbooks do Automation do Systems Manager e executá-los para resolver problemas.

Quais são os recursos do OpsCenter?

- Criação de OpsItem automatizada e manual

  O OpsCenter é integrado ao Amazon CloudWatch. Isso significa que você pode configurar o CloudWatch para criar automaticamente um OpsItem em OpsCenter quando um alarme entra no `ALARM` ou quando o Amazon EventBridge processa um evento de qualquer AWS que publica eventos. Você também pode criar OpsItems manualmente.

  O OpsCenter também é integrado ao Amazon CloudWatch Application Insights para .NET e SQL Server. Isso significa que você pode criar automaticamente OpsItems para problemas detectados em seus aplicativos.
- OpsItems detalhados e pesquisáveis

  Cada OpsItem inclui vários campos de informações, inclusive título, ID, prioridade, descrição, origem do OpsItem e data e hora em que foi atualizado pela última vez. Cada OpsItem também inclui os seguintes recursos configuráveis:
  - Status: Aberto, Em andamento, Resolvido ou Aberto e em andamento.

- Recursos relacionados: um recurso relacionado é o recurso afetado ou o recurso que iniciou o evento do EventBridge que criou o OpsItem. Cada OpsItem inclui uma seção Related resources (Recursos relacionados) em que o OpsCenter lista automaticamente o nome do recurso da Amazon (ARN) do recurso selecionado. Você também pode especificar manualmente ARNs de recursos relacionados. Para alguns tipos de ARN, OpsCenter cria automaticamente um deep link que exibe detalhes sobre o recurso sem a necessidade de visitar outras páginas do console para visualizar essas informações. Por exemplo, se especificar o ARN de uma instância do EC2, você poderá visualizar todos os detalhes fornecidos pelo EC2 sobre essa instância em OpsCenter. Você pode adicionar manualmente os ARNs de recursos relacionados adicionais. Cada OpsItem pode listar, no máximo, 100 ARNs de recursos relacionados. Para ter mais informações, consulte Adição de recursos relacionados para um OpsItem.
- Relacionados e similaresOpsItems: com o recurso OpsItems relacionado, é possível especificar os IDs do OpsItems que estão de alguma forma relacionados com o OpsItem. O recurso Similar OpsItem revisa automaticamenteOpsItem títulos e descrições e, em seguida, lista outros OpsItems que possam estar relacionadas ou sejam de seu interesse.
- Dados operacionais pesquisáveis e privados: dados operacionais são dados personalizados que fornecem detalhes da referência úteis sobre o OpsItem. Por exemplo, você pode especificar arquivos de log, strings de erro, chaves de licença, dicas para solução de problemas ou outros dados relevantes. Você insere dados operacionais como pares de chave/valor. A chave tem um tamanho máximo de 128 caracteres. O valor tem um tamanho máximo de 20 KB.

  Esses dados personalizados são pesquisáveis, mas com restrições. Para o recurso Searchable operational data (Dados operacionais pesquisáveis), todos os usuários com acesso à página OpsItem Overview (Visão geral do Opsitem) (conforme fornecido pela operação de API DescribeOpsItems) podem visualizar e pesquisar nos dados especificados. Para o recurso Private operational data (Dados operacionais privados), os dados só são visualizáveis por usuários que tenham acesso ao OpsItem (conforme fornecido pela operação de API GetOpsItem).
- Desduplicação: ao especificar recursos relacionados, o OpsCenter usa a lógica integrada para ajudar a evitar a criação de OpsItems duplicados. O OpsCenter também inclui um recurso chamado Insights operacionais, que exibe informações sobre OpsItems duplicados. Para limitar ainda mais o número de duplicadosOpsItemsna sua conta, você pode especificar manualmente uma string de desduplicação para uma regra de evento EventBridge. Para ter mais informações, consulte Gerenciamento de OpsItems duplicados.

- Edição em massa do OpsItems: você pode selecionar vários OpsItems em OpsCenter e editar um dos seguintes campos: Status (Status), Priority (Prioridade), Severity (Gravidade), Category (Categoria).
- Correção fácil usando registros

  Cada OpsItem inclui uma seção Runbooks com uma lista de runbooks do Systems Manager Automation que você pode usar para corrigir automaticamente problemas comuns com os recursos da AWS. Se você abrir um OpsItem, escolha um recurso da AWS para esse OpsItem e, em seguida, escolha o botão Run automation no console, então o OpsCenter fornece uma lista de runbooks de automação que você pode executar no recurso da AWS que gerou o OpsItem. Depois de executar um runbook de automação em um OpsItem, o runbook será associado automaticamente ao recurso relacionado do OpsItem para referência futura e execução fácil. Além disso, se você configurar automaticamente as regras do OpsItem no EventBridge usando o OpsCenter, o EventBridge associará automaticamente runbooks a eventos comuns. O OpsCenter mantém um registro de 30 dias dos runbooks de automação executados para um OpsItem específico. Para ter mais informações, consulte Correção de problemas do OpsItem.
- Alterar notificação: você pode especificar o ARN de um tópico do Amazon Simple Notification Service (SNS) e publicar notificações sempre que um OpsItem for alterado ou editado. O tópico do SNS deve existir na mesma Região da AWS que o OpsItem.
- Recursos de pesquisa do OpsItem abrangentes: o OpsCenter fornece várias opções de pesquisa para ajudar a localizar rapidamente os OpsItems. Aqui estão vários exemplos de como você pode pesquisar: ID do OpsItem, título, hora da modificação mais recente, valor de dados operacionais, origem e ID do Automation de execução de um runbook, para mencionar alguns. Você pode limitar ainda mais resultados de pesquisa usando filtros de status.
- Relatórios de resumo do OpsItem

  O OpsCenter inclui uma página de relatório de resumo que exibe automaticamente as seguintes seções:

  - Status summary (Resumo de status): um resumo de OpsItems por status (Aberto, Em andamento, Resolvido, Aberto e em andamento).
  - Sources with most open OpsItems (Origens com OIS mais abertas): um detalhamento dos principais serviços da AWS com OpsItems pendentes.
  - OpsItems by source and age (OpsItems por origem e idade): uma contagem de OpsItems, agrupadso por fonte e dias desde a criação.

Para obter mais informações sobre como visualizar relatórios de resumo do OpsCenter, consulte Visualizar relatórios de resumo do OpsCenter.

- Registro de auditoria de suporte a recursos

  Você pode auditar e registrar ações do usuário do OpsCenter em sua Conta da AWS por meio da integração com outros serviços da AWS. Para ter mais informações, consulte Visualização de logs e relatórios do OpsCenter.

- Acesso via console, CLI, PowerShell e SDK à ferramenta OpsCenter

  Você pode trabalhar com o OpsCenter usando console do AWS Systems Manager, o AWS Command Line Interface (AWS CLI), o Ferramentas da AWS para PowerShell ou o SDK da AWS de sua escolha.

O OpsCenter se integra ao meu sistema de gerenciamento de casos existente?

O OpsCenter foi projetado para complementar seus sistemas de gerenciamento de casos existentes. Você pode integrar o OpsItems ao sistema de gerenciamento de casos existentes usando ações de API públicas. Você também pode manter fluxos de trabalho do ciclo manual em seus sistemas atuais e usar o OpsCenter como uma central de investigação e correção.

Para obter informações sobre o as operações de API públicas do OpsCenter, consulte as seguintes operações de API na Referência de API do AWS Systems Manager.

- CreateOpsItem
- DescribeOpsItems
- GetOpsItem
- GetOpsSummary
- UpdateOpsItem

Há cobrança pelo uso do OpsCenter?

Sim. Para obter mais informações, consulte Preços do AWS Systems Manager.

O OpsCenter funciona com meus nós gerenciados híbridos e on-premises?

Sim. Você pode usar o OpsCenter para investigar e corrigir problemas com os nós gerenciados on-premises configurados para o Systems Manager. Para obter mais informações sobreno como definir

e configurar servidores on-premises e máquinas virtuais do Systems Manager, consulte Gerenciar nós em ambientes híbridos e multinuvem com o Systems Manager.

Quais são as cotas para o OpsCenter?

Visualize cotas de todas as ferramentas do Systems Manager em Cotas de serviço do Systems Manager no Referência geral da Amazon Web Services. A menos que especificado de outra forma, cada cota é específica da região.

Configurar o OpsCenter

O AWS Systems Manager usa uma experiência de configuração integrada para ajudar você a começar a usar o OpsCenter e o Explorer, que são ferramentas do Systems Manager. O Explorer corresponde a um painel de operações personalizável que relata informações sobre os recursos da AWS. Nesta documentação, a configuração do Explorer e do OpsCenter é chamada de configuração integrada.

Você deve usar a configuração integrada para configurar o OpsCenter com o Explorer. A Configuração Integrada só está disponível no console do AWS Systems Manager. Você não pode configurar o Explorer ou o OpsCenter programaticamente. Para ter mais informações, consulte Conceitos básicos do Systems Manager Explorer e do OpsCenter.

Antes de começar

Ao configurar OpsCenter, você habilita regras padrão no Amazon EventBridge que são criadas automaticamente OpsItems. A tabela a seguir descreve as regras padrão do EventBridge que criam OpsItems automaticamente. Você pode desativar as regras do EventBridge na página OpsCenterConfiguraçõesOpsItem, em regras.

> **Important**
>
> Sua conta é cobrada por OpsItems criados por regras padrão. Para obter mais informações, consulte Preços do AWS Systems Manager.

Nome da regra	Descrição
SSMOpsItems-Autoscaling-instance-launch-failure	Essa regra cria OpsItems quando o lançamento de uma instância EC2 de ajuste de escala automático falhou.
SSMOpsItems-Autoscaling-instance-termination-failure	Essa regra cria OpsItems quando o término da instância do EC2 de ajuste de escala automático falhou.
SSMOpsItems-EBS-snapshot-copy-failed	Essa regra cria OpsItems quando o sistema falhou ao copiar um snapshot do Amazon Elastic Block Store (Amazon EBS).
SSMOpsItems-EBS-snapshot-creation-failed	Essa regra cria OpsItems quando o sistema não consegue criar o snapshot do Amazon EBS.
SSMOpsItems-EBS-volume-performance-issue	Essa regra corresponde a uma regra de rastreamento AWS Health. A regra cria OpsItems sempre que houver um problema de desempenho com um volume do Amazon EBS (evento de saúde = `AWS_EBS_DEGRADED_EBS_VOLUME_PERFORMANCE`).
SSMOpsItems-EC2-issue	Essa regra corresponde a uma regra de rastreamento AWS Health para eventos inesperados que afetam AWS serviços ou recursos. A regra cria OpsItems quando, por exemplo, um serviço envia comunicações sobre problemas operacionais que estão causando degradação do serviço ou para conscientizar sobre problemas localizados no nível de recursos. Por exemplo, esta regra cria um OpsItem para o seguinte evento: `AWS_EC2_OPERATIONAL_ISSUE` .

Nome da regra	Descrição
SSMOpsItems-EC2-scheduled-change	Essa regra corresponde a uma regra de rastreamento AWS Health. AWS pode programar eventos para suas instâncias, como reinicialização, interrupção ou início. A regra cria OpsItems para eventos programados do EC2. Para obter mais informações sobre eventos programados, consulte Eventos programados para suas instâncias no Guia do usuário do Amazon EC2.
SSMOpsItems-RDS-issue	Essa regra corresponde a uma regra de rastreamento AWS Health para eventos inesperados que afetam AWS serviços ou recursos. A regra cria OpsItems quando, por exemplo, um serviço envia comunicações sobre problemas operacionais que estão causando degradação do serviço ou para conscientizar sobre problemas localizados no nível de recursos. Por exemplo, essa regra cria um OpsItem para os seguintes eventos: `AWS_RDS_MYSQL_DATABASE_CRASHING_REPEATEDLY` `AWS_RDS_EXPORT_TASK_FAILED` e `AWS_RDS_CONNECTIVITY_ISSUE` .

Nome da regra	Descrição
SSMOpsItems-RDS-scheduled-change	Essa regra corresponde a uma regra de rastreamento AWS Health. A regra é criada OpsItems para eventos programados do Amazon RDS. Eventos programados fornecem informações sobre futuras alterações nos recursos do Amazon RDS. Alguns eventos podem recomendar que você tome medidas para evitar interrupções no serviço. Outros eventos ocorrem de forma automática, não exigindo intervenção de sua parte. Seu recurso pode estar temporariamente indisponível durante a atividade de alteração programada. Por exemplo, essa regra cria um OpsItem para os seguintes eventos: `AWS_RDS_SYSTEM_UPGRADE_SCHEDULED` e `AWS_RDS_MAINTENANCE_SCHEDULED` . Para obter mais informações sobre eventos programados, consulte Categorias de tipos de eventos no AWS HealthGuia do usuário.
SSMOpsItems-SSM-maintenance-window-execution-failed	Essa regra cria OpsItems quando a execução da janela de manutenção do Systems Manager falha.
SSMOpsItems-SSM-maintenance-window-execution-timedout	Essa regra cria OpsItems quando a execução da janela de manutenção do Systems Manager ultrapassa o tempo limite.

Utilize o procedimento a seguir para configurar OpsCenter.

Como configurar o OpsCenter

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha OpsCenter.
3. Na página inicial do OpsCenter, escolha Começar a usar.

4. Na página de configuração do OpsCenter, escolha Habilitar esta opção para que o Explorer configure o AWS Config e os eventos do Amazon CloudWatch para criar OpsItems automaticamente com base em regras e eventos comumente usados. Se você não escolher esta opção, o OpsCenter permanecerá desabilitado.

> **Note**
>
> O Amazon EventBridge (antigo Amazon CloudWatch Events) fornece todas as funcionalidades do CloudWatch Events e alguns novos recursos, como barramentos de eventos personalizados, origens de eventos de terceiros e registro dos esquemas.

5. Escolha HabilitarOpsCenter.

Após habilitar o OpsCenter, é possível fazer o seguinte em Configurações:

- Criar alarmes do CloudWatch usando o botão Abrir console do CloudWatch. Para ter mais informações, consulte Configuração dos alarmes do CloudWatch para criar OpsItems.
- Habilitar insights operacionais. Para ter mais informações, consulte Analisar insights operacionais para reduzir OpsItems.
- Habilitar os alarmes de descoberta do AWS Security Hub. Para ter mais informações, consulte Noções básicas sobre a integração do OpsCenter com o AWS Security Hub.

Conteúdo

- (Opcional) Configuração do OpsCenter para gerenciar OpsItems de forma centralizada entre contas
- (Opcional) Configurar o Amazon SNS para receber notificações sobre OpsItems

(Opcional) Configuração do OpsCenter para gerenciar OpsItems de forma centralizada entre contas

Você pode usar o OpsCenter do Systems Manager para gerenciar OpsItems entre várias Contas da AWS, de maneira centralizada, em uma Região da AWS selecionada. O atributo fica disponível depois que você configura sua organização no AWS Organizations. O AWS Organizations é um serviço de gerenciamento de contas que permite consolidar várias contas da AWS em uma única organização que você cria e gerencia de maneira centralizada. O AWS Organizations inclui todas as funcionalidades de gerenciamento de contas e de faturamento consolidado que permitem atender

melhor às necessidades de orçamento, segurança e conformidade de sua empresa. Para obter mais informações, consulte O que é o AWS Organizations? no Guia do usuário do AWS Organizations

Os usuários que pertencem à conta de gerenciamento do AWS Organizations podem configurar uma conta de administrador delegado para o Systems Manager. No contexto do OpsCenter, administradores delegados podem criar, editar e visualizar OpsItems nas contas de membro. O administrador delegado também pode usar os runbooks do Systems Manager Automation para resolver OpsItems em massa ou corrigir problemas com os recursos da AWS que estão gerando OpsItems.

> **Note**
>
> É possível atribuir somente uma conta de administrador delegado do Systems Manager. Para obter mais informações, consulte Criar um administrador delegado do AWS Organizations para o Systems Manager.

O Systems Manager oferece os métodos a seguir para configurar o OpsCenter para gerenciamento centralizado de OpsItems entre várias Contas da AWS.

- Configuração Rápida: a Configuração Rápida é uma ferramenta do Systems Manager que simplifica as tarefas de instalação e configuração dos recursos do Systems Manager. Para obter mais informações, consulte AWS Systems Manager Quick Setup.

  A Configuração Rápida para o OpsCenter ajuda a concluir as seguintes tarefas para gerenciar OpsItems entre contas:

  - Registrar uma conta como administrador delegado (se o administrador delegado ainda não estiver designado)
  - Criar políticas e perfis do AWS Identity and Access Management (IAM) necessários
  - Especificar uma organização do AWS Organizations ou unidades organizacionais (UOs) em que um administrador delegado pode gerenciar OpsItems entre contas

  Para obter mais informações, consulte (Opcional) Configurar o OpsCenter para gerenciar OpsItems entre contas usando a Quick Setup.

> **Note**
>
> A Configuração Rápida não está disponível em todas as Regiões da AWS em que o Systems Manager está disponível atualmente. Se a Configuração Rápida não estiver disponível em uma região em que você deseja usá-la para configurar o OpsCenter para gerenciamento centralizado de OpsItems entre contas, use o método manual. Para ver uma lista de Regiões da AWS onde a Configuração Rápida está disponível, consulte Disponibilidade do Quick Setup nas Regiões da AWS.

- Configuração manual: se a Configuração Rápida não estiver disponível em uma região em que você deseja configurar o OpsCenter para gerenciamento centralizado de OpsItems entre contas, você poderá usar o procedimento manual para isso. Para obter mais informações, consulte (Opcional) Configuração do OpsCenter para gerenciar OpsItems de forma centralizada entre contas.

(Opcional) Configurar o OpsCenter para gerenciar OpsItems entre contas usando a Quick Setup

A Quick Setup, uma ferramenta do AWS Systems Manager, simplifica as tarefas de instalação e configuração das ferramentas do Systems Manager. A Quick Setup para OpsCenter ajuda a concluir as seguintes tarefas de gerenciamento de OpsItems entre contas:

- Especificar a conta de administrador delegado
- Criar políticas e perfis do AWS Identity and Access Management (IAM) necessários
- Especificar uma organização do AWS Organizations ou um subconjunto de contas de membro em que um administrador delegado pode gerenciar OpsItems entre contas

Quando você configura o OpsCenter para gerenciar OpsItems entre contas usando a Configuração Rápida, a Quick Setup cria os recursos a seguir nas contas especificadas. Esses recursos dão às contas especificadas permissão para trabalhar com OpsItems e usar runbooks do Automation para corrigir problemas com recursos da AWS que geram OpsItems.

Recursos	Contas
Perfil do AWS Identity and Access Management (IAM) vinculado ao serviço `AWSServic`	Conta de gerenciamento e conta de administrador delegado do AWS Organizations

Recursos	Contas
`eRoleForAmazonSSM_AccountDiscovery` Para obter mais informações sobre essa função, consulte Usar perfis para coletar informações da Conta da AWS para o OpsCenter e o Explorer.	
Perfil do IAM `OpsItem-CrossAccountManagementRole` Perfil do IAM `AWS-SystemsManager-AutomationAdministrationRole`	Conta de administrador delegado
Perfil do IAM `OpsItem-CrossAccountExecutionRole` Perfil do IAM `AWS-SystemsManager-AutomationExecutionRole` Política de recursos do Systems Manager `AWS::SSM::ResourcePolicy` para o grupo de OpsItem padrão (`OpsItemGroup`)	Todas as contas de membro do AWS Organizations

> **Note**
>
> Se você configurou o OpsCenter anteriormente para gerenciar OpsItems entre contas usando o método manual, será necessário excluir as pilhas ou conjuntos de pilhas do AWS CloudFormation criados durante as etapas 4 e 5 do processo. Se os recursos estiverem presentes em sua conta quando você concluir o procedimento a seguir, a Quick Setup não configurará corretamente o gerenciamento de OpsItem entre contas.

Para configurar o OpsCenter para gerenciar OpsItems entre contas usando a Configuração Rápida

1. Faça login no AWS Management Console usando a conta de gerenciamento do AWS Organizations.
2. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
3. No painel de navegação, escolha Quick Setup.
4. Escolha a guia Biblioteca.
5. Role até a parte inferior e localize o bloco de configuração do OpsCenter. Escolha Criar.
6. Na página de Quick Setup do OpsCenter, na seção Administrador delegado, insira um ID de conta. Se você não conseguir editar esse campo, é porque uma conta de administrador delegado já foi especificada para o Systems Manager.
7. Na seção Select Targets by, escolha uma opção. Se você escolher Personalizar, selecione as unidades organizacionais (UO) nas quais deseja gerenciar OpsItems entre contas.
8. Escolha Criar.

A Quick Setup cria a configuração do OpsCenter e implanta os recursos da AWS necessários nas UOs designadas.

> **Note**
>
> Se você não quiser gerenciar OpsItems entre várias contas, você poderá excluir a configuração da Quick Setup. Ao excluir a configuração, a Quick Setup exclui as seguintes políticas e perfis do IAM criados quando a configuração foi originalmente implantada:
>
> - `OpsItem-CrossAccountManagementRole` da conta de administrador delegado
> - `OpsItem-CrossAccountExecutionRole` e `SSM::ResourcePolicy` de todas as contas de membro do Organizations
>
> A Quick Setup remove a configuração de todas as unidades organizacionais e das Regiões da AWS onde a configuração foi originalmente implantada.

Solução de problemas com uma configuração da Quick Setup para o OpsCenter

Esta seção contém informações que ajudam a solucionar problemas ao configurar o gerenciamento de OpsItem entre contas usando a Quick Setup.

Tópicos

- Falha na implantação destes StackSets: delegatedAdmin
- O status da configuração da Quick Setup é Failed

Falha na implantação destes StackSets: delegatedAdmin

Ao criar uma configuração do OpsCenter, a Quick Setup implanta dois conjuntos de pilhas do AWS CloudFormation na conta de gerenciamento do Organizations. Os conjuntos de pilhas usam o seguinte prefixo: `AWS-QuickSetup-SSMOpsCenter`. Se a Quick Setup exibir este erro: `Deployment to these StackSets failed: delegatedAdmin`, use o procedimento a seguir para corrigir o problema.

Para solucionar um erro failed:delegatedAdmin do StackSets

1. Se você recebeu o erro `Deployment to these StackSets failed: delegatedAdmin` em um banner vermelho no console da Quick Setup, entre na conta do administrador delegado e na Região da AWS designada como a região da Quick Setup de origem.
2. Abra o console do AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
3. Escolha a pilha criada pelo sua configuração do Quick Setup. O nome da pilha contém o seguinte: AWS-QuickSetup-SSMOpsCenter.

   > **Note**
   >
   > Às vezes, o CloudFormation exclui implantações de pilha com falha. Se a pilha não estiver disponível na tabela Stacks (Pilhas), escolha Deleted (Excluído) na lista de filtros.

4. Visualize o Status e Status reason (Motivo do status). Para obter mais informações sobre status de pilhas, consulte Códigos de status da pilha no Guia do usuário do AWS CloudFormation.
5. Para entender a etapa exata que falhou, veja a guia Events (Eventos) e revise cada Status de evento. Para obter mais informações, consulte Troubleshooting no Guia do usuário do AWS CloudFormation.

> **Note**
>
> Se você não conseguir resolver a falha de implantação usando as etapas de solução de problemas do CloudFormation, exclua a configuração e tente novamente.

O status da configuração da Quick Setup é Failed

Se a tabela Detalhes da configuração na página Detalhes da configuração mostrar o status de configuração `Failed`, entre na Conta da AWS e na região em que houve falha.

Para solucionar uma falha da Quick Setup ao criar uma configuração do OpsCenter

1. Faça login na Conta da AWS e na Região da AWS onde a falha ocorreu.
2. Abra o console do AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
3. Escolha a pilha criada pelo sua configuração do Quick Setup. O nome da pilha contém o seguinte: AWS-QuickSetup-SSMOpsCenter.

   > **Note**
   >
   > Às vezes, o CloudFormation exclui implantações de pilha com falha. Se a pilha não estiver disponível na tabela Stacks (Pilhas), escolha Deleted (Excluído) na lista de filtros.

4. Visualize o Status e Status reason (Motivo do status). Para obter mais informações sobre status de pilhas, consulte Códigos de status da pilha no Guia do usuário do AWS CloudFormation.
5. Para entender a etapa exata que falhou, veja a guia Events (Eventos) e revise cada Status de evento. Para obter mais informações, consulte Troubleshooting no Guia do usuário do AWS CloudFormation.

A configuração da conta de membro mostra ResourcePolicyLimitExceededException

Se o status da pilha for `ResourcePolicyLimitExceededException`, a conta já foi integrada ao gerenciamento do OpsCenter entre contas usando o método manual. Para resolver esse problema, é necessário excluir as pilhas ou conjuntos de pilhas do AWS CloudFormation criados durante as etapas 4 e 5 do processo de integração manual. Para obter mais informações, consulte Delete a stack set e Deleting a stack on the AWS CloudFormation console no Guia do usuário do AWS CloudFormation.

(Opcional) Configuração do OpsCenter para gerenciar OpsItems de forma centralizada entre contas

Esta seção descreve como configurar manualmente o OpsCenter para gerenciamento de OpsItem entre contas. Embora ainda haja suporte para esse processo, ele foi substituído por um processo mais novo que usa a Quick Setup do Systems Manager. Para obter mais informações, consulte (Opcional) Configurar o OpsCenter para gerenciar OpsItems entre contas usando a Quick Setup.

É possível configurar uma conta central para criar OpsItems de forma manual para contas de membros, e gerenciar e corrigir esses OpsItems. A conta central pode ser a conta de gerenciamento do AWS Organizations ou a conta de gerenciamento do AWS Organizations e a conta de administrador delegado do Systems Manager. É recomendável usar a conta de administrador delegado do Systems Manager como conta central. É possível usar esse recurso somente depois que você configura o AWS Organizations.

Com o AWS Organizations, é possível consolidar diversas Contas da AWS em uma organização que você cria e gerencia de forma centralizada. O usuário da conta central pode criar OpsItems para todas as contas de membro selecionadas simultaneamente e gerenciar os OpsItems.

Use o processo nesta seção para habilitar a entidade principal de serviço do Systems Manager no Organizations e configurar as permissões do AWS Identity and Access Management (IAM) para trabalhar com OpsItems entre contas.

Tópicos

- Antes de começar
- Etapa 1: criar uma sincronização de dados de recursos
- Etapa 2: habilitar a entidade principal de serviço do Systems Manager no AWS Organizations
- Etapa 3: criar o perfil vinculado ao serviço AWSServiceRoleForAmazonSSM_AccountDiscovery
- Etapa 4: configurar permissões para trabalhar com OpsItems entre contas
- Etapa 5: configurar permissões para trabalhar com recursos relacionados entre contas

> **Note**
>
> Há suporte somente para OpsItems do tipo `/aws/issue` para operação com o OpsCenter entre contas.

Antes de começar

Antes de configurar o OpsCenter para trabalhar com OpsItems entre contas, certifique-se de ter configurado o seguinte:

- Uma conta de administrador delegado do Systems Manager. Para obter mais informações, consulte Configuração de um administrador delegado para Explorer.

- Uma organização definida e configurada no Organizations. Para obter mais informações, consulte Criando e gerenciando uma organização no Guia do UsuárioAWS Organizations.
- Você configurou o Systems Manager Automation para executar runbooks de automação entre Regiões da AWS e contas da AWS. Para obter mais informações, consulte Executar automações em várias regiões e contas da Regiões da AWS.

Etapa 1: criar uma sincronização de dados de recursos

Depois de instalar e configurar o AWS Organizations, será possível agregar OpsItems ao OpsCenter para uma organização inteira criando uma sincronização de dados de recursos. Para obter mais informações, consulte Criar uma sincronização de dados de recursos. Ao criar a sincronização, na seção Adicionar contas, escolha a opção Incluir todas as contas da minha configuração do AWS Organizations.

Etapa 2: habilitar a entidade principal de serviço do Systems Manager no AWS Organizations

Para permitir que um usuário trabalhe com OpsItems entre contas, a entidade principal de serviço do Systems Manager deve estar habilitada no AWS Organizations. Se você configurou anteriormente o Systems Manager para cenários de várias contas usando outras ferramentas, talvez a entidade principal de serviço do Systems Manager já esteja configurada no Organizations. Execute os comandos a seguir a partir da AWS Command Line Interface (AWS CLI) para verificar. Se você não configurou o Systems Manager para outros cenários de diversas contas, vá para o próximo procedimento, Para habilitar a entidade principal de serviço do Systems Manager no AWS Organizations.

Para verificar se a entidade principal de serviço do Systems Manager está habilitada no AWS Organizations

1. Faça download da versão mais recente da AWS CLI para sua máquina local.
2. Abra a AWS CLI e execute o seguinte comando para especificar suas credenciais e uma Região da AWS.

   ```
   aws configure
   ```

   O sistema solicita que você especifique o seguinte. No exemplo a seguir, substitua cada *espaço reservado para entrada do usuário* por suas próprias informações.

   ```
   AWS Access Key ID [None]: key_name
   ```

```
AWS Secret Access Key [None]: key_name
Default region name [None]: region
Default output format [None]: ENTER
```

3. Execute o comando a seguir para verificar se a entidade principal de serviço do Systems Manager está ativada para o AWS Organizations.

```
aws organizations list-aws-service-access-for-organization
```

O comando retorna informações semelhantes às mostradas no exemplo a seguir.

```
{
    "EnabledServicePrincipals": [
        {
            "ServicePrincipal":
 "member.org.stacksets.cloudformation.amazonaws.com",
            "DateEnabled": "2020-12-11T16:32:27.732000-08:00"
        },
        {
            "ServicePrincipal": "opsdatasync.ssm.amazonaws.com",
            "DateEnabled": "2022-01-19T12:30:48.352000-08:00"
        },
        {
            "ServicePrincipal": "ssm.amazonaws.com",
            "DateEnabled": "2020-12-11T16:32:26.599000-08:00"
        }
    ]
}
```

Para habilitar a entidade principal de serviço do Systems Manager no AWS Organizations

Se você não configurou previamente a entidade principal de serviço do Systems Manager para o Organizations, use o procedimento a seguir para fazer isso. Para obter mais informações sobre este comando, consulte enable-aws-service-access na Referência de comandos da AWS CLI.

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito. Para obter informações, consulte Instalar a CLI e Configurar a CLI.
2. Faça download da versão mais recente da AWS CLI para sua máquina local.
3. Abra a AWS CLI e execute o seguinte comando para especificar suas credenciais e uma Região da AWS.

```
aws configure
```

O sistema solicita que você especifique o seguinte. No exemplo a seguir, substitua cada *espaço reservado para entrada do usuário* por suas próprias informações.

```
AWS Access Key ID [None]: key_name
AWS Secret Access Key [None]: key_name
Default region name [None]: region
Default output format [None]: ENTER
```

4. Execute o comando a seguir para habilitar a entidade principal de serviço do Management Manager para o AWS Organizations.

```
aws organizations enable-aws-service-access --service-principal "ssm.amazonaws.com"
```

Etapa 3: criar o perfil vinculado ao serviço **AWSServiceRoleForAmazonSSM_AccountDiscovery**

Uma função vinculada ao serviço como a função `AWSServiceRoleForAmazonSSM_AccountDiscovery` é um tipo exclusivo de perfil do IAM vinculado diretamente a um AWS service (Serviço da AWS), como o Systems Manager. As funções vinculadas a serviços são predefinidas pelo serviço e incluem todas as permissões que o serviço requer para chamar outros Serviços da AWS em seu nome. Para obter mais informações sobre a função vinculada ao serviço do `AWSServiceRoleForAmazonSSM_AccountDiscovery`, consulte Permissões de função vinculada ao serviço para detecção de conta do Systems Manager.

Use o procedimento a seguir para criar a função vinculada ao serviço `AWSServiceRoleForAmazonSSM_AccountDiscovery` usando a AWS CLI. Para obter mais informações sobre o comando usado neste procedimento, consulte create-service-linked-role na Referência de comandos da AWS CLI.

Para criar a função vinculada ao serviço **AWSServiceRoleForAmazonSSM_AccountDiscovery**

1. Faça login na conta de gerenciamento do AWS Organizations.
2. Enquanto estiver conectado à conta de gerenciamento do Organizations, execute o comando a seguir.

```
aws iam create-service-linked-role \
    --aws-service-name accountdiscovery.ssm.amazonaws.com \
```

```
    --description "Systems Manager account discovery for AWS Organizations service-
linked role"
```

Etapa 4: configurar permissões para trabalhar com OpsItems entre contas

Use stacksets do AWS CloudFormation para criar uma política de recursos de `OpsItemGroup` e um perfil de execução do IAM que forneça aos usuários permissão para trabalhar com OpsItems entre contas. Para começar, baixe e descompacte o arquivo `OpsCenterCrossAccountMembers.zip`. Este arquivo contém o arquivo de modelo do `OpsCenterCrossAccountMembers.yaml` AWS CloudFormation. Quando você cria um conjunto de pilhas usando esse modelo, o CloudFormation cria automaticamente a política de recursos `OpsItemCrossAccountResourcePolicy` e a função de execução `OpsItemCrossAccountExecutionRole` na conta. Para obter mais informações sobre a criação de conjuntos de pilhas, consulte Criação de um conjunto de pilhas no Guia do usuário do AWS CloudFormation.

Important

Observe as seguintes informações importantes sobre esta tarefa:

- O stackset deve ser implantado enquanto você está conectado com a conta de gerenciamento do AWS Organizations.
- Você deve repetir esse procedimento enquanto estiver em conexão com todas as contas de destino com as quais deseja trabalhar com OpsItems entre contas, incluindo a conta de administrador delegado.
- Se você quiser habilitar a administração de OpsItems entre contas em diferentes Regiões da AWS, escolha Add all regions (Adicionar todas as regiões) na seção Specify regions (Especificar regiões) do modelo. Não há suporte para a administração de OpsItem entre contas com regiões opcionais.

Etapa 5: configurar permissões para trabalhar com recursos relacionados entre contas

Um OpsItem pode incluir informações detalhadas sobre os recursos afetados, como as instâncias do Amazon Elastic Compute Cloud (Amazon EC2) ou os buckets do Amazon Simple Storage Service (Amazon S3). O perfil de execução `OpsItemCrossAccountExecutionRole`, criado na Etapa 4, fornece ao OpsCenter permissões somente de leitura para que as contas de membros visualizem os recursos relacionados. Você também deve criar um perfil do IAM para fornecer às contas de

gerenciamento permissão para visualizar e interagir com os recursos relacionados, o que você concluirá nesta tarefa.

Para começar, baixe e descompacte o arquivo `OpsCenterCrossAccountManagementRole.zip`. Este arquivo contém o arquivo de modelo do `OpsCenterCrossAccountManagementRole.yaml` AWS CloudFormation. Quando você cria uma pilha usando esse modelo, o CloudFormation cria automaticamente o perfil do IAM `OpsCenterCrossAccountManagementRole` na conta. Para obter mais informações sobre a criação de uma pilha, consulte Criação de uma pilha no console do AWS CloudFormation no Guia do usuário do AWS CloudFormation.

> **Important**
>
> Observe as seguintes informações importantes sobre esta tarefa:
>
> - Se você planeja especificar uma conta como um administrador delegado para o OpsCenter, certifique-se de especificar essa Conta da AWS ao criar a pilha.
> - Você deve executar esse procedimento enquanto estiver em conexão com a conta de gerenciamento do AWS Organizations e, novamente, enquanto estiver em conexão com a conta de administrador delegado.

(Opcional) Configurar o Amazon SNS para receber notificações sobre OpsItems

É possível configurar o OpsCenter para enviar notificações a um tópico do Amazon Simple Notification Service (Amazon SNS) quando o sistema criar um OpsItem ou atualizar um OpsItem existente.

Conclua as etapas a seguir para receber notificações de OpsItems.

- Etapa 1: criar e assinar um tópico do Amazon SNS
- Etapa 2: atualizar a política de acesso do Amazon SNS
- Etapa 3: atualizar a política de acesso do AWS KMS

> **Note**
>
> Caso ative a criptografia no lado do servidor do AWS Key Management Service (AWS KMS) na Etapa 2, você deverá concluir a Etapa 3. Caso contrário, você pode ignorar a Etapa 3.

- Etapa 4: ativar as regras padrão do OpsItems para envio de notificações para novos OpsItems

Etapa 1: criar e assinar um tópico do Amazon SNS

Para receber notificações, crie e assine um tópico do Amazon SNS. Para obter informações, consulte Criar um tópico do Amazon SNS e Assinar tópico do Amazon SNS no Guia do desenvolvedor do Amazon Simple Notification Service.

> **Note**
>
> Caso esteja usando o OpsCenter em diversas contas ou Regiões da AWS, você deverá criar e assinar um tópico do Amazon SNS em cada região ou conta em que deseja receber notificações do OpsItem.

Etapa 2: atualizar a política de acesso do Amazon SNS

É necessário associar um tópico do Amazon SNS a OpsItems. Use o procedimento a seguir para configura uma política de acesso do Amazon SNS para que o Systems Manager possa publicar notificações do OpsItems no tópico do Amazon SNS que você criou na Etapa 1.

1. Faça login no AWS Management Console e abra o console do Amazon SNS em https://console.aws.amazon.com/sns/v3/home.
2. No painel de navegação, escolha Tópicos.
3. Escolha o tópico criado na Etapa 1 e, em seguida, selecione Editar.
4. Expanda Access policy (Política de acesso).
5. Adicione o seguinte bloco `Sid` à política existente. Substitua cada *`espaço reservado para recurso de exemplo`* por suas próprias informações.

```
{
      "Sid": "Allow OpsCenter to publish to this topic",
      "Effect": "Allow",
      "Principal": {
        "Service": "ssm.amazonaws.com"
      },
      "Action": "SNS:Publish",
      "Resource": "arn:aws:sns:region:account ID:topic name", // Account ID of the
 SNS topic owner
      "Condition": {
```

```
      "StringEquals": {
        "AWS:SourceAccount": "account ID" //  Account ID of the OpsItem owner
      }
    }
}
```

> **Note**
>
> A chave de condição global `aws:SourceAccount` protege contra o cenário confused deputy. Para usar essa chave de condição, defina o valor como o ID da conta do proprietário do OpsItem. Para obter mais informações, consulte O problema de "confused deputy" no Guia do usuário do IAM.

6. Escolha Salvar alterações.

O sistema agora envia notificações ao tópico do Amazon SNS quando OpsItems são criados ou atualizados.

> **Important**
>
> Caso configure o tópico do Amazon SNS com uma chave de criptografia no lado do servidor do AWS Key Management Service (AWS KMS) na Etapa 2, você deverá concluir a Etapa 3. Caso contrário, você pode ignorar a Etapa 3.

Etapa 3: atualizar a política de acesso do AWS KMS

Caso tenha ativado a criptografia no lado do servidor do AWS KMS para seu tópico do Amazon SNS, você também deverá atualizar a política de acesso da AWS KMS key escolhida ao configurar o tópico. Use o procedimento a seguir para atualizar a política de acesso para que o Systems Manager possa publicar notificações do OpsItem no tópico do Amazon SNS criado na Etapa 1.

> **Note**
>
> O OpsCenter não oferece suporte à publicação de OpsItems em um tópico do Amazon SNS configurado com uma Chave gerenciada pela AWS.

1. Abra o console do AWS KMS em https://console.aws.amazon.com/kms.

2. Para alterar a Região da AWS, use o seletor de regiões no canto superior direito da página.
3. No painel de navegação, escolha Customer managed keys (Chaves gerenciadas pelo cliente).
4. Escolha o ID da chave do KMS escolhida ao criar o tópico.
5. Na seção Key Policy (Política de chaves), selecione Switch to policy view (Alternar para a visualização da política).
6. Selecione a opção Editar.
7. Adicione o seguinte bloco `Sid` à política existente. Substitua cada *`espaço reservado para recurso de exemplo`* por suas próprias informações.

```
{
      "Sid": "Allow OpsItems to decrypt the key",
      "Effect": "Allow",
      "Principal": {
        "Service": "ssm.amazonaws.com"
      },
      "Action": ["kms:Decrypt", "kms:GenerateDataKey*"],
       "Resource": "arn:aws:kms:region:account ID:key/key ID"
    }
```

No exemplo a seguir, o novo bloco é inserido na linha 14.



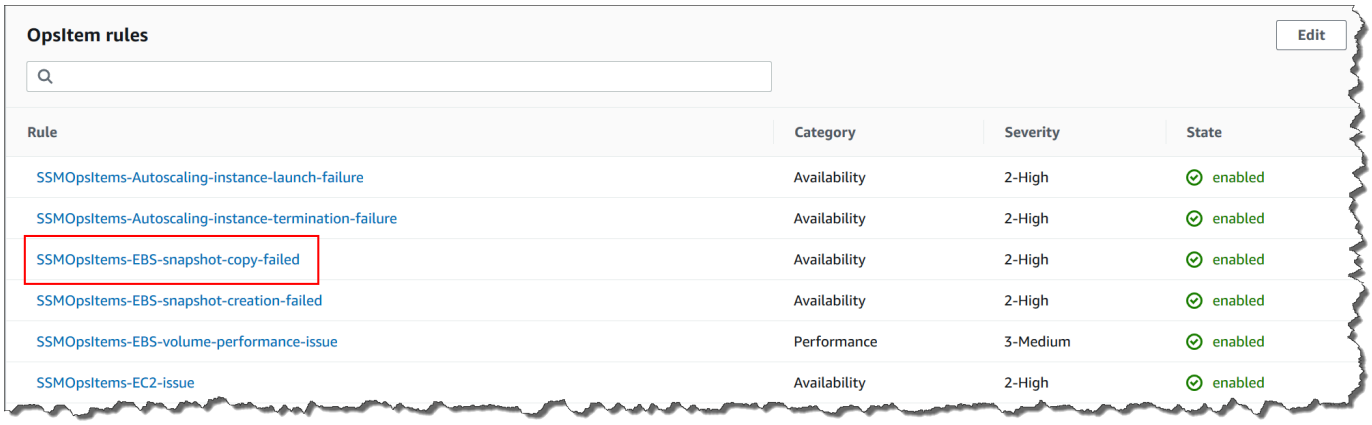
8. Escolha Salvar alterações.

Etapa 4: ativar as regras padrão do OpsItems para envio de notificações para novos OpsItems

As regras padrão de OpsItems no Amazon EventBridge não são configuradas com um nome do recurso da Amazon (ARN) para notificações do Amazon SNS. Use o procedimento a seguir para editar uma regra no EventBridge e inserir um bloco de `notifications`.

Para adicionar um bloco de notificações a uma regra de OpsItem padrão

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha OpsCenter.
3. Escolha a guia OpsItems e escolha Configure sources (Configurar origens).
4. Escolha o nome da regra de origem que você deseja configurar com um bloco `notifications`, conforme mostrado no exemplo a seguir.



   A regra é aberta no Amazon EventBridge.
5. Na página de detalhes, na guia Targets (Destinos), selecione Edit (Editar).
6. Na seção Additional settings (Configurações adicionais), escolha Configure input transformer (Configurar transformador de entrada).
7. Na caixa Modelo, adicione um bloco `notifications` no formato a seguir.

```
"notifications":[{"arn":"arn:aws:sns:region:account ID:topic name"}],
```

   Aqui está um exemplo.

```
"notifications":[{"arn":"arn:aws:sns:us-west-2:1234567890:MySNSTopic"}],
```

Insira o bloco de notificações antes do bloco `resources`, conforme mostrado no exemplo a seguir para a região Oeste dos EUA (Oregon) (us-west-2).

```
{
    "title": "EBS snapshot copy failed",
    "description": "CloudWatch Event Rule SSMOpsItems-EBS-snapshot-copy-failed was
 triggered. Your EBS snapshot copy has failed. See below for more details.",
    "category": "Availability",
    "severity": "2",
    "source": "EC2",
    "notifications": [{
        "arn": "arn:aws:sns:us-west-2:1234567890:MySNSTopic"
    }],
    "resources": <resources>,
    "operationalData": {
        "/aws/dedup": {
            "type": "SearchableString",
            "value": "{\"dedupString\":\"SSMOpsItems-EBS-snapshot-copy-failed\"}"
        },
        "/aws/automations": {
            "value": "[ { \"automationType\": \"AWS:SSM:Automation\",
 \"automationId\": \"AWS-CopySnapshot\" } ]"
        },
        "failure-cause": {
            "value": <failure - cause>
        },
        "source": {
            "value": <source>
        },
        "start-time": {
            "value": <start - time>
        },
        "end-time": {
            "value": <end - time>
        }
    }
}
```

8. Selecione a opção Confirmar.
9. Escolha Próximo.
10. Escolha Próximo.

11. Escolha Upgrade rule (Atualizar regra).

Na próxima vez que o sistema criar um OpsItem para a regra padrão, ele publicará uma notificação no tópico do Amazon SNS.

Integrar o OpsCenter a outros Serviços da AWS

O OpsCenter, uma ferramenta do AWS Systems Manager, integra-se a diversos Serviços da AWS para diagnosticar e corrigir problemas com recursos da AWS. Você deve configurar o AWS service (Serviço da AWS) antes de integrá-lo ao OpsCenter.

Por padrão, os Serviços da AWS a seguir são integrados ao OpsCenter e podem criar OpsItems automaticamente:

- Amazon CloudWatch
- Amazon CloudWatch Application Insights
- Amazon EventBridge
- AWS Config
- AWS Systems Manager Incident Manager

Você precisa integrar os serviços a seguir com o OpsCenter para criar OpsItems automaticamente:

- Amazon DevOps Guru
- AWS Security Hub

Quando qualquer um desses serviços cria um OpsItem, é possível gerenciar e corrigir o OpsItem do OpsCenter. Para obter mais informações, consulte Gerenciamento de OpsItems e Correção de problemas do OpsItem.

Para obter mais informações sobre cada AWS service (Serviço da AWS) e como ele se integra ao OpsCenter, consulte os tópicos a seguir.

Tópicos

- Noções básicas sobre a integração do OpsCenter com o Amazon CloudWatch
- Noções básicas sobre a integração do OpsCenter com o Amazon CloudWatch Application Insights
- Noções básicas sobre a integração do OpsCenter com o Amazon DevOps Guru

- Noções básicas sobre a integração do OpsCenter com o Amazon Eventbridge
- Noções básicas sobre a integração do OpsCenter com o AWS Config
- Noções básicas sobre a integração do OpsCenter com o AWS Security Hub
- Noções básicas sobre a integração do OpsCenter com o Incident Manager

Noções básicas sobre a integração do OpsCenter com o Amazon CloudWatch

O Amazon CloudWatch monitora seus recursos e serviços da AWS e exibe métricas em cada AWS service (Serviço da AWS) que você usa. O CloudWatch cria um OpsItem quando um alarme entra no estado de alarme. Por exemplo, você pode configurar um alarme para criar automaticamente um OpsItem se houver um pico de erros HTTP gerados pelo Application Load Balancer.

Alguns alarmes que você pode configurar no CloudWatch para criar OpsItems são mostrados na lista a seguir:

- Amazon DynamoDB: ações de leitura e gravação de banco de dados atingem um limite
- Amazon EC2: a utilização da CPU atinge um limite
- Faturamento da AWS: cobranças estimadas atingem um limite
- Amazon EC2: falha na verificação de status de uma instância
- Amazon Elastic Block Store (EBS): a utilização do espaço em disco atinge um limite

É possível criar um alarme ou editar um alarme existente para criar um OpsItem. Para obter mais informações, consulte Configuração dos alarmes do CloudWatch para criar OpsItems.

Quando você habilita o OpsCenter usando a configuração integrada, ele integra o CloudWatch com o OpsCenter.

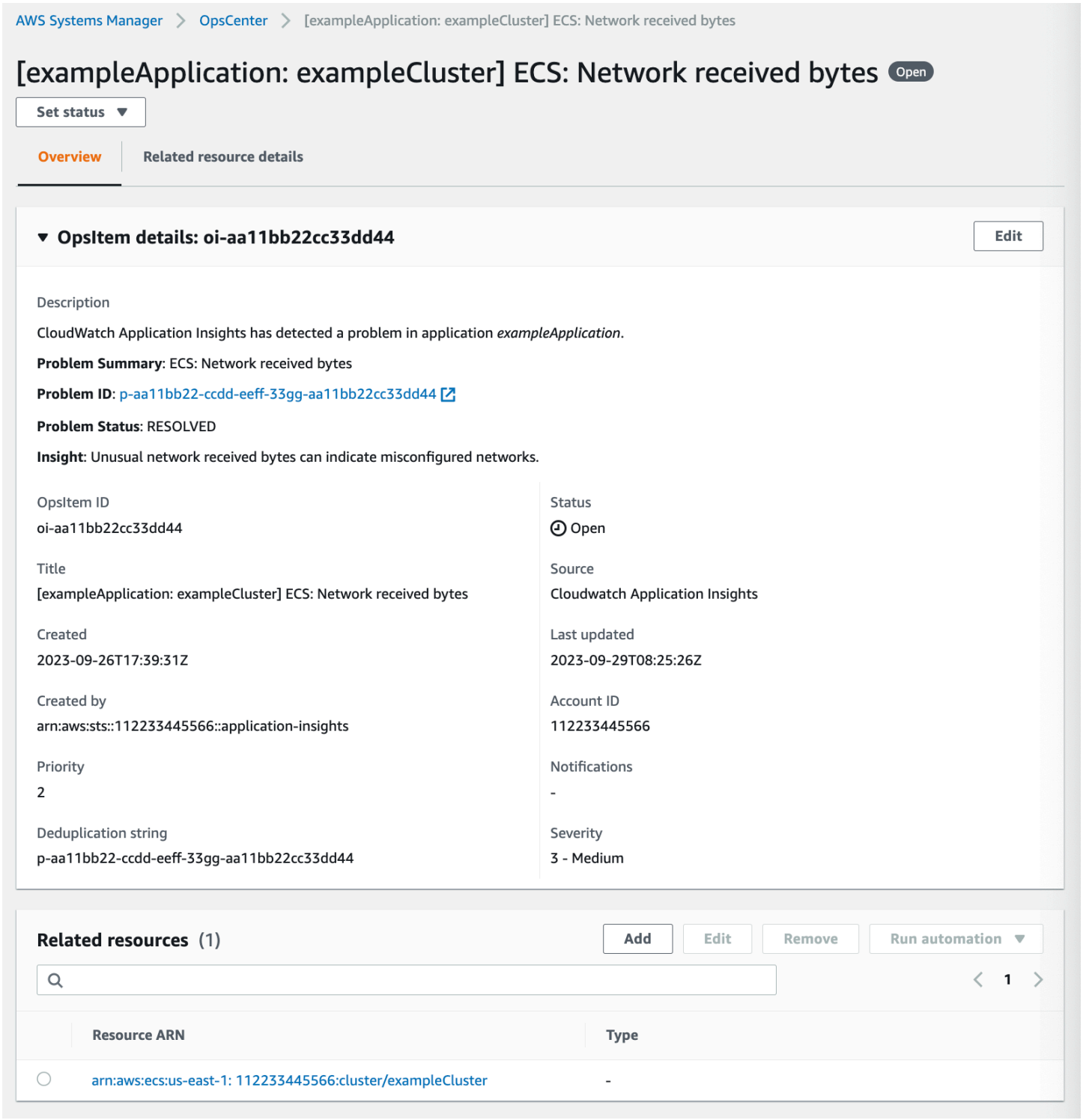
Noções básicas sobre a integração do OpsCenter com o Amazon CloudWatch Application Insights

Ao usar o Amazon CloudWatch Application Insights, é possível configurar os monitores mais apropriados para seus recursos de aplicação a fim de analisar dados continuamente em busca de sinais de problemas com as aplicações. Ao configurar recursos de aplicação no CloudWatch Application Insights, é possível optar por fazer com que o sistema crie OpsItems no OpsCenter. Um OpsItem é criado no console do OpsCenter para cada problema detectado com a aplicação. Para obter informações, consulte Instalar, configurar e gerenciar sua aplicação para monitoramento no Guia do usuário do Amazon CloudWatch.

> **Note**
>
> A partir de 16 de outubro de 2023, o título e a descrição de OpsItems criados pelo CloudWatch Application Insights passarão a usar este formato aprimorado:

```
OpsItem title: [<APPLICATION NAME>: <RESOURCE ID>] <PROBLEM SUMMARY>

OpsItem description:

CloudWatch Application Insights has detected a problem in application <APPLICATION
 NAME>.
Problem summary: <PROBLEM SUMMARY>
Problem ID: <PROBLEM ID> (hyperlinks to the Application Insights problem summary page)
Problem Status: <PROBLEM STATUS>
Insight: <INSIGHT>
```

Exemplo:



Noções básicas sobre a integração do OpsCenter com o Amazon DevOps Guru

O Amazon DevOps Guru aplica machine learning para analisar dados operacionais, métricas de aplicações e eventos de aplicações para identificar comportamentos que se desviam dos padrões

operacionais normais. Se você permitir que o DevOps Guru gere um OpsItem no OpsCenter, cada insight gerará um novo OpsItem. É possível usar o OpsCenter para gerenciar seus OpsItems.

O DevOps Guru cria OpsItems automaticamente. É possível habilitar o Amazon DevOps Guru para criar OpsItems usando a Quick Setup, uma ferramenta do Systems Manager. O sistema cria OpsItems ao usar o perfil vinculado ao serviço AWSServiceRoleForDevOpsGuru do AWS Identity and Access Management (IAM).

Como integrar o OpsCenter com o DevOps Guru

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Quick Setup.
3. Na página Personalizar opções de configuração do DevOps Guru, escolha a guia Biblioteca.
4. No painel DevOps Guru, escolha Criar.
5. Em Opções de configuração, selecione Habilitar OpsItems do AWS Systems Manager.
6. Selecione Criar após concluir a configuração.

Noções básicas sobre a integração do OpsCenter com o Amazon Eventbridge

O Amazon EventBridge oferece uma transmissão de eventos que descrevem as alterações nos recursos da AWS. Quando você habilita o OpsCenter usando a configuração integrada, ele integra o EventBridge com o OpsCenter e habilita as regras padrão do EventBridge. Com base nessas regras, o EventBridge cria OpsItems. Ao usar regras, é possível filtrar e rotear eventos para o OpsCenter para investigação e correção.

> **Note**
>
> O Amazon EventBridge (antigo Amazon CloudWatch Events) fornece todas as funcionalidades do CloudWatch Events e alguns novos recursos, como barramentos de eventos personalizados, origens de eventos de terceiros e registro dos esquemas.

A seguir estão algumas regras que você pode configurar no EventBridge para criar um OpsItem:

- Security Hub: alerta de segurança emitido
- Amazon DynamoDB: um evento de controle de utilização
- Amazon Elastic Compute Cloud Auto Scaling: falha ao iniciar uma instância

- Systems Manager: falha na execução de uma automação
- AWS Health: um alerta para manutenção programada
- Amazon EC2: estado da instância alterado de em execução para interrompida

Com base em seus requisitos, é possível criar uma regra ou editar uma regra existente para criar OpsItems. Para obter instruções sobre como editar uma regra para criar um OpsItem, consulte Configurar regras do EventBridge para criar OpsItems.

Noções básicas sobre a integração do OpsCenter com o AWS Config

AWS Config fornece uma visão detalhada da configuração dos atributos da AWS em sua Conta da AWS.

O AWS Config não se integra diretamente ao OpsCenter. Em vez disso, você cria uma regra do AWS Config que envia um evento ao Amazon EventBridge, como quando o AWS Config detecta uma instância fora de conformidade. Em seguida, o EventBridge avalia esse evento segundo uma regra do EventBridge que você criou. Se a regra corresponder, o EventBridge transformará o evento em um OpsItem e o transmitirá ao OpsCenter como destino.

Usando esse OpsItem, é possível acompanhar detalhes do recurso que não está em conformidade, registrar ações investigativas e fornecer acesso a ações de correção consistentes.

Informações relacionadas

Configurar regras do EventBridge para criar OpsItems

Usar o AWS Systems Manager OpsCenter e o AWS Config para monitoramento de conformidade

Noções básicas sobre a integração do OpsCenter com o AWS Security Hub

O AWS Security Hub coleta dados de segurança, chamados descobertas, entre Contas da AWS e serviços. Utilizando um conjunto de regras para detectar e gerar descobertas, o Security Hub ajuda a identificar, priorizar e corrigir problemas de segurança dos recursos que você gerencia. Depois que você configura a integração, conforme descrito neste tópico, o Systems Manager cria OpsItems para as descobertas do Security Hub no OpsCenter.

> **Note**
>
> O OpsCenter tem integração bidirecional com o Security Hub. Isso significa que, se você atualizar o campo Status ou Gravidade de um OpsItem relacionado a uma descoberta

de segurança, o sistema sincronizará as alterações com o Security Hub. Da mesma forma, todas as alterações da descoberta são atualizadas automaticamente no OpsItems correspondente no OpsCenter.
Quando um OpsItem é criado a partir de uma descoberta do Security Hub, os metadados do Security Hub são adicionados automaticamente ao campo de dados operacionais do OpsItem. Se esses metadados forem excluídos, as atualizações bidirecionais não funcionarão mais.

Por padrão, o Systems Manager cria OpsItems para descobertas críticas e de alta gravidade. É possível configurar o OpsCenter manualmente para criar OpsItems para descobertas de gravidade média e baixa. O OpsCenter não cria OpsItems para descobertas informativas porque elas não precisam de correções. Para obter mais informações sobre níveis de gravidade do Security Hub, consulte Severity na AWS Security Hub API Reference.

Antes de começar

Antes de configurar o OpsCenter para criar OpsItems com base nas descobertas do Security Hub, verifique se você concluiu as tarefas de configuração do Security Hub. Para obter mais informações , consulte Configurar o Security Hub no Manual do usuário do AWS Security Hub.

Quando você integra o Security Hub com o OpsCenter, o sistema cria OpsItems usando o perfil vinculado ao serviço `AWSServiceRoleForSystemsManagerOpsDataSync` do IAM. Para obter mais informações sobre essa função, consulte Usar perfis para criar OpsData e OpsItems para o Explorer.

Warning

Observe as seguintes informações importantes sobre os preços da integração do OpsCenter com o Security Hub:

- Se você estiver conectado à conta de administrador do Security Hub ao configurar uma integração do OpsCenter com o Security Hub, o sistema cria OpsItems para descobertas no administrador e em todas as contas de membro. Todos os OpsItems são criados na conta do administrador. Dependendo de vários fatores, poderá resultar em uma fatura inesperadamente grande da AWS.

  Se você estiver conectado a uma conta de membro ao configurar a integração, o sistema só criará OpsItems para descobertas nessa conta individual. Para obter mais informações

sobre a conta do administrador do Security Hub, as contas de membro e sua relação com o feed de eventos do EventBridge para descobertas, consulte Types of Security Hub integration with EventBridge no Guia do usuário do AWS Security Hub.

- Para cada descoberta que cria um OpsItem, cobra-se o preço normal pela criação do OpsItem. Você também será cobrado se editar OpsItem ou se a descoberta correspondente for atualizada no Security Hub (o que aciona uma OpsItem atualização).
- Os OpsItems que são criados por uma integração com o AWS Security Hub não estão atualmente limitados pela cota máxima de 500.000 OpsItems por conta em uma Região. Portanto, é possível que os alertas do Security Hub criem mais de 500.000 OpsItems cobráveis em cada Região em uma conta.

  Portanto, para ambientes de alta produção, recomendamos limitar o escopo das descobertas do Security Hub somente a problemas de alta severidade.

Para configurar o OpsCenter para criar OpsItems para descobertas do Security Hub

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha OpsCenter.
3. Escolha Settings.
4. Na seção Descobertas do Security Hub, escolha Editar.
5. Escolha o controle deslizante para alterar Desabilitado para Habilitado.
6. Se você quiser que o sistema crie OpsItems para descobertas de gravidade média ou baixa, ative essas opções.
7. Selecione Save (Salvar) para salvar suas configurações.

Use o procedimento a seguir, caso não queira mais que o sistema crie OpsItems para descobertas do Security Hub.

Para interromper o recebimento de OpsItems para descobertas do Security Hub

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha OpsCenter.
3. Escolha Settings.
4. Na seção Descobertas do Security Hub, escolha Editar.

5. Escolha o controle deslizante para alterar Habilitado para Desabilitado. Se não conseguir ativar o controle deslizante, o Security Hub não foi ativado para o seu Conta da AWS.
6. Escolha Salvar para salvar a configuração. O OpsCenter não cria mais OpsItems com base nas descobertas do Security Hub.

> **Important**
>
> Um administrador delegado do Systems Manager ou a conta de gerenciamento do AWS Organizations podem habilitar as descobertas do Security Hub no OpsCenter para várias contas e Regiões da AWS e criar uma sincronização de dados de recursos no Explorer. Se a fonte do Security Hub estiver habilitada no Explorer e houver uma sincronização de dados do recurso que tenha como alvo a conta-membro em que você desabilitou a integração do Security Hub, as configurações selecionadas pelo administrador terão precedência. O OpsCenter continua a criar OpsItems para as descobertas do Security Hub. Para parar de criar OpsItems para descobertas do Security Hub em uma conta-membro visada por uma sincronização de dados de recursos, entre em contato com seu administrador e peça que ele remova sua conta da sincronização de dados do recurso ou desative a origem Security Hub no Explorer. Para obter informações sobre como alterar essa configuração no Explorer, consulte Edite as fontes de dados do Systems Manager Explorer.

Noções básicas sobre a integração do OpsCenter com o Incident Manager

O Incident Manager, uma ferramenta do AWS Systems Manager, fornece um console de gerenciamento de incidentes que ajuda você a mitigar e se recuperar de incidentes que afetam suas aplicações hospedadas na AWS. Um incidente é qualquer interrupção ou redução não planejada na qualidade dos serviços. Após instalar e configurar o Incident Manager, o sistema cria automaticamente OpsItems no OpsCenter.

Quando o sistema cria um incidente no Incident Manager, ele também cria um OpsItem no OpsCenter e exibe o incidente como um item relacionado. Se o OpsItem já existir, o Incident Manager não criará um OpsItem. O primeiro OpsItem é conhecido como OpsItem pai. Se um incidente crescer em escala e escopo, é possível adicionar incidentes a um OpsItem existente. Se necessário, é possível criar manualmente um incidente para um OpsItem. Após o fechamento de um incidente, você pode criar uma análise no Incident Manager para analisar e aprimorar o processo de correção para problemas semelhantes.

Por padrão, o OpsCenter se integra com o Incident Manager. Se o Incident Manager não estiver configurado, a página do OpsCenter exibirá uma mensagem para configuração do Incident Manager. Quando o Incident Manager cria um OpsItem, é possível gerenciar e corrigir o OpsItem do OpsCenter. Para obter instruções sobre como criar um incidente para um OpsItem, consulte Criação de um incidente para um OpsItem.

Criar OpsItems

Após configurar o OpsCenter, uma ferramenta do AWS Systems Manager, e integrá-lo aos seus Serviços da AWS, seus Serviços da AWS criam OpsItems automaticamente com base em regras, eventos ou alarmes padrões.

É possível visualizar os status e os níveis de severidade das regras padrão do Amazon EventBridge. Se necessário, você pode criar ou editar essas regras no Amazon EventBridge. Também é possível visualizar alarmes do Amazon CloudWatch e criar ou editar alarmes. Ao usar regras e alarmes, você pode configurar eventos para os quais deseja gerar OpsItems automaticamente.

Quando o sistema cria um OpsItem, ele está no status Aberto. É possível alterar o status para Em andamento ao iniciar a investigação do OpsItem e para Resolvido após corrigir o OpsItem. Para obter mais informações sobre como configurar alarmes e regras nos Serviços da AWS para criar OpsItems e como criar OpsItems manualmente, consulte os tópicos a seguir.

Tópicos

- Configurar regras do EventBridge para criar OpsItems
- Configuração dos alarmes do CloudWatch para criar OpsItems
- Criar OpsItems manualmente

Configurar regras do EventBridge para criar OpsItems

Quando o Amazon EventBridge recebe um evento, ele cria um novo OpsItem com base nas regras padrão. É possível criar uma regra ou editar uma regra existente para definir o OpsCenter como o destino de um evento do EventBridge. Para obter informações sobre como criar uma regra de evento, consulte Criar uma regra para um AWS service (Serviço da AWS), no Guia do usuário do Amazon EventBridge.

Como configurar uma regra do EventBridge para criar OpsItems no OpsCenter

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Regras.

3. Na página Rules (Regras), para Event bus (Barramento de eventos), escolha default (padrão).
4. Em Regras, escolha uma regra marcando a caixa de seleção ao lado de seu nome.
5. Selecione o nome da regra para abrir sua página de detalhes. Em Detalhes da regra, verifique se o Status está definido como Habilitado.

> **Note**
>
> Se necessário, é possível atualizar o status usando Editar no canto superior direito da página.

6. Escolha a guia Destinos.
7. Na guia Destinos, selecione Editar.
8. Para Tipos de destino, selecione AWS service (Serviço da AWS).
9. Para Select a target (Selecionar um destino), escolha Systems Manager OpsItem.
10. Para muitos tipos de destino, o EventBridge precisa de permissão para enviar eventos ao destino. Nesses casos, o Eventbridge pode criar a função do AWS Identity and Access Management (IAM) necessária para que sua regra seja executada.
    - Para criar um perfil do IAM automaticamente, escolha Create a new role for this specific resource.
    - Para usar um perfil do IAM que você criou para dar permissão ao EventBridge para criar OpsItems no OpsCenter, escolha Use existing role (Usar função existente).
11. Em Configurações adicionais, para Configurar entrada de destino, escolha Transformador de entrada.

    É possível usar a opção Transformador de entrada para especificar uma string de eliminação de duplicação e outras informações importantes para os OpsItems, como título e severidade.
12. Escolha Configurar o transformador de entrada.
13. Em Transformador de entrada de destino, para Caminho de entrada, especifique os valores a serem analisados do evento de acionamento. Por exemplo, para analisar a hora de início, a hora de término e outros detalhes do evento que aciona a regra, use o seguinte JSON.

```
{
    "end-time": "$.detail.EndTime",
    "failure-cause": "$.detail.cause",
    "resources": "$.resources[0]",
```

```
    "source": "$.detail.source",
    "start-time": "$.detail.StartTime"
}
```

14. Em Template (Modelo), especifique as informações a serem enviadas ao destino. Por exemplo, use o seguinte JSON para transmitir informações ao OpsCenter. As informações são utilizadas para criar um OpsItem.

> **Note**
>
> Se o modelo de entrada estiver no formato JSON, o valor do objeto no modelo não poderá conter aspas. Por exemplo, os valores de recursos, a causa da falha, a origem, a hora de início e a hora de fim não podem estar entre aspas.

```
{
    "title": "EBS snapshot copy failed",
    "description": "CloudWatch Event Rule SSMOpsItems-EBS-snapshot-copy-failed was
 triggered. Your EBS snapshot copy has failed. See below for more details.",
    "category": "Availability",
    "severity": "2",
    "source": "EC2",
    "operationalData": {
        "/aws/dedup": {
            "type": "SearchableString",
            "value": "{\"dedupString\":\"SSMOpsItems-EBS-snapshot-copy-failed\"}"
        },
        "/aws/automations": {
            "value": "[ { \"automationType\": \"AWS:SSM:Automation\",
 \"automationId\": \"AWS-CopySnapshot\" } ]"
        },
        "failure-cause": {
            "value": <failure-cause>
        },
        "source": {
            "value": <source>
        },
        "start-time": {
            "value": <start-time>
        },
        "end-time": {
            "value": <end-time>
```

```
        },
         },
        "resources": {
            "value": <resources>
        }
    }
}
```

Para obter mais informações sobre esses campos, consulte Transformar a entrada de destino no Manual do usuário do Amazon EventBridge.

15. Escolha Confirmar.
16. Escolha Próximo.
17. Escolha Próximo.
18. Escolha Upgrade rule (Atualizar regra).

Depois que uma OpsItem for criada com base em um evento, você poderá visualizar os detalhes do evento abrindo a OpsItem e rolando para baixo até a seção Private operational data (Dados operacionais privados). Para obter informações sobre como configurar as opções em um OpsItem, consulte Gerenciamento de OpsItems.

Configuração dos alarmes do CloudWatch para criar OpsItems

Durante a configuração integrada do OpsCenter, uma ferramenta do AWS Systems Manager, você habilita o Amazon CloudWatch para criar OpsItems automaticamente com base em alarmes comuns. É possível criar um alarme ou editar um alarme existente para criar OpsItems no OpsCenter.

O CloudWatch cria um novo perfil vinculado ao serviço no AWS Identity and Access Management (IAM) quando você configura um alarme para criar OpsItems. A nova função é nomeada como `AWSServiceRoleForCloudWatchAlarms_ActionSSM`. Para obter mais informações sobre os perfis vinculados ao serviço do CloudWatch, consulte Usar funções vinculadas ao serviço para o CloudWatch no Guia do usuário do Amazon CloudWatch.

Quando um alarme do CloudWatch gera um OpsItem, o OpsItem exibe o alarme do CloudWatch "***alarm_name***" está no estado ALARM.

Para visualizar detalhes sobre um OpsItem específico, escolha o OpsItem e, em seguida, selecione a guia Detalhes de recursos relacionados. É possível editar os OpsItems manualmente para alterar detalhes, como a severidade ou a categoria. No entanto, ao editar a severidade ou a categoria de um alarme, o Systems Manager não pode atualizar a severidade ou a categoria dos OpsItems

que já foram criados no alarme. Se um alarme criou um OpsItem e você especificou uma string de desduplicação, o alarme não criará outro OpsItems, mesmo se você editá-lo no CloudWatch. Se o OpsItem for resolvido no OpsCenter, o CloudWatch criará um novo OpsItem.

Para obter mais informações sobre como configurar alarmes do CloudWatch, consulte os tópicos a seguir.

Tópicos

- Configuração de um alarme do CloudWatch para criar OpsItems (console)
- Configuração de um alarme existente do CloudWatch para criar OpsItems (programaticamente)

Configuração de um alarme do CloudWatch para criar OpsItems (console)

É possível criar um alarme ou atualizar um alarme existente para criar OpsItems manualmente no Amazon CloudWatch.

Como criar um alarme do CloudWatch e configurar o Systems Manager como destino desse alarme

1. Conclua as etapas de 1 a 9, conforme especificado em Criar um alarme do CloudWatch com base em um limite estático no Guia do usuário do Amazon CloudWatch.
2. Na seção Ação do Systems Manager, escolha Adicionar ação do OpsCenter do Systems Manager.
3. Escolha OpsItems.
4. Em Severidade, escolha de 1 a 4.
5. (Opcional) Em Categoria, escolha uma categoria para o OpsItem.
6. Conclua as etapas de 11 a 13, conforme especificado em Criar um alarme do CloudWatch com base em um limite estático no Guia do usuário do Amazon CloudWatch.
7. Selecione Next (Próximo) e conclua o assistente.

Para editar um alarme existente e configurar o Systems Manager como destino desse alarme

1. Abra o console do CloudWatch, em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Alarmes.
3. Selecione o alarme e escolha Actions (Ações), Edit (Editar).
4. (Opcional) Altere as configurações nas seções Metrics (Métricas) e Conditions (Condições) e, em seguida, escolha Next (Próximo).

5. NoSystems Manager (Gerenciador de sistemas), selecioneAdicionar Systems ManagerOpsCenterAção do.
6. Para Severity (Gravidade), selecione um número.

> **Note**
>
> A gravidade é um valor definido pelo usuário. Você ou sua organização determinam o que significa cada valor de gravidade e todo contrato de nível de serviço associado a cada gravidade.

7. (Opcional) Em Category (Categoria), escolha uma categoria.
8. Selecione Next (Próximo) e conclua o assistente.

Configuração de um alarme existente do CloudWatch para criar OpsItems (programaticamente)

É possível configurar alarmes do Amazon CloudWatch para criar OpsItems programaticamente usando a AWS Command Line Interface (AWS CLI), modelos do AWS CloudFormation ou trechos de código Java.

Tópicos

- Antes de começar
- Como configurar alarmes do CloudWatch para criar OpsItems (AWS CLI)
- Como configurar alarmes do CloudWatch para criar ou atualizar OpsItems (CloudFormation)
- Como configurar alarmes do CloudWatch para criar ou atualizar OpsItems (Java)

Antes de começar

Caso edite um alarme existente programaticamente ou criar um alarme que crie OpsItems, você deverá especificar um nome do recurso da Amazon (ARN). Este ARN identifica o OpsCenter do Systems Manager como alvo para OpsItems criados a partir do alerta. Você pode personalizar o ARN para queOpsItemscriado a partir do alarme incluem informações específicas, como gravidade ou categoria. Cada ARN inclui as informações descritas na tabela a seguir.

Parameter	Detalhes
`Region` (obrigatório)	ORegião da AWSonde o alarme existe. Por exemplo: `us-west-2` . Para obter informaçõ es sobre as Regiões da AWS onde você pode usar o OpsCenter, consulte AWS Systems Manager endpoints and quotas (Endpoints e cotas do Systems Manager).
`account_ID` (obrigatório)	O mesmoConta da AWSO ID usado para criar o alarme. Por exemplo: `123456789012` . O ID da conta deve ser seguido por dois pontos (`:`) e o parâmetro `opsitem`, conforme mostrado nos exemplos a seguir.
`severity` (obrigatório)	Um nível de gravidade definido pelo usuário para OpsItems, criado no alerta. Valores válidos: `1`, `2`, `3`, `4`
`Category` (opcional)	Uma categoria para o OpsItems criada a partir do alerta. Valores válidos: `Availability` , `Cost`, `Performance` , `Recovery` e `Security`.

Crie o ARN usando a sintaxe a seguir. Este ARN não inclui o opcional`Category`parâmetro .

```
arn:aws:ssm:Region:account_ID:opsitem:severity
```

Veja um exemplo a seguir.

```
arn:aws:ssm:us-west-2:123456789012:opsitem:3
```

Para criar um ARN que use o`Category`, use a sintaxe a seguir.

```
arn:aws:ssm:Region:account_ID:opsitem:severity#CATEGORY=category_name
```

Veja um exemplo a seguir.

```
arn:aws:ssm:us-west-2:123456789012:opsitem:3#CATEGORY=Security
```

Como configurar alarmes do CloudWatch para criar OpsItems (AWS CLI)

Este comando requer que você especifique um ARN para o parâmetro `alarm-actions`. Para obter informações sobre como criar o ARN, consulte Antes de começar.

Como configurar um alarme do CloudWatch para criar OpsItems (AWS CLI)

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.
2. Execute o seguinte comando a fim de coletar informações sobre o alarme que você deseja configurar.

   ```
   aws cloudwatch describe-alarms --alarm-names "alarm name"
   ```
3. Execute o comando a seguir para atualizar um alarme. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   ```
   aws cloudwatch put-metric-alarm --alarm-name name \
   --alarm-description "description" \
   --metric-name name --namespace namespace \
   --statistic statistic --period value --threshold value \
   --comparison-operator value \
   --dimensions "dimensions" --evaluation-periods value \
       --alarm-actions
    arn:aws:ssm:Region:account_ID:opsitem:severity#CATEGORY=category_name \
   --unit unit
   ```

   Aqui está um exemplo.

   Linux & macOS

   ```
   aws cloudwatch put-metric-alarm --alarm-name cpu-mon \
   --alarm-description "Alarm when CPU exceeds 70 percent" \
   --metric-name CPUUtilization --namespace AWS/EC2 \
   --statistic Average --period 300 --threshold 70 \
   --comparison-operator GreaterThanThreshold \
   --dimensions "Name=InstanceId,Value=i-12345678" --evaluation-periods 2 \
   ```

```
--alarm-actions arn:aws:ssm:us-east-1:123456789012:opsitem:3#CATEGORY=Security \
--unit Percent
```

Windows

```
aws cloudwatch put-metric-alarm --alarm-name cpu-mon ^
--alarm-description "Alarm when CPU exceeds 70 percent" ^
--metric-name CPUUtilization --namespace AWS/EC2 ^
--statistic Average --period 300 --threshold 70 ^
--comparison-operator GreaterThanThreshold ^
--dimensions "Name=InstanceId,Value=i-12345678" --evaluation-periods 2 ^
--alarm-actions arn:aws:ssm:us-east-1:123456789012:opsitem:3#CATEGORY=Security ^
--unit Percent
```

Como configurar alarmes do CloudWatch para criar ou atualizar OpsItems (CloudFormation)

Esta seção inclui modelos do AWS CloudFormation que você pode usar para configurar alarmes do CloudWatch para criar ou atualizar OpsItems automaticamente. Cada modelo requer que você especifique um ARN para o parâmetro `AlarmActions`. Para obter informações sobre como criar o ARN, consulte Antes de começar.

Alarme de métrica: use o modelo do CloudFormation a seguir para criar ou atualizar um alarme de métrica do CloudWatch. O alarme especificado neste modelo monitora as verificações de status da instância do Amazon Elastic Compute Cloud (Amazon EC2). Se o alarme entrar no estado `ALARM`, ele criará um OpsItem no OpsCenter.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters" : {
    "RecoveryInstance" : {
      "Description" : "The EC2 instance ID to associate this alarm with.",
      "Type" : "AWS::EC2::Instance::Id"
    }
  },
  "Resources": {
    "RecoveryTestAlarm": {
      "Type": "AWS::CloudWatch::Alarm",
      "Properties": {
```

```
            "AlarmDescription": "Run a recovery action when instance status check fails
 for 15 consecutive minutes.",
            "Namespace": "AWS/EC2" ,
            "MetricName": "StatusCheckFailed_System",
            "Statistic": "Minimum",
            "Period": "60",
            "EvaluationPeriods": "15",
            "ComparisonOperator": "GreaterThanThreshold",
            "Threshold": "0",
            "AlarmActions": [ {"Fn::Join" : ["",
 ["arn:arn:aws:ssm:Region:account_ID:opsitem:severity#CATEGORY=category_name",
 { "Ref" : "AWS::Partition" }, ":ssm:", { "Ref" : "AWS::Region" }, { "Ref" : "AWS::
 AccountId" }, ":opsitem:3" ]]} ],
            "Dimensions": [{"Name": "InstanceId","Value": {"Ref": "RecoveryInstance"}}]
          }
        }
      }
    }
```

Alarme composto: use o modelo do CloudFormation a seguir para criar ou atualizar um alarme composto. Um alarme compite consiste em vários alarmes métricos. Se o alarme entrar no estado ALARM, ele criará um OpsItem no OpsCenter.

```
"Resources":{
       "HighResourceUsage":{
          "Type":"AWS::CloudWatch::CompositeAlarm",
          "Properties":{
             "AlarmName":"HighResourceUsage",
             "AlarmRule":"(ALARM(HighCPUUsage) OR ALARM(HighMemoryUsage)) AND NOT
 ALARM(DeploymentInProgress)",

 "AlarmActions":"arn:aws:ssm:Region:account_ID:opsitem:severity#CATEGORY=category_name",
             "AlarmDescription":"Indicates that the system resource usage is high while
 no known deployment is in progress"
          },
          "DependsOn":[
             "DeploymentInProgress",
             "HighCPUUsage",
             "HighMemoryUsage"
          ]
       },
       "DeploymentInProgress":{
          "Type":"AWS::CloudWatch::CompositeAlarm",
```

```
         "Properties":{
            "AlarmName":"DeploymentInProgress",
            "AlarmRule":"FALSE",
            "AlarmDescription":"Manually updated to TRUE/FALSE to disable other
alarms"
         }
      },
      "HighCPUUsage":{
         "Type":"AWS::CloudWatch::Alarm",
         "Properties":{
            "AlarmDescription":"CPUusageishigh",
            "AlarmName":"HighCPUUsage",
            "ComparisonOperator":"GreaterThanThreshold",
            "EvaluationPeriods":1,
            "MetricName":"CPUUsage",
            "Namespace":"CustomNamespace",
            "Period":60,
            "Statistic":"Average",
            "Threshold":70,
            "TreatMissingData":"notBreaching"
         }
      },
      "HighMemoryUsage":{
         "Type":"AWS::CloudWatch::Alarm",
         "Properties":{
            "AlarmDescription":"Memoryusageishigh",
            "AlarmName":"HighMemoryUsage",
            "ComparisonOperator":"GreaterThanThreshold",
            "EvaluationPeriods":1,
            "MetricName":"MemoryUsage",
            "Namespace":"CustomNamespace",
            "Period":60,
            "Statistic":"Average",
            "Threshold":65,
            "TreatMissingData":"breaching"
         }
      }
   }
```

Como configurar alarmes do CloudWatch para criar ou atualizar OpsItems (Java)

Esta seção inclui trechos de código Java que você pode usar para configurar alarmes do CloudWatch para criar ou atualizar OpsItems automaticamente. Cada trecho requer que você

especifique um ARN para o parâmetro `validSsmActionStr`. Para obter informações sobre como criar o ARN, consulte Antes de começar.

Um alarme específico: use o trecho de código Java a seguir para criar ou atualizar um alarme do CloudWatch. O alarme especificado neste modelo monitora as verificações de status de instância do Amazon EC2. Se o alarme entrar no estado `ALARM`, ele criará um OpsItem no OpsCenter.

```
import com.amazonaws.services.cloudwatch.AmazonCloudWatch;
    import com.amazonaws.services.cloudwatch.AmazonCloudWatchClientBuilder;
    import com.amazonaws.services.cloudwatch.model.ComparisonOperator;
    import com.amazonaws.services.cloudwatch.model.Dimension;
    import com.amazonaws.services.cloudwatch.model.PutMetricAlarmRequest;
    import com.amazonaws.services.cloudwatch.model.PutMetricAlarmResult;
    import com.amazonaws.services.cloudwatch.model.StandardUnit;
    import com.amazonaws.services.cloudwatch.model.Statistic;

    private void putMetricAlarmWithSsmAction() {
        final AmazonCloudWatch cw =
                AmazonCloudWatchClientBuilder.defaultClient();

        Dimension dimension = new Dimension()
                .withName("InstanceId")
                .withValue(instanceId);

        String validSsmActionStr =
 "arn:aws:ssm:Region:account_ID:opsitem:severity#CATEGORY=category_name";

        PutMetricAlarmRequest request = new PutMetricAlarmRequest()
                .withAlarmName(alarmName)
                .withComparisonOperator(
                        ComparisonOperator.GreaterThanThreshold)
                .withEvaluationPeriods(1)
                .withMetricName("CPUUtilization")
                .withNamespace("AWS/EC2")
                .withPeriod(60)
                .withStatistic(Statistic.Average)
                .withThreshold(70.0)
                .withActionsEnabled(false)
                .withAlarmDescription(
                        "Alarm when server CPU utilization exceeds 70%")
                .withUnit(StandardUnit.Seconds)
                .withDimensions(dimension)
                .withAlarmActions(validSsmActionStr);
```

```
        PutMetricAlarmResult response = cw.putMetricAlarm(request);
    }
```

Atualizar todos os alarmes: use o trecho de código Java a seguir para atualizar todos os alarmes do CloudWatch em sua Conta da AWS para criar OpsItems quando um alarme entrar no estado `ALARM`.

```
import com.amazonaws.services.cloudwatch.AmazonCloudWatch;
    import com.amazonaws.services.cloudwatch.AmazonCloudWatchClientBuilder;
    import com.amazonaws.services.cloudwatch.model.DescribeAlarmsRequest;
    import com.amazonaws.services.cloudwatch.model.DescribeAlarmsResult;
    import com.amazonaws.services.cloudwatch.model.MetricAlarm;

    private void listMetricAlarmsAndAddSsmAction() {
        final AmazonCloudWatch cw = AmazonCloudWatchClientBuilder.defaultClient();

        boolean done = false;
        DescribeAlarmsRequest request = new DescribeAlarmsRequest();

        String validSsmActionStr =
 "arn:aws:ssm:Region:account_ID:opsitem:severity#CATEGORY=category_name";

        while(!done) {

            DescribeAlarmsResult response = cw.describeAlarms(request);

            for(MetricAlarm alarm : response.getMetricAlarms()) {
                // assuming there are no alarm actions added for the metric alarm
                alarm.setAlarmActions(ImmutableList.of(validSsmActionStr));
            }

            request.setNextToken(response.getNextToken());

            if(response.getNextToken() == null) {
                done = true;
            }
        }
    }
```

Criar OpsItems manualmente

Ao encontrar um problema operacional, é possível criar um OpsItem manualmente no OpsCenter, uma ferramenta do AWS Systems Manager, para gerenciar e resolver o problema.

Se você configurar o OpsCenter para administração entre contas, um administrador delegado do Systems Manager ou uma conta de gerenciamento do AWS Organizations poderá criar OpsItems para contas de membros. Para ter mais informações, consulte (Opcional) Configuração do OpsCenter para gerenciar OpsItems de forma centralizada entre contas.

É possível criar OpsItems usando o console do AWS Systems Manager, a AWS Command Line Interface (AWS CLI) ou as AWS Tools for Windows PowerShell.

Tópicos

- Criação de OpsItems manualmente (console)
- Criação de OpsItems manualmente (AWS CLI)
- Criação de OpsItems manualmente (PowerShell)

Criação de OpsItems manualmente (console)

É possível criar OpsItems manualmente usando o console do AWS Systems Manager. Quando você cria um OpsItem, ele é exibido em sua conta do OpsCenter. Se você configurar o OpsCenter para administração entre contas, o OpsCenter fornecerá ao administrador delegado ou à conta de gerenciamento a opção de criar OpsItems para contas de membros selecionados. Para ter mais informações, consulte (Opcional) Configuração do OpsCenter para gerenciar OpsItems de forma centralizada entre contas.

Como criar um OpsItem usando o console do AWS Systems Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha OpsCenter.
3. Escolha CriarOpsItem. Se este botão não for exibido, escolha OpsItems e, em seguida, escolha CreateOpsItem (Criar).
4. (Opcional) Escolha Outra conta e, em seguida, selecione a conta na qual você deseja criar o OpsItem.

> **Note**
>
> Esta etapa será necessária se você estiver criando OpsItems para uma conta de membro.

5. Em Title (Título), insira um nome descritivo para ajudar a compreender a finalidade do OpsItem.
6. Em Source (Origem), insira o tipo de recurso da AWS afetado ou outras informações de origem para ajudar usuários a compreender a origem do OpsItem.

> **Note**
>
> Você não poderá editar o campo Source (Origem) depois de criar a OpsItem.

7. Em Priority (Prioridade), escolha o nível de prioridade.
8. (Opcional) Em Severity (Gravidade), escolha o nível de gravidade.
9. (Opcional) Em Category (Categoria), escolha uma categoria.
10. Em Description (Descrição), insira informações sobre esse OpsItem, inclusive (se aplicáveis) etapas para reproduzir o problema.

> **Note**
>
> O console oferece suporte à maior parte da formatação de markdown no campo de descrição do OpsItem. Para obter mais informações, consulte Usar o Markdown no console no Guia de conceitos básicos do AWS Management Console.

11. Em String de eliminação de duplicação, insira palavras que o sistema pode usar para verificar se há OpsItems duplicados. Para obter mais informações sobre strings de desduplicação, consulte Gerenciamento de OpsItems duplicados.
12. (Opcional) Em Notificações, especifique o nome do recurso da Amazon (ARN) do tópico do Amazon SNS para o qual você deseja que as notificações sejam enviadas quando esse OpsItem for atualizado. Você deve especificar um ARN do Amazon SNS que esteja na mesma Região da AWS em que a OpsItem.
13. (Opcional) Em Recursos relacionados, escolha Adicionar para especificar o ID ou o ARN do recurso afetado e quaisquer recursos relacionados.
14. Escolha CriarOpsItem.

Se tiver êxito, a página exibirá o OpsItem. Quando um administrador delegado ou uma conta de gerenciamento cria um OpsItem para contas de membros selecionados, os novos OpsItems são exibidos no OpsCenter das contas de administrador e de membros. Para obter informações sobre como configurar as opções em um OpsItem, consulte Gerenciamento de OpsItems.

Criação de OpsItems manualmente (AWS CLI)

O procedimento a seguir descreve como criar um OpsItem usando a AWS Command Line Interface (AWS CLI).

Como criar um OpsItem usando a AWS CLI

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Abra a AWS CLI e execute o comando a seguir para criar uma OpsItem. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

```
aws ssm create-ops-item \
    --title "Descriptive_title" \
    --description "Information_about_the_issue" \
    --priority Number_between_1_and_5 \
    --source Source_of_the_issue \
    --operational-data Up_to_20_KB_of_data_or_path_to_JSON_file \
    --notifications Arn="SNS_ARN_in_same_Region" \
    --tags "Key=key_name,Value=a_value"
```

   Especificar dados operacionais de um arquivo

   Ao criar uma OpsItem, você pode especificar dados operacionais de um arquivo. O arquivo deve ser um arquivo JSON e o conteúdo do arquivo deve usar o formato a seguir.

```
{
  "key_name": {
    "Type": "SearchableString",
    "Value": "Up to 20 KB of data"
  }
}
```

   Aqui está um exemplo.

```
aws ssm create-ops-item ^
    --title "EC2 instance disk full" ^
    --description "Log clean up may have failed which caused the disk to be full" ^
    --priority 2 ^
    --source ec2 ^
    --operational-data file:///Users/TestUser1/Desktop/OpsItems/opsData.json ^
    --notifications Arn="arn:aws:sns:us-west-1:12345678:TestUser1" ^
    --tags "Key=EC2,Value=Production"
```

> **Note**
>
> Para obter informações sobre como inserir parâmetros formatados em JSON na linha de comando em diferentes sistemas operacionais locais, consulte Uso de aspas com strings na AWS CLI no Guia do usuário da AWS Command Line Interface.

O sistema retorna informações como estas.

```
{
    "OpsItemId": "oi-1a2b3c4d5e6f"
}
```

3. Execute o comando a seguir para visualizar detalhes sobre o OpsItem que você criou.

```
aws ssm get-ops-item --ops-item-id ID
```

O sistema retorna informações como estas.

```
{
    "OpsItem": {
        "CreatedBy": "arn:aws:iam::12345678:user/TestUser",
        "CreatedTime": 1558386334.995,
        "Description": "Log clean up may have failed which caused the disk to be
 full",
        "LastModifiedBy": "arn:aws:iam::12345678:user/TestUser",
        "LastModifiedTime": 1558386334.995,
        "Notifications": [
            {
                "Arn": "arn:aws:sns:us-west-1:12345678:TestUser"
            }
```

```
        ],
        "Priority": 2,
        "RelatedOpsItems": [],
        "Status": "Open",
        "OpsItemId": "oi-1a2b3c4d5e6f",
        "Title": "EC2 instance disk full",
        "Source": "ec2",
        "OperationalData": {
            "EC2": {
                "Value": "12345",
                "Type": "SearchableString"
            }
        }
    }
}
```

4. Execute o comando a seguir para atualizar a OpsItem. Esse comando altera o status de `Open` (o padrão) para `InProgress`.

```
aws ssm update-ops-item --ops-item-id ID --status InProgress
```

   O comando não tem uma saída.

5. Execute o comando a seguir novamente para verificar se o status foi alterado para `InProgress`.

```
aws ssm get-ops-item --ops-item-id ID
```

Exemplos de criação de um OpsItem

Os exemplos de código a seguir mostram como criar um OpsItem usando o portal de gerenciamento do Linux, macOS ou Windows.

Portal de gerenciamento do Linux ou macOS

O comando a seguir cria um OpsItem quando um disco de instância do Amazon Elastic Compute Cloud (Amazon EC2) está cheio.

```
aws ssm create-ops-item \
    --title "EC2 instance disk full" \
    --description "Log clean up may have failed which caused the disk to be full" \
    --priority 2 \
```

```
    --source ec2 \
    --operational-data '{"EC2":{"Value":"12345","Type":"SearchableString"}}' \
    --notifications Arn="arn:aws:sns:us-west-1:12345678:TestUser1" \
    --tags "Key=EC2,Value=ProductionServers"
```

O comando a seguir usa a chave `/aws/resources` em `OperationalData` para criar um OpsItem com um recurso relacionado ao Amazon DynamoDB.

```
aws ssm create-ops-item \
    --title "EC2 instance disk full" \
    --description "Log clean up may have failed which caused the disk to be full" \
    --priority 2 \
    --source ec2 \
    --operational-data '{"/aws/resources":{"Value":"[{\"arn\": \"arn:aws:dynamodb:us-
west-2:12345678:table/OpsItems\"}]","Type":"SearchableString"}}' \
    --notifications Arn="arn:aws:sns:us-west-2:12345678:TestUser"
```

O comando a seguir usa a chave `/aws/automations` em `OperationalData` para criar um OpsItem que especifica o documento `AWS-ASGEnterStandby` como um runbook do Automation associado.

```
aws ssm create-ops-item \
    --title "EC2 instance disk full" \
    --description "Log clean up may have failed which caused the disk to be full" \
    --priority 2 \
    --source ec2 \
    --operational-data '{"/aws/automations":{"Value":"[{\"automationId
\": \"AWS-ASGEnterStandby\", \"automationType\": \"AWS::SSM::Automation
\"}]","Type":"SearchableString"}}' \
    --notifications Arn="arn:aws:sns:us-west-2:12345678:TestUser"
```

Windows

O comando a seguir cria um OpsItem quando uma instância do Amazon Relational Database Service (Amazon RDS) não está respondendo.

```
aws ssm create-ops-item ^
    --title "RDS instance not responding" ^
    --description "RDS instance not responding to ping" ^
    --priority 1 ^
    --source RDS ^
    --operational-data={\"RDS\":{\"Value\":\"abcd\",\"Type\":\"SearchableString\"}} ^
```

```
    --notifications Arn="arn:aws:sns:us-west-1:12345678:TestUser1" ^
    --tags "Key=RDS,Value=ProductionServers"
```

O comando a seguir usa a chave `/aws/resources` em `OperationalData` para criar um OpsItem com um recurso relacionado à instância do Amazon EC2.

```
aws ssm create-ops-item ^
    --title "EC2 instance disk full" ^
    --description "Log clean up may have failed which caused the disk to be full" ^
    --priority 2 ^
    --source ec2 ^
    --operational-data={\"/aws/resources\":{\"Value\":\"[{\\\"""arn\\""":\
\"""arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0\\"""}]\",\"Type\":
\"SearchableString\"}}
```

O comando a seguir usa a chave `/aws/automations` em `OperationalData` para criar um OpsItem que especifica o runbook `AWS-RestartEC2Instance` como um runbook do Automation associado.

```
aws ssm create-ops-item ^
    --title "EC2 instance disk full" ^
    --description "Log clean up may have failed which caused the disk to be full" ^
    --priority 2 ^
    --source ec2 ^
    --operational-data={\"/aws/automations\":{\"Value\":\"[{\\\"""automationId\\""":
\\"""AWS-RestartEC2Instance\\”"",\\"""automationType\\""":\\"""AWS::SSM::Automation\
\"""}]\",\"Type\":\"SearchableString\"}}
```

Criação de OpsItems manualmente (PowerShell)

O procedimento a seguir descreve como criar um OpsItem usando as AWS Tools for Windows PowerShell.

Como criar um OpsItem usando as AWS Tools for Windows PowerShell

1. Abra o AWS Tools for Windows PowerShell e execute o seguinte comando para especificar suas credenciais.

   ```
   Set-AWSCredentials -AccessKey key-name -SecretKey key-name
   ```

2. Execute o comando a seguir para definir a Região da AWS para sua sessão do PowerShell.

```
Set-DefaultAWSRegion -Region Region
```

3. Execute o comando a seguir para criar um novo OpsItem. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações. Este comando especifica um runbook de automação do Systems Manager para corrigir esseOpsItem.

```
$opsItem = New-Object Amazon.SimpleSystemsManagement.Model.OpsItemDataValue
$opsItem.Type = [Amazon.SimpleSystemsManagement.OpsItemDataType]::SearchableString
$opsItem.Value = '[{\"automationId\":\"runbook_name\",\"automationType\":
\"AWS::SSM::Automation\"}]'
$newHash = @{" /aws/
automations"=[Amazon.SimpleSystemsManagement.Model.OpsItemDataValue]$opsItem}

New-SSMOpsItem `
    -Title "title" `
    -Description "description" `
    -Priority priority_number `
    -Source AWS_service `
    -OperationalData $newHash
```

Se for bem-sucedido, o comando mostrará o ID do novoOpsItem.

O exemplo a seguir especifica o nome do recurso da Amazon (ARN) de uma instância comprometida do Amazon Elastic Compute Cloud (Amazon EC2).

```
$opsItem = New-Object Amazon.SimpleSystemsManagement.Model.OpsItemDataValue
$opsItem.Type = [Amazon.SimpleSystemsManagement.OpsItemDataType]::SearchableString
$opsItem.Value = '[{\"arn\":\"arn:aws:ec2:us-east-1:123456789012:instance/
i-1234567890abcdef0\"}]'
$newHash = @{" /aws/
resources"=[Amazon.SimpleSystemsManagement.Model.OpsItemDataValue]$opsItem}
New-SSMOpsItem -Title "EC2 instance disk full still" -Description "Log clean up may
 have failed which caused the disk to be full" -Priority 2 -Source ec2 -OperationalData
 $newHash
```

Gerenciamento de OpsItems

O OpsCenter, uma ferramenta do AWS Systems Manager, acompanha OpsItems desde a criação até a resolução. Se você configurar o OpsCenter para administração entre contas, um administrador delegado ou uma conta de gerenciamento poderá gerenciar OpsItems usando a própria conta. Para

obter mais informações, consulte (Opcional) Configuração do OpsCenter para gerenciar OpsItems de forma centralizada entre contas.

É possível visualizar e gerenciar OpsItems usando as páginas a seguir no console do Systems Manager:

- Resumo: exibe a contagem de OpsItems abertos e em andamento, a contagem de OpsItems por origem e por idade e os insights operacionais. Você pode filtrar os OpsItems por origem e pelo status dos OpsItems.

- OpsItems: exibe uma lista de OpsItems com diversos campos de informações, como título, ID, prioridade, descrição, origem do OpsItem e data e horário da última atualização. Ao usar esta página, é possível criar OpsItems manualmente, configurar origens, alterar o status de um OpsItem e filtrar OpsItems por novos incidentes. Você pode escolher um OpsItem para exibir a página Detalhes de OpsItems dele.

- Detalhes do OpsItem: fornece insights e ferramentas detalhadas que você pode usar para gerenciar um OpsItem. A página de detalhes de OpsItems tem as guias a seguir:
  - Visão geral: exibe recursos relacionados, runbooks executados nos últimos 30 dias e uma lista de runbooks disponíveis que você pode executar. Também é possível exibir OpsItems semelhantes, adicionar dados operacionais e adicionar OpsItems relacionados.
  - Detalhes de recursos relacionados: exibe informações sobre o recurso de diversos serviços da AWS. Expanda a seção Resource details (Detalhes do recurso) para visualizar informações sobre esse recurso, conforme fornecido pelo serviço da AWS que o hospeda. Você também pode alternar entre outros recursos relacionados associados a esse OpsItem usando a lista Related resources (Recursos relacionados).

Para obter mais informações sobre como gerenciar OpsItems, consulte os tópicos a seguir.

Tópicos

- Visualização de detalhes de um OpsItem
- Editar um OpsItem
- Adição de recursos relacionados para um OpsItem
- Adição de OpsItems relacionados para um OpsItem
- Adição de dados operacionais a um OpsItem

- Criação de um incidente para um OpsItem
- Gerenciamento de OpsItems duplicados
- Analisar insights operacionais para reduzir OpsItems
- Visualização de logs e relatórios do OpsCenter

Visualização de detalhes de um OpsItem

Para obter uma visualização abrangente de um OpsItem, use a página Detalhes do OpsItem no console do OpsCenter. A página Visão geral exibe as informações a seguir:

- Detalhes de OpsItems: exibe informações gerais para o OpsItem selecionado.
- Recursos relacionados: um recurso relacionado corresponde ao recurso afetado ou ao recurso que iniciou o evento que criou o OpsItem.
- Execuções do Automation nos últimos 30 dias: uma lista de runbooks executados nos últimos 30 dias.
- Runbooks: é possível escolher um runbook em uma lista de runbooks disponíveis.
- OpsItems semelhantes: esta é uma lista gerada pelo sistema de OpsItems que podem estar relacionados ou ser de seu interesse. Para gerar a lista, o sistema examina os títulos e as descrições de todas as OpsItems e retorna OpsItems que usam as palavras semelhantes.
- Dados operacionais: dados operacionais são dados personalizados que fornecem detalhes de referência úteis sobre o OpsItem. Por exemplo, você pode especificar arquivos de log, strings de erro, chaves de licença, dicas para solução de problemas ou outros dados relevantes.
- OpsItems relacionados: é possível especificar os IDs de OpsItems que estão de alguma forma relacionados ao OpsItem atual.
- Detalhes de recursos relacionados: exibe provedores de dados, incluindo as métricas e os alarmes do Amazon CloudWatch, os logs do AWS CloudTrail e detalhes do AWS Config.

Para visualizar detalhes de uma OpsItem

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha OpsCenter.
3. Escolha um OpsItem para visualizar seus detalhes.

Editar um OpsItem

A seção Detalhes do OpsItem inclui informações sobre um OpsItem, incluindo a descrição, o título, a origem, o ID do OpsItem e o status. É possível editar um único OpsItem ou selecionar diversos OpsItems e editar os seguintes campos: Status, Prioridade, Severidade e Categoria.

Quando o Amazon EventBridge cria um OpsItem, ele preenche os campos Título, Origem e Descrição. É possível editar os campos Título e Descrição, mas não o campo Origem.

> **Note**
>
> O console oferece suporte à maior parte da formatação de markdown no campo de descrição do OpsItem. Para obter mais informações, consulte Usar o Markdown no console no Guia de conceitos básicos do AWS Management Console.

Geralmente, é possível editar os dados configuráveis a seguir para um OpsItem:

- Título: nome do OpsItem. A origem cria o título do OpsItem.
- Descrição: informações sobre esse OpsItem, inclusive (se aplicáveis) etapas para reproduzir o problema.
- Status: o status de um OpsItem. Para ver uma lista de valores de status válidos, consulte Status do OpsItem na Referência da API do AWS Systems Manager.
- Prioridade: a prioridade de um OpsItem pode estar entre 1 e 5. Recomendamos que sua organização determine o significado de cada nível de prioridade e faça um acordo de nível de serviço correspondente para cada nível.
- Severidade: a severidade de um OpsItem pode estar entre 1 e 4. 1 significa crítica, 2 significa alta, 3 significa média e 4 significa baixa.
- Categoria: a categoria de um OpsItem pode ser disponibilidade, custo, performance, recuperação ou segurança.
- Notificações: ao editar um OpsItem, é possível especificar o nome do recurso da Amazon (ARN) de um tópico do Amazon Simple Notification Service no campo Notificações. Ao especificar um ARN, você garante que todas as partes interessadas recebam uma notificação quando a OpsItem é editada, inclusive uma alteração de status. Para obter mais informações, consulte o Manual do desenvolvedor do Amazon Simple Notification Service.

> **Important**
>
> O tópico do Amazon SNS deve existir na mesma Região da AWS que o OpsItem. Se o tópico e o OpsItem estiverem em regiões diferentes, o sistema retornará um erro.

O OpsCenter tem integração bidirecional com o AWS Security Hub. Quando você atualiza o status e a severidade de um OpsItem relacionado a uma descoberta de segurança, as alterações são enviadas automaticamente ao Security Hub para garantir que você sempre veja as informações mais recentes e corretas.

Quando um OpsItem é criado a partir de uma descoberta do Security Hub, os metadados do Security Hub são adicionados automaticamente ao campo de dados operacionais do OpsItem. Se esses metadados forem excluídos, as atualizações bidirecionais não funcionarão mais.

Para editar detalhes da OpsItem

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha OpsCenter.
3. Escolha um ID do OpsItem para abrir a página de detalhes ou selecione vários OpsItems. Se você escolher vários OpsItems, você só poderá editar o status, a prioridade, a gravidade ou a categoria. Se você editar vários OpsItems, o OpsCenter atualiza e salva suas alterações assim que você escolher o novo status, prioridade, gravidade ou categoria.
4. Na seção OpsItem details (Detalhes de OpsItem), escolha Edit (Editar).
5. Edite os detalhes da OpsItem de acordo com os requisitos e as diretrizes especificados pela organização.
6. Ao terminar, escolha Salvar.

Adição de recursos relacionados para um OpsItem

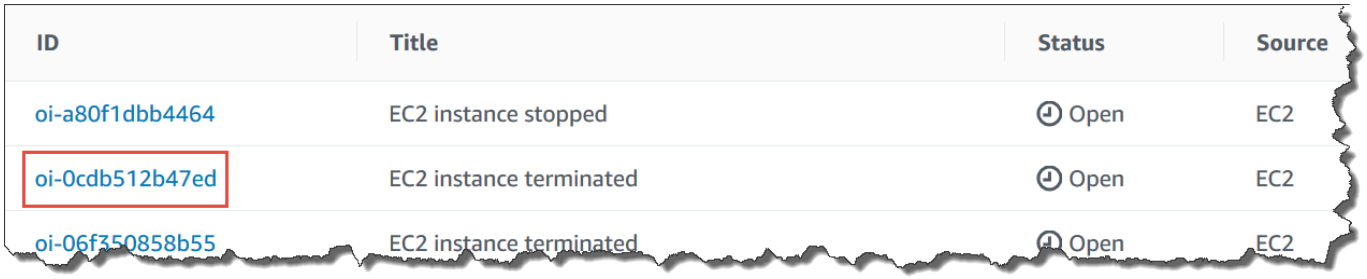
Cada OpsItem inclui uma seção chamada Recursos relacionados que lista o nome do recurso da Amazon (ARN) para o recurso relacionado. Um recurso relacionado corresponde ao recurso da AWS afetado que precisa ser investigado.

Se o Amazon EventBridge criar o OpsItem, o sistema preencherá automaticamente o OpsItem com o ARN do recurso. É possível especificar manualmente os ARNs para os recursos relacionados. Para certos tipos de ARN, o OpsCenter cria automaticamente um link direto que exibe detalhes sobre

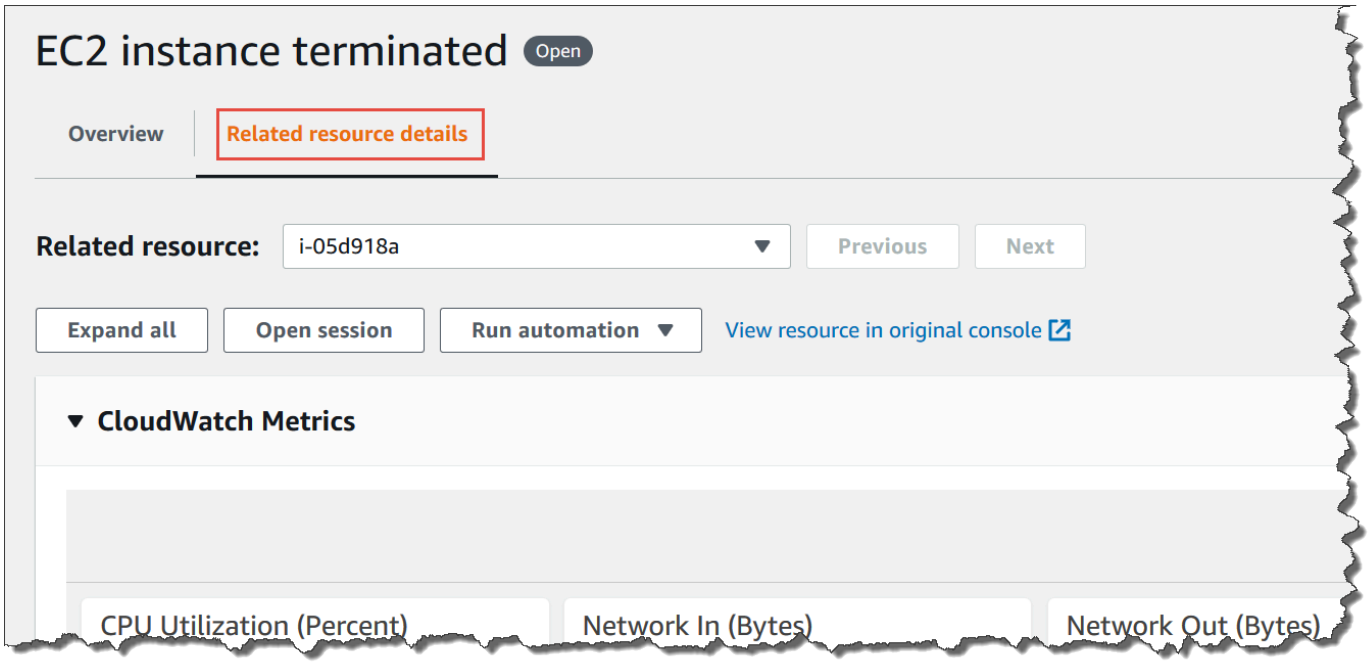
o recurso diretamente no console do OpsCenter. Por exemplo, se você especificar o ARN de uma instância do Amazon Elastic Compute Cloud (Amazon EC2) como recurso relacionado, o OpsCenter extrai detalhes sobre essa instância do EC2. Isso permite visualizar informações detalhadas sobre os recursos da AWS afetados sem a necessidade de sair do OpsCenter.

Como visualizar e adicionar recursos relacionados para um OpsItem

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha OpsCenter.
3. Escolha a guia OpsItems.
4. Escolha um ID da OpsItem.



5. Para visualizar informações sobre o recurso afetado, escolha a guia Related resources details (Detalhes de recursos relacionados).



Essa guia exibe informações sobre o recurso de vários Serviços da AWS. Expanda a seção Resource details (Detalhes do recurso) para visualizar informações sobre esse recurso,

conforme fornecido pelo AWS service (Serviço da AWS) que o hospeda. Você também pode alternar entre outros recursos relacionados associados a esse OpsItem usando a lista Related resources (Recursos relacionados).

6. Para adicionar recursos relacionados adicionais, escolha a guia Overview (Visão geral).
7. Na seção Related resources (Recursos relacionados), escolha Add (Adicionar).
8. Para Resource type (Tipo de recurso), escolha um recurso da lista.
9. Para Resource ID (ID do recurso), insira o ID ou o nome do recurso da Amazon (ARN). O tipo de informação que você escolher depende do recurso escolhido na etapa anterior.

> **Note**
>
> Você pode adicionar manualmente os ARNs de recursos relacionados adicionais. Cada OpsItem pode listar, no máximo, 100 ARNs de recursos relacionados.

A tabela a seguir lista os tipos de recursos que criam links diretos de forma automática para os recursos relacionados.

Tipos de recursos compatíveis

Nome do recurso	Formato ARN
Certificado AWS Certificate Manager	`arn:aws:acm: region:account-id :certificate/ certificate-id`
Grupo do Amazon EC2 Auto Scaling	`arn:aws:autoscaling: region:account-id :autoScalingGroup: groupid:autoScalingGroupName/ groupfriendlyname`
Distribuição do Amazon CloudFront	`arn:aws:cloudfront:: account-id :*`
Pilha AWS CloudFormation	`arn:aws:cloudformation: region:account-id :stack/stackname /additionalidentifier`

Nome do recurso	Formato ARN
Alarmes do Amazon CloudWatch	`arn:aws:cloudwatch: region:account-id :alarm:alarm-name`
Trilha AWS CloudTrail	`arn:aws:cloudtrail: region:account-id :trail/trailname`
projeto da AWS CodeBuild	`arn:aws:codebuild: region:account-id :resourcetype /resource`
AWS CodePipeline	`arn:aws:codepipeline: region:account-id :resource-specifier`
Insight do Amazon DevOps Guru	`arn:aws:devops-guru: region:account-id :insight/ proactive or reactive/resource-id`
Tabela do Amazon DynamoDB	`arn:aws:dynamodb: region:account-id :table/tablename`
Gateway do cliente do Amazon Elastic Compute Cloud (Amazon EC2)	`arn:aws:ec2: region:account-id :customer-gateway/ cgw-id`
IP elástico do Amazon EC2	`arn:aws:ec2: region:account-id :eip/eipalloc-id`
Host dedicado do Amazon EC2	`arn:aws:ec2: region:account-id :dedicated-host/ host-id`
Instância do Amazon EC2	`arn:aws:ec2: region:account-id :instance/ instance-id`

Nome do recurso	Formato ARN
Gateway de Internet do Amazon EC2	`arn:aws:ec2: region:account-id :internet-gateway/ igw-id`
Lista de controle de acesso à rede (ACL da rede) do Amazon EC2	`arn:aws:ec2: region:account-id :network-acl/ nacl-id`
Interface de rede do Amazon EC2	`arn:aws:ec2: region:account-id :network-interface/ eni-id`
Tabela de rotas do Amazon EC2	`arn:aws:ec2: region:account-id :route-table/ route-table-id`
Grupo de segurança do Amazon EC2	`arn:aws:ec2: region:account-id :security-group/ security-group-id`
Sub-rede do Amazon EC2	`arn:aws:ec2: region:account-id :subnet/subnet-id`
Volume do Amazon EC2	`arn:aws:ec2: region:account-id :volume/volume-id`
VPC do Amazon EC2	`arn:aws:ec2: region:account-id :vpc/vpc-id`
Conexão da VPN do Amazon EC2	`arn:aws:ec2: region:account-id :vpn-connection/ vpn-id`
Gateway da VPN do Amazon EC2	`arn:aws:ec2: region:account-id :vpn-gateway/ vgw-id`

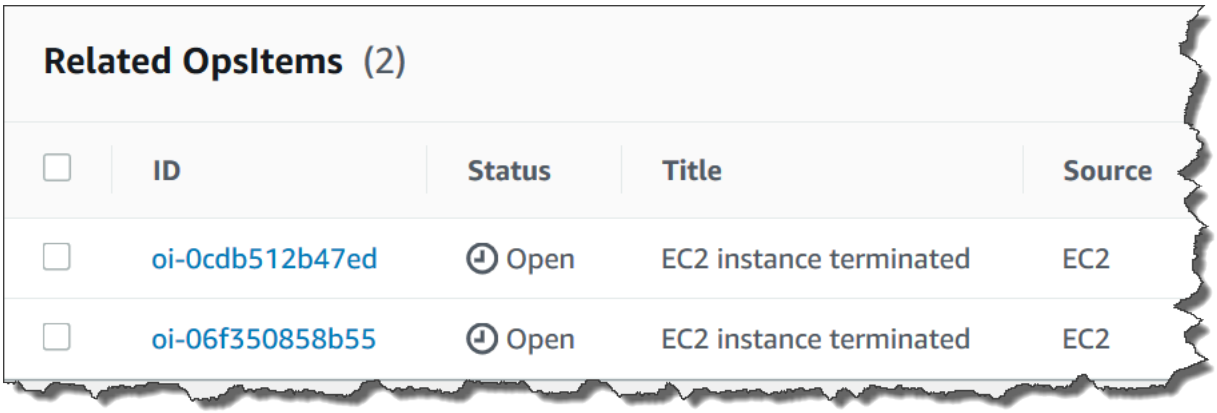
Nome do recurso	Formato ARN
Aplicação do AWS Elastic Beanstalk	`arn:aws:elasticbeanstalk: region:account-id :application/ applicationname`
Elastic Load Balancing (Classic Load Balancer)	`arn:aws:elasticloadbalancing: region:account-id :loadbalancer/ name`
Elastic Load Balancing (Application Load Balancer)	`arn:aws:elasticloadbalancing: region:account-id :loadbalancer/app/ load-balancer-name /load-balancer-id`
Elastic Load Balancing (Network Load Balancer)	`arn:aws:elasticloadbalancing: region:account-id :loadbalancer/net/ load-balancer-name /load-balancer-id`
AWS Identity and Access Management Grupo do IAM	`arn:aws:iam:: account-id :group/group-name`
Política do IAM	`arn:aws:iam:: account-id :policy/policy-name`
IAM role (Perfil do IAM)	`arn:aws:iam:: account-id :role/role-name`
IAM user (Usuário do IAM)	`arn:aws:iam:: account-id :user/user-name`
Função do AWS Lambda	`arn:aws:lambda: region:account-id :function: function-name`

Nome do recurso	Formato ARN
Cluster do Amazon Relational Database Service (Amazon RDS)	`arn:aws:rds: region:account-id :cluster: db-cluster-name`
Instância de bancos de dados do Amazon RDS	`arn:aws:rds: region:account-id :db:db-instance-name`
Assinatura do Amazon RDS	`arn:aws:rds: region:account-id :es:subscription-name`
Grupos de segurança do Amazon RDS	`arn:aws:rds: region:account-id :secgrp:security-group-name`
Snapshot do cluster do Amazon RDS	`arn:aws:rds: region:account-id :cluster-snapshot: cluster-snapshot-name`
Grupo de sub-rede do Amazon RDS	`arn:aws:rds: region:account-id :subgrp:subnet-group-name`
Cluster do Amazon Redshift	`arn:aws:redshift: region:account-id :cluster: cluster-name`
Grupos de parâmetros do Amazon Redshift	`arn:aws:redshift: region:account-id :parametergroup: parameter-group-name`
Grupo de segurança do Amazon Redshift	`arn:aws:redshift: region:account-id :securitygroup: security-group-name`

Nome do recurso	Formato ARN
Snapshots do cluster do Amazon Redshift	`arn:aws:redshift:region:account-id:snapshot:cluster-name/snapshot-name`
Grupos de sub-rede do cluster do Amazon Redshift	`arn:aws:redshift:region:account-id:subnetgroup:subnet-group-name`
Bucket do Amazon Simple Storage Service (Amazon S3)	`arn:aws:s3:::bucket_name`
Gravação AWS Config do inventário do nó gerenciado do AWS Systems Manager	`arn:aws:ssm:region:account-id:managed-instance-inventory/node_id`
Systems Manager (Gerenciador de sistemas) State ManagerAssociação	`arn:aws:ssm:region:account-id:association/association_ID`

Adição de OpsItems relacionados para um OpsItem

Ao usar OpsItems relacionados da página Detalhes de OpsItems, é possível investigar problemas de operações e fornecer contexto para um problema. Os OpsItems podem ser relacionados de diferentes maneiras, incluindo uma relação pai/filho entre OpsItems, uma causa-raiz ou uma duplicata. É possível associar um OpsItem a outro para exibi-lo na seção OpsItem relacionado. Você pode especificar, no máximo, dez IDs para outros OpsItems relacionados ao OpsItem atual.

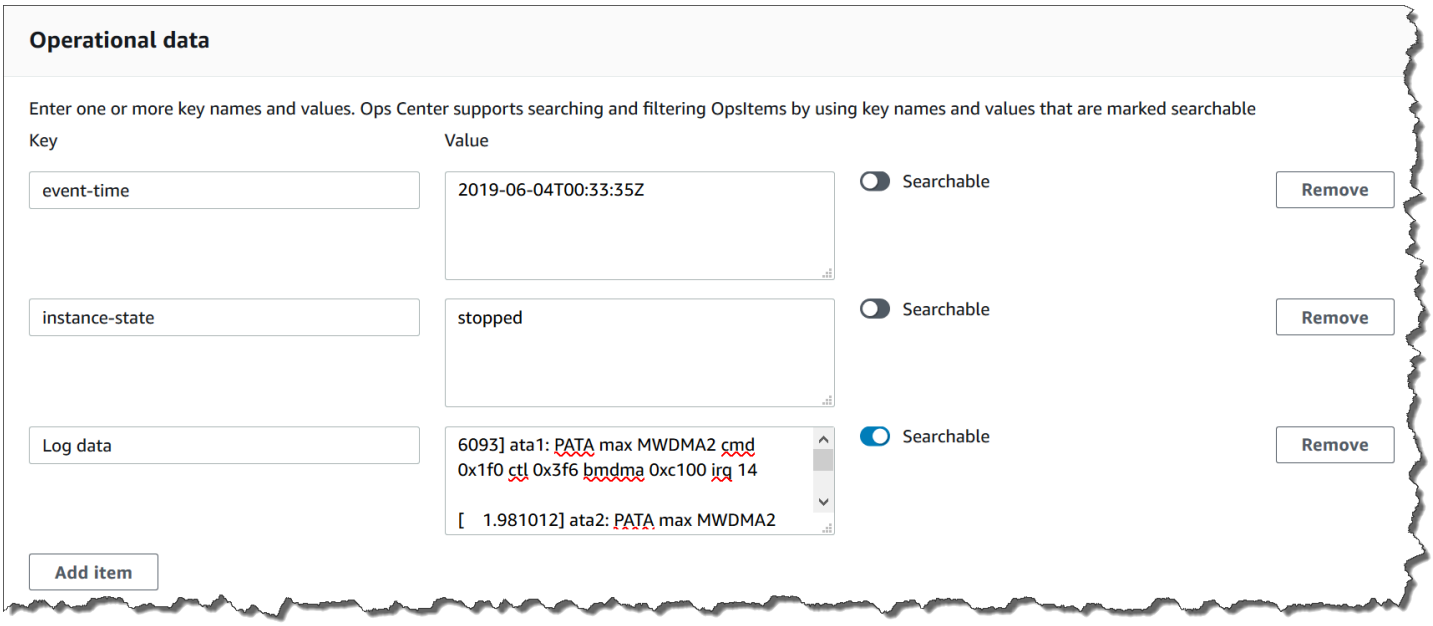


Para adicionar um relacionadoOpsItem

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha OpsCenter.
3. Escolha um ID da OpsItem para abrir a página de detalhes.
4. Na seção Related OpsItem (OI relacionada), escolha Add (Adicionar).
5. Em OpsItem ID (ID da OI), especifique um ID.
6. Escolha Adicionar.

Adição de dados operacionais a um OpsItem

Dados operacionais são dados personalizados que fornecem detalhes de referência úteis sobre um OpsItem. Você pode inserir vários pares de chave-valor de dados operacionais. Por exemplo, você pode especificar arquivos de log, strings de erro, chaves de licença, dicas para solução de problemas ou outros dados relevantes. O comprimento máximo da chave é 128 caracteres e o tamanho máximo do valor é 20 KB.



É possível tornar os dados pesquisáveis por outros usuários na conta ou restringir o acesso à pesquisa. Dados pesquisáveis significam que todos os usuários com acesso à página Visão geral do OpsItem (conforme fornecido pela operação de API DescribeOpsItems) podem visualizar e pesquisar os dados especificados. Dados operacionais que não são pesquisáveis só podem ser visualizados por usuários que tenham acesso ao OpsItem (conforme fornecido pela operação de API GetOpsItem).

Para adicionar dados operacionais a uma OpsItem

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha OpsCenter.
3. Escolha um ID do OpsItem para abrir a página de detalhes.
4. Expanda Dados operacionais.
5. Se não houver dados operacionais para o OpsItem, escolha Adicionar. Se já houver dados operacionais para a OpsItem, escolha Manage (Gerenciar).

   Depois de criar dados operacionais, você poderá editar a chave e o valor, remover os dados operacionais ou adicionar pares de chave-valor adicionais escolhendo Manage (Gerenciar).
6. Em Key (Chave), especifique uma palavra ou palavras para ajudar usuários a entender a finalidade dos dados.

   > **Important**
   >
   > Chaves de dados operacionaisnão podeComece com o seguinte:`amazon`,`aws`,`amzn`,`ssm`,`/amazon`,`/aws`,`/amzn`,`/ssm`.
7. Em Value (Valor), especifique os dados.
8. Escolha Salvar.

> **Note**
>
> Você pode filtrar OpsItems usando o operador Operational data (Dados operacionais) na página do OpsItems. Na caixa Pesquisar, escolha Dados operacionais e, em seguida, insira um par chave-valor em JSON. É necessário inserir o par de chave-valor usando o seguinte formato: `{"key":"key_name","value":"a_value"}`

Criação de um incidente para um OpsItem

Use o procedimento a seguir para criar um incidente para um OpsItem manualmente com a finalidade de acompanhá-lo e gerenciá-lo no AWS Systems Manager Incident Manager, que é uma ferramenta do AWS Systems Manager. Um incidente é qualquer interrupção ou redução não planejada na qualidade dos serviços. Para obter mais informações sobre o Incident Manager, consulte the section called “Integrar o OpsCenter a outros Serviços da AWS”.

Para criar manualmente um incidente para umaOpsItem

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha OpsCenter.
3. Se o Incident Manager criou um OpsItem para você, escolha-o e vá para o passo 5. Caso contrário, escolha Create OpsItem (Criar OpsItem) e preencha o formulário. Se este botão não for exibido, escolha OpsItems e, em seguida, escolha CreateOpsItem (Criar).
4. Se você criou um OpsItem, abra-o.
5. Selecione Start Incident (Iniciar incidente).
6. Em Plano de resposta, escolha o plano de resposta do Incident Manager que você deseja atribuir a este incidente.
7. (Opcional) Em Title (Título), insira um nome descritivo para ajudar outros membros da equipe a compreender a natureza do incidente. Se você não inserir um novo título, o OpsCenter cria o OpsItem e o incidente correspondente no Incident Manager usando o título no plano de resposta.
8. (Opcional) Em Incident impact (Impacto do incidente), escolha um nível de impacto para este incidente. Se você não escolher um nível de impacto, OpsCenter cria o OpsItem e o incidente correspondente no Incident Manager usando o nível de impacto no plano de resposta.
9. Escolha Iniciar.

Gerenciamento de OpsItems duplicados

O OpsCenter pode receber diversos OpsItems duplicados para uma única origem de diversos Serviços da AWS. O OpsCenter usa uma combinação de lógica integrada e strings de eliminação de duplicação configuráveis para evitar a criação de OpsItems duplicados. O AWS Systems Manager aplica a lógica integrada de eliminação de duplicação quando a operação de API CreateOpsItem é chamada.

O AWS Systems Manager usa a seguinte lógica de eliminação de duplicação:

1. Ao criar o OpsItem, o Systems Manager cria e armazena um hash baseado na string de desduplicação e no recurso que iniciou o OpsItem.
2. Quando outra solicitação é realizada para criar um OpsItem, o sistema verifica a string de eliminação de duplicação da nova solicitação.
3. Se existir um hash correspondente para a string de eliminação de duplicação, o Systems Manager verificará o status do OpsItem existente. Se o status de um OpsItem existente for aberto ou em

andamento, o OpsItem não será criado. Se o OpsItem existente estiver como resolvido, o Systems Manager criará um novo OpsItem.

Depois de criar um OpsItem, não será possível editar ou alterar as strings de desduplicação nesse OpsItem.

Para gerenciar OpsItems duplicados, é possível fazer o seguinte:

- Editar a string de eliminação de duplicação para uma regra do Amazon EventBridge direcionada ao OpsCenter. Para ter mais informações, consulte Como editar uma string de eliminação de duplicação em uma regra padrão do EventBridge.
- Especificar uma string de eliminação de duplicação ao criar um OpsItem manualmente. Para ter mais informações, consulte Como especificar uma string de eliminação de duplicação usando a AWS CLI.
- Analisar e resolver OpsItems duplicados usando insights operacionais. É possível usar runbooks para resolver OpsItems duplicados.

  Para ajudar você a resolver OpsItems duplicados e reduzir o número de OpsItems criados por uma origem, o Systems Manager fornece runbooks de automação. Para ter mais informações, consulte Resolver duplicadosOpsItemsCom base em insights.

Como editar uma string de eliminação de duplicação em uma regra padrão do EventBridge

Use o procedimento a seguir a fim de especificar uma string de desduplicação para uma regra do EventBridge que segmente OpsCenter.

Como editar uma string de eliminação de duplicação para uma regra do EventBridge

1. Faça login noAWS Management Consolee abra o console do Amazon EventBridge emhttps://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Regras.
3. Escolha a regra e Edit (Editar).
4. Acesse a página Select target(s) (Selecionar destinos).
5. Na seção Additional settings (Configurações adicionais), escolha Configure input transformer (Configurar transformador de entrada).
6. Na caixa Template (Modelo), localize a entrada JSON `"operationalData": { "/aws/dedup"` e as strings de desduplicação que você deseja editar.

A entrada da string de desduplicação em regras do EventBridge usa o formato JSON a seguir.

```
"operationalData": { "/aws/dedup": {"type": "SearchableString","value":
 "{\"dedupString\":\"Words the system should use to check for duplicate
 OpsItems\"}"}}
```

Aqui está um exemplo.

```
"operationalData": { "/aws/dedup": {"type": "SearchableString","value":
 "{\"dedupString\":\"SSMOpsCenter-EBS-volume-performance-issue\"}"}}
```

7. Edite as strings de eliminação de duplicação e, em seguida, escolha Confirmar.
8. Escolha Próximo.
9. Escolha Próximo.
10. Escolha Upgrade rule (Atualizar regra).

Como especificar uma string de eliminação de duplicação usando a AWS CLI

É possível especificar uma string de eliminação de duplicação ao criar um novo OpsItem manualmente usando o console do AWS Systems Manager ou a AWS CLI. Para obter informações sobre como inserir strings de desduplicação ao criar manualmente um OpsItem no console, consulte Criar OpsItems manualmente. Caso esteja usando a AWS CLI, você pode inserir a string de eliminação de duplicação para o parâmetro `OperationalData`. A sintaxe do parâmetro usa JSON, conforme mostrado no exemplo a seguir.

```
--operational-data '{"/aws/dedup":{"Value":"{\"dedupString\": \"Words the system should
 use to check for duplicate OpsItems\"}","Type":"SearchableString"}}'
```

Aqui está um comando de exemplo que especifica uma string de desduplicação de `disk full`.

Linux & macOS

```
aws ssm create-ops-item \
    --title "EC2 instance disk full" \
    --description "Log clean up may have failed which caused the disk to be full" \
    --priority 1 \
    --source ec2 \
```

```
    --operational-data '{"/aws/dedup":{"Value":"{\"dedupString\": \"disk full
\"}","Type":"SearchableString"}}' \
    --tags "Key=EC2,Value=ProductionServers" \
    --notifications Arn="arn:aws:sns:us-west-1:12345678:TestUser"
```

Windows

```
aws ssm create-ops-item ^
    --title "EC2 instance disk full" ^
    --description "Log clean up may have failed which caused the disk to be full" ^
    --priority 1 ^
    --source EC2 ^
    --operational-data={\"/aws/dedup\":{\"Value\":\"{\\"""dedupString\\""":\\"""disk
 full\\"""}\",\"Type\":\"SearchableString\"}} ^
    --tags "Key=EC2,Value=ProductionServers" --notifications Arn="arn:aws:sns:us-
west-1:12345678:TestUser"
```

Analisar insights operacionais para reduzir OpsItems

Os insights operacionais do OpsCenter exibem informações sobre OpsItems duplicados. O OpsCenter analisa automaticamente os OpsItems da conta e gera três tipos de insights. Você pode visualizar essas informações na seção Insights operacionais da guia Resumo do OpsCenter.

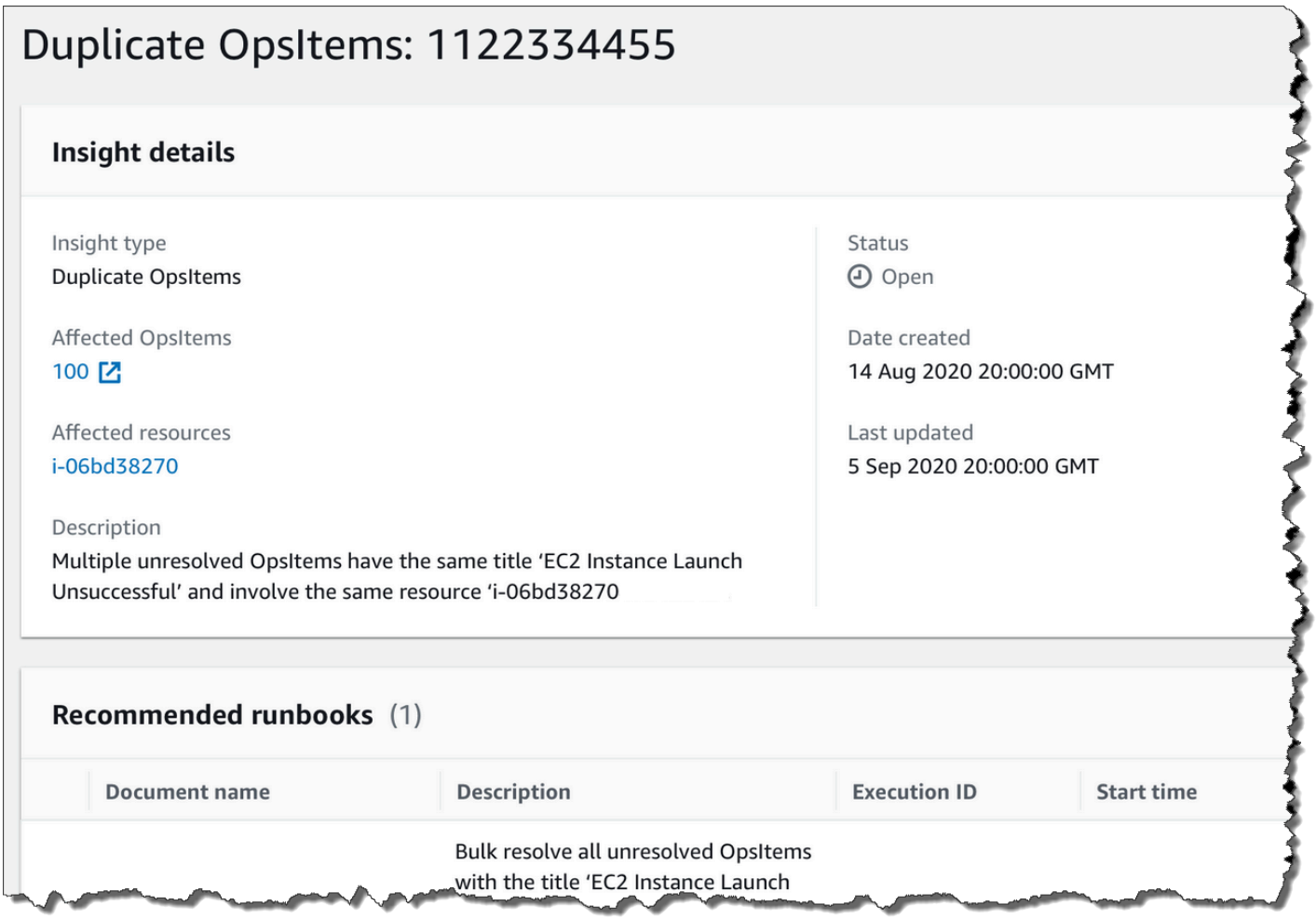
- OpsItems duplicados: um insight é gerado quando oito ou mais OpsItems têm o mesmo título para o mesmo recurso.
- Títulos mais comuns: um insight é gerado quando mais de 50 OpsItems têm o mesmo título.
- Recursos que geram mais OpsItems: um insight é gerado quando um recurso da AWS tem mais de dez OpsItems abertos. Esses insights e os recursos correspondentes são exibidos na tabela Recursos que geram mais OpsItems na guia Resumo do OpsCenter. Os recursos são listados em ordem decrescente da contagem de OpsItem.

> **Note**
>
> O OpsCenter cria insights de recursos que geram mais OpsItems para os seguintes tipos de recursos:
>
> - Instâncias do Amazon Elastic Compute Cloud (Amazon EC2)
> - Grupos de segurança do Amazon EC2

- Grupo do Amazon EC2 Auto Scaling
- Banco de dados do Amazon Relational Database Service (Amazon RDS)
- Cluster do Amazon RDS
- Função do AWS Lambda
- Tabela do Amazon DynamoDB
- Balanceador de carga do Elastic Load Balancing
- Cluster do Amazon Redshift
- Certificado AWS Certificate Manager
- Volume do Amazon Elastic Block Store

O OpsCenter aplica um limite de 15 insights por tipo. Se um tipo atingir esse limite, o OpsCenter interrompe a exibição de mais insights desse tipo. Para visualizar outros insights, é necessário resolver todos os OpsItems associados a um OpsInsight daquele tipo. Se um insight pendente for impedido de ser exibido no console por causa do limite de 15 insights, ele ficará visível depois que outro insight for fechado.

Quando você escolhe um insight, o OpsCenter exibe informações sobre os OpsItems e os recursos afetados. A captura de tela a seguir mostra um exemplo com os detalhes de um insight de OpsItem duplicado.



Os insights operacionais estão desativados por padrão. Para obter mais informações sobre como trabalhar com os insights operacionais, consulte os tópicos a seguir.

Tópicos

- Habilitar insights operacionais
- Resolver duplicadosOpsItemsCom base em insights
- Desativar insights operacionais

Habilitar insights operacionais

É possível ativar os insights operacionais na página OpsCenter do console do Systems Manager. Quando você habilita os insights operacionais, o Systems Manager cria um perfil vinculado ao serviço do AWS Identity and Access Management (IAM), chamado `AWSServiceRoleForAmazonSSM_OpsInsights`. Uma função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente ao Systems Manager. Os perfis vinculados ao

serviço são definidos previamente e incluem todas as permissões que o serviço requer para chamar outros Serviços da AWS em seu nome. Para obter mais informações sobre a função vinculada ao serviço do `AWSServiceRoleForAmazonSSM_OpsInsights`, consulte Usar perfis para criar insights operacionais de OpsItems no OpsCenter do Systems Manager.

> **Note**
>
> Observe as seguintes informações importantes:
>
> - Sua Conta da AWS será cobrada pelos insights operacionais. Para obter mais informações, consulte Preços do AWS Systems Manager.
> - O OpsCenter atualiza periodicamente os insights usando um processo em lote. Isso significa que a lista de insights exibida no OpsCenter pode não estar sincronizada.

Use o procedimento a seguir para habilitar e visualizar insights operacionais no OpsCenter.

Como habilitar e visualizar insights operacionais

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha OpsCenter.
3. Na caixa de mensagem Insight operacional disponível, escolha Habilitar. Caso não veja essa mensagem, role para baixo até a seção Insights operacionais e escolha Habilitar.
4. Depois de habilitar esse atributo, na guia Resumo, role para baixo até a seção Insights operacionais.
5. Para visualizar uma lista filtrada de insights, escolha o link ao lado de Duplicar OpsItems, Títulos mais comuns ou Recursos que geram mais OpsItems. Para visualizar todos os insights, escolha View all operational insights (Visualizar todas as informações operacionais).
6. Escolha um ID de insight para visualizar mais informações.

Resolver duplicadosOpsItemsCom base em insights

Para resolver insights, resolva primeiro todos os OpsItems associados a uma visão. Você pode usar o runbook `AWS-BulkResolveOpsItemsForInsight` para resolver o OpsItems associado a um insight.

Para ajudar você a resolver OpsItems duplicados e reduzir o número de OpsItems criados por uma origem, o Systems Manager fornece os runbooks de automação a seguir:

- O`AWS-BulkResolveOpsItems`runbook resolveOpsItemsque correspondem a um filtro especificado.
- O runbook `AWS-AddOpsItemDedupStringToEventBridgeRule` adiciona uma string de eliminação de duplicação para todos os destinos do OpsItem que estão associados a uma regra específica do Amazon EventBridge. Este runbook não adicionará uma string de eliminação de duplicação se a regra já tiver uma.
- O `AWS-DisableEventBridgeRule` desativará uma regra no EventBridge se a regra estiver gerando dezenas ou centenas de OpsItems.

Para resolver um insight operacional

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha OpsCenter.
3. NoVisão geral, role para baixo atéInsights operacionais.
4. Escolha Visualizar todos os insights operacionais.
5. Escolha um ID de insight para visualizar mais informações.
6. Escolha um runbook e escolha Executar.

Desativar insights operacionais

Quando você desativa os insights operacionais, o sistema para de criar novos insights e de exibir insights no console. Quaisquer insights ativos permanecem inalterados no sistema, embora eles não sejam exibidos no console. Se você habilitar esse recurso novamente, o sistema exibirá insights não resolvidos anteriormente e começará a criar novos insights. Use o procedimento a seguir para desativar os insights operacionais.

Como desativar insights operacionais

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha OpsCenter.
3. Escolha Configurações.
4. NoInsights operacionais, selecioneEditee, em seguida, alterne oDesabilitar oopção.

5. Escolha Salvar.

Visualização de logs e relatórios do OpsCenter

O AWS CloudTrail registra em log as chamadas de API do OpsCenter para o AWS Systems Manager no console, na AWS Command Line Interface (AWS CLI) e no SDK. Você pode visualizar as informações no console do CloudTrail ou em um bucket do Amazon Simple Storage Service (Amazon S3). O Amazon S3 usa um bucket para armazenar todos os logs do CloudTrail para sua conta.

Logs de ações OpsCenter mostram atividades de criação, atualização, obtenção e descrição da OpsItem. Para obter mais informações sobre como visualizar e utilizar os logs do CloudTrail de atividades do Systems Manager, consulte Registrar em log chamadas de API do AWS Systems Manager com o AWS CloudTrail.

O OpsCenter do AWS Systems Manager fornece as informações a seguir sobre os OpsItems:

- Resumo por status do OpsItem: fornece um resumo dos OpsItems por status (Aberto e Em andamento, Aberto ou Em andamento).
- Origens com a maioria dos OpsItems abertos: fornece um detalhamento dos principais Serviços da AWS com OpsItems abertos.
- OpsItems por origem e por idade: fornece uma contagem de OpsItems, agrupados com base na origem e na quantidade de dias desde a criação.

Como visualizar o relatório resumido do OpsCenter

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha OpsCenter.
3. Na página Visão geral dos OpsItems, escolha Resumo.
4. Em OpsItems by source and age (OpsItems por origem e idade), escolha a barra de pesquisa para filtrar OpsItems de acordo com Source (Origem). Use a lista para filtrar de acordo com Status.

Exclua OpsItems

Você pode excluir uma pessoa OpsItem chamando a operação da OpsItem API Delete usando AWS Command Line Interface ou AWS SDK. Você não pode excluir um OpsItem no AWS Management

Console. Para excluir um OpsItem, seu usuário (IAM) AWS Identity and Access Management, grupo ou função deve ter permissão de administrador ou você deve ter recebido permissão para chamar a operação da `DeleteOpsItem` API.

> **Important**
>
> Observe as seguintes informações importantes sobre esta operação:
>
> - A exclusão de um OpsItem é irreversível. Não é possível recuperar OpsItem excluídas.
> - Essa operação usa um modelo de consistência eventual, o que significa que o sistema pode levar alguns minutos para concluir essa operação. Se você excluir uma chamada OpsItem e imediatamente, por exemplo, Get (Obter)OpsItem, a chamada excluída ainda OpsItem poderá aparecer na resposta.
> - Essa operação é idempotente. O sistema não lançará uma exceção se você chamar essa operação repetidamente para a mesma OpsItem. Se a primeira chamada for bem-sucedida, todas as chamadas adicionais retornarão a mesma resposta de sucesso da primeira chamada.
> - Essa operação não é compatível com chamadas entre contas. Um administrador delegado ou uma conta de gerenciamento não pode excluir OpsItems em outras contas, mesmo que OpsCenter tenha sido configurada para administração entre contas. Para obter mais informações sobre a administração entre contas, consulte (Opcional) Configuração do OpsCenter para gerenciar OpsItems de forma centralizada entre contas.
> - Se você receber o `OpsItemLimitExceededException`, poderá excluir um ou mais OpsItems para reduzir seu número total OpsItems abaixo dos limites da cota. Para obter mais informações sobre a exceção, consulte Solução de problemas com o OpsCenter.

Excluir um OpsItem

Use o procedimento a seguir para excluir um OpsItem.

Como excluir uma OpsItem

1. Instale e configure o AWS CLI, caso ainda não o tenha feito. Para obter mais informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.
2. Execute o seguinte comando: Substitua *ID* pelo ID do OpsItem que você deseja excluir.

```
aws ssm delete-ops-item --ops-item-id ID
```

Se for bem-sucedido, o comando não retornará dados.

Correção de problemas do OpsItem

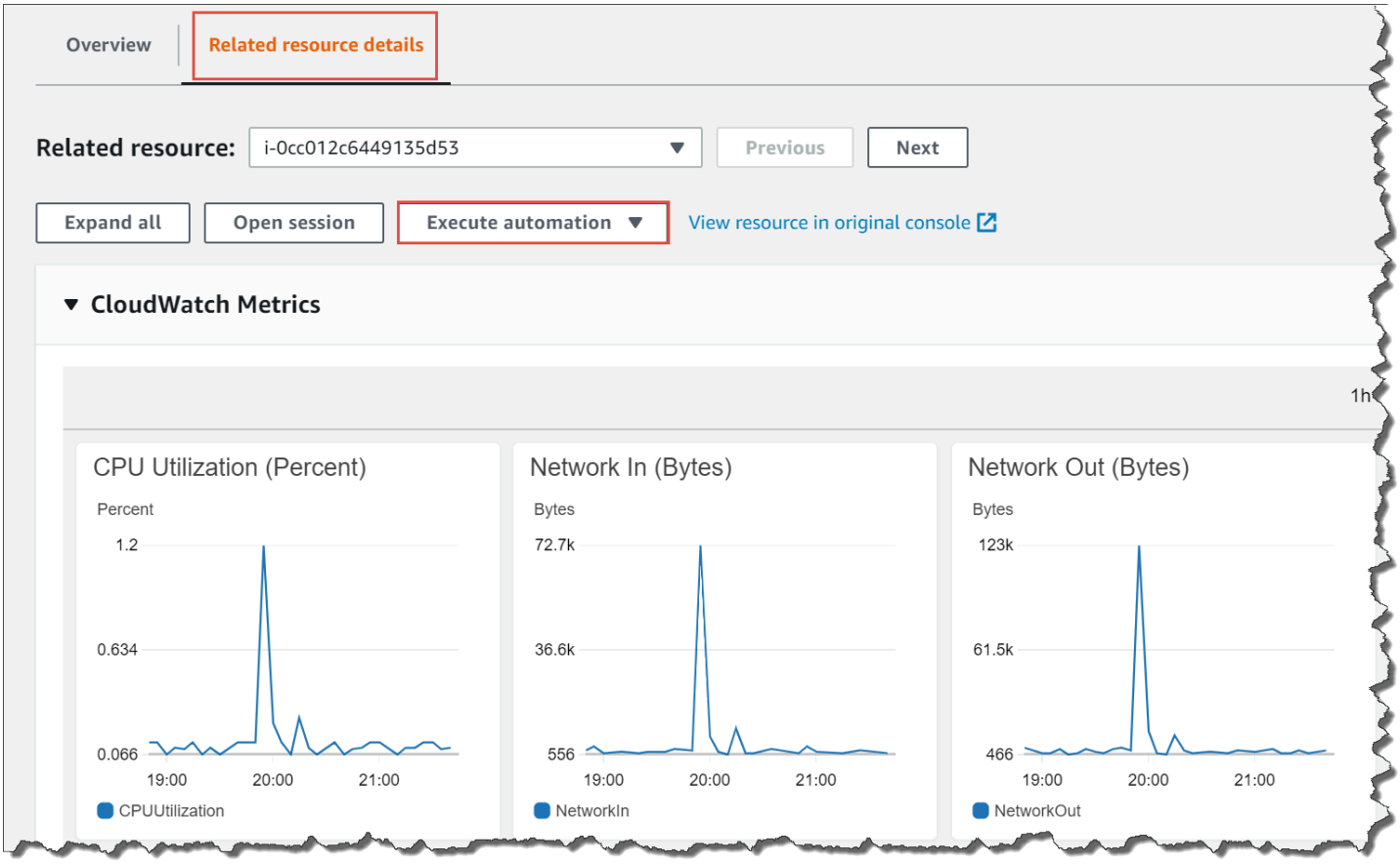
Ao usar runbooks do AWS Systems Manager Automation, é possível corrigir problemas com recursos da AWS identificados em um OpsItem. O Automation usa runbooks predefinidos para corrigir problemas comuns com recursos da AWS.

Cada OpsItem inclui a seção Runbooks que fornece uma lista de runbooks que podem ser usados para correção. Quando você escolhe um runbook do Automation na lista, o OpsCenter exibe automaticamente alguns dos campos requeridos para executar o documento. Ao executar um runbook do Automation, o sistema associa o runbook ao recurso relacionado do OpsItem. Se o Amazon EventBridge criar um OpsItem, ele associará um runbook ao OpsItem. O OpsCenter manterá um registro de 30 dias dos runbooks do Automation para um OpsItem.

É possível escolher um status para visualizar detalhes importantes sobre o runbook, como o motivo da falha de uma automação e qual etapa do runbook do Automation estava em execução quando ocorreu a falha, conforme mostrado no exemplo a seguir.



A página Detalhes do recurso relacionado para um OpsItem selecionado inclui a lista Executar automação. É possível escolher runbooks do Automation recentes ou específicos para o recurso e executá-los para corrigir problemas. Esta página também inclui provedores de dados, incluindo as métricas e os alarmes do Amazon CloudWatch, os logs do AWS CloudTrail e os detalhes do AWS Config.



Você pode visualizar informações sobre um runbook do Automation, escolhendo o nome dele no console ou usando o Referência de runbooks do Systems Manager Automation.

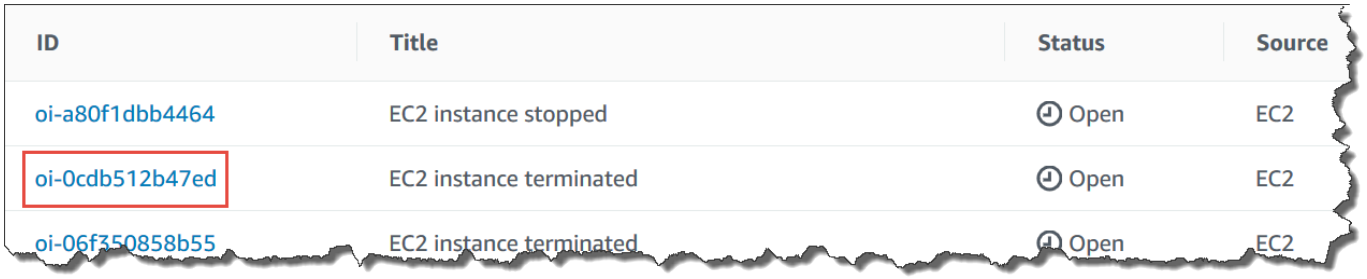
Como corrigir um OpsItem usando um runbook

Antes de usar um runbook do Automation para corrigir um problema do OpsItem, faça o seguinte:

- Verifique se você tem permissão para executar runbooks do Systems Manager Automation. Para ter mais informações, consulte Configurar a automação.
- Colete informações de ID específicas do recurso para a automação que você deseja executar. Por exemplo, caso deseje executar uma automação que reinicia uma instância do EC2, você deverá especificar o ID da instância do EC2 a ser reiniciada.

Para executar um runbook do Automation para corrigir um problema de OpsItem

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha OpsCenter.
3. Escolha o ID da OpsItem para abrir a página de detalhes.



4. Role até a seção Runbooks (Registros).
5. Use a barra de pesquisa ou os números no canto superior direito para localizar o runbook do Automation que você deseja executar.
6. Escolha um runbook e Executar.
7. Insira as informações necessárias do runbook e escolha Enviar.

   Depois que você iniciar o runbook, o sistema retornará à tela anterior e exibirá o status.
8. Na seção Execuções de automação nos últimos 30 dias, escolha o link ID da execução para visualizar as etapas e o status da execução.

Como corrigir um OpsItem usando um runbook associado

Após executar um runbook do Automation para um OpsItem, o OpsCenter associa o runbook ao OpsItem. Um runbook associado é classificado acima dos outros runbooks na lista Runbooks.

Use o procedimento a seguir para executar um runbook do Automation que foi associado a um recurso relacionado em um OpsItem. Para obter informações sobre como adicionar recursos relacionados, consulte Gerenciamento de OpsItems.

Para executar um runbook associado ao recurso para corrigir um problema da OpsItem

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha OpsCenter.
3. Abra o OpsItem.
4. Na seção Related resources (Recursos relacionados), escolha o recurso no qual você deseja executar o runbook do Automation.
5. Selecione Run automation (Executar automação) e escolha o runbook associado que você deseja executar.
6. Insira as informações necessárias do runbook e escolha Executar.

Depois que você iniciar o runbook, o sistema retornará à tela anterior e exibirá o status.

7. Na seção Execuções de automação nos últimos 30 dias, escolha o link ID da execução para visualizar as etapas e o status da execução.

Visualizar relatórios de resumo do OpsCenter

O AWS Systems Manager OpsCenter inclui uma página de resumo que automaticamente exibe as seguintes informações:

- Resumo do status do OpsItem: um resumo de OpsItems por status, como `Open` e `In progress`.
- Origens com a maioria de OpsItems abertos: um detalhamento dos principais Serviços da AWS com OpsItems abertos.
- OpsItems por origem e por idade: uma contagem de OpsItems, agrupados com base na origem e na quantidade de dias desde a criação.

Para visualizar relatórios do OpsCenter resumidos

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha OpsCenter e escolha a guia Resumo.
3. Na seção OpsItems por origem e idade, faça o seguinte:
    1. (Opcional) No campo de filtro, escolha Origem, selecione `Equal`, `Begin With` ou `Not Equal` e insira um parâmetro de pesquisa.
    2. Na lista adjacente, selecione um destes valores de status:
        - `Open`
        - `In progress`
        - `Resolved`
        - `Open and in progress`
        - `All`

Solução de problemas com o OpsCenter

Este tópico inclui informações para ajudar você a solucionar erros e problemas comuns com OpsCenter.

Você recebe o OpsItemLimitExceedException

Se você Conta da AWS atingiu o número máximo OpsItems permitido ao chamar a operação da API CreateOpsItem, você recebe um `OpsItemLimitExceededException`. OpsCenter retornará a exceção se sua chamada exceder o número máximo de OpsItems para qualquer uma das seguintes cotas:

- Número total de OpsItems por Conta da AWS por região (incluindo `Open` e `Resolved` OpSitems): 500.000
- Número máximo de OpsItems por Conta da AWS por mês: 10.000

Essas cotas se aplicam aos OpsItems criados de qualquer fonte, exceto o seguinte:

- OpsItems criado por AWS Security Hub descobertas
- OpsItems que são gerados automaticamente quando um incidente do gerente de incidentes é aberto

OpsItems criados a partir dessas fontes não contam para suas OpsItem cotas, mas você é cobrado por cada uma OpsItem.

Se você receber um `OpsItemLimitExceededException`, poderá excluí-lo manualmente OpsItems até ficar abaixo da cota, impedindo a criação de um novo OpsItem. Novamente, a exclusão OpsItems criada para descobertas do Security Hub ou incidentes do Incident Manager não reduzirá o número total de OpsItems impostas pelas cotas. Você deve excluir OpsItems de outras fontes. Para obter informações sobre como excluir um OpsItem, consulte Exclua OpsItems.

Você recebe uma fatura grande de AWS um grande número de contas geradas automaticamente OpsItems

Se você configurou a integração com AWS Security Hub, OpsCenter cria OpsItems para as descobertas do Security Hub. Dependendo do número de descobertas que o Security Hub gera e da conta em que você estava logado ao configurar a integração, OpsCenter pode gerar um grande número de OpsItems, com custos associados. Aqui estão detalhes mais específicos relacionados às descobertas OpsItems geradas pelo Security Hub:

- Se você estiver conectado à conta de administrador do Security Hub ao configurar uma integração do OpsCenter com o Security Hub, o sistema cria OpsItems para descobertas no administrador

e em todas as contas de membro. Todos os OpsItems são criados na conta do administrador. Dependendo de vários fatores, poderá resultar em uma fatura inesperadamente grande da AWS.

Se você estiver conectado a uma conta de membro ao configurar a integração, o sistema só criará OpsItems para descobertas nessa conta individual. Para obter mais informações sobre a conta do administrador do Security Hub, as contas de membro e sua relação com o feed de eventos do EventBridge para descobertas, consulte Types of Security Hub integration with EventBridge no Guia do usuário do AWS Security Hub.

- Para cada descoberta que cria um OpsItem, cobra-se o preço normal pela criação do OpsItem. Você também será cobrado se editar OpsItem ou se a descoberta correspondente for atualizada no Security Hub (o que aciona uma OpsItem atualização).
- Os OpsItems que são criados por uma integração com o AWS Security Hub não estão atualmente limitados pela cota máxima de 500.000 OpsItems por conta em uma Região. Portanto, é possível que os alertas do Security Hub criem mais de 500.000 OpsItems cobráveis em cada Região em uma conta.

  Portanto, para ambientes de alta produção, recomendamos limitar o escopo das descobertas do Security Hub somente a problemas de alta severidade.

> **Important**
>
> Se você acredita que um grande número deles OpsItems foi criado por engano e que sua AWS fatura é injustificada, entre em contato. Suporte

Use o procedimento a seguir, caso não queira mais que o sistema crie OpsItems para descobertas do Security Hub.

Para interromper o recebimento de OpsItems para descobertas do Security Hub

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha OpsCenter.
3. Escolha Settings.
4. Na seção Descobertas do Security Hub, escolha Editar.
5. Escolha o controle deslizante para alterar Habilitado para Desabilitado. Se não conseguir ativar o controle deslizante, o Security Hub não foi ativado para o seu Conta da AWS.

6. Escolha Salvar para salvar a configuração. O OpsCenter não cria mais OpsItems com base nas descobertas do Security Hub.

> **Important**
>
> Se OpsCenter a configuração voltar para Ativada e continuar a criar OpsItems as descobertas, faça login na conta de administrador delegado do Systems Manager ou na conta de AWS Organizations gerenciamento e repita esse procedimento. Se você não tiver permissão para entrar em nenhuma dessas contas, entre em contato com o seu administrador e peça a eles que repitam este procedimento para desativar a integração para a sua conta.

Usar painéis do Amazon CloudWatch hospedados pelo Systems Manager

Os painéis do Amazon CloudWatch são páginas iniciais personalizáveis no console do CloudWatch que você pode usar para monitorar seus recursos em uma única visualização, mesmo os recursos distribuídos em diferentes Regiões da AWS. Você pode usar os painéis do CloudWatch para criar visualizações personalizadas das métricas e dos alarmes para os recursos da AWS. Com os painéis, você pode criar o seguinte:

- Uma única visualização para determinadas métricas e alarmes para ajudá-lo a avaliar a saúde dos seus recursos e aplicações em uma ou mais Regiões da AWS. Você pode selecionar a cor usada para cada métrica em cada gráfico, para que possa monitorar a mesma métrica em vários gráficos.
- Um guia estratégico que fornece orientação para os membros da equipe durante eventos operacionais sobre como responder a incidentes específicos.
- Uma visualização comum de medições de recursos e aplicativos críticos que pode ser compartilhada pelos membros da equipe para obter um fluxo de comunicação mais rápido durante eventos operacionais.

Você pode criar painéis usando o console, a AWS Command Line Interface (AWS CLI) ou a API `PutDashboard` do CloudWatch. Para obter mais informações, consulte Utilizar painéis do Amazon CloudWatch no Manual do usuário do Amazon CloudWatch.

Trabalhar com o SSM Agent

O AWS Systems Manager Agent (SSM Agent) é um software da Amazon executado em instâncias, dispositivos de borda, servidores on-premises e máquinas virtuais (VMs) do Amazon Elastic Compute Cloud (Amazon EC2). O SSM Agent possibilita que o Systems Manager atualize, gerencie e configure esses recursos. O agente processa solicitações do serviço Systems Manager na Nuvem AWS e as executa conforme especificado. Em seguida, o SSM Agent retorna informações de status e execução ao serviço Systems Manager usando o Amazon Message Gateway Service (`ssmmessages`). (Nas Regiões da AWS lançadas antes de 2024, as informações de status e execução também podem ser enviadas de volta pelo Amazon Message Delivery Service (prefixo do serviço: `ec2messages`).)

Se você monitorar o tráfego, verá que seus nós gerenciados se comunicam com endpoints `ssmmessages.*` e, possivelmente, endpoints `ec2messages.*`. Para ter mais informações, consulte Referência: ec2messages, ssmmessages e outras operações da API. Para obter mais informações sobre portabilidade de logs do SSM Agent no Amazon CloudWatch Logs, consulte Registrar em log e monitorar no AWS Systems Manager.

Conteúdo

- Conheça os detalhes técnicos sobre o SSM Agent
- Encontrar AMIs com o SSM Agent pré-instalado
- Trabalhar com o SSM Agent em instâncias do EC2 para Linux
- Trabalhar com o SSM Agent em instâncias do EC2 para macOS
- Trabalhar com o SSM Agent em instâncias do EC2 para Windows Server
- Verificar o status do SSM Agent e iniciar o agente
- Verificar o número de versão do SSM Agent
- Visualizar logs do SSM Agent
- Restringir o acesso aos comandos em nível raiz por meio do SSM Agent
- Automatizar atualizações do SSM Agent
- Assinar as notificações do SSM Agent
- Solução de problemas do SSM Agent

Conheça os detalhes técnicos sobre o SSM Agent

Use as informações contidas neste tópico para implementar o AWS Systems Manager Agent (SSM Agent) e entender como ele funciona.

Tópicos

- Comportamento da credencial do SSM Agent versão 3.2.x.x
- Precedência de credenciais do SSM Agent
- Configurar o SSM Agent para usar com o Federal Information Processing Standard (FIPS, Padrão federal de processamento de informações)
- Sobre a conta local ssm-user
- SSM Agent e Instance Metadata Service (IMDS)
- Manter o SSM Agent atualizado
- Garantir que o diretório de instalação do SSM Agent não seja modificado, movido ou excluído
- Atualizações contínuas do SSM Agent nas Regiões da AWS
- Comunicações do SSM Agent com os buckets do S3 gerenciados pela AWS

Comportamento da credencial do SSM Agent versão 3.2.x.x

O SSM Agent armazena um conjunto de credenciais temporárias em `/var/lib/amazon/ssm/credentials` (para Linux e macOS) ou em `%PROGRAMFILES%\Amazon\SSM\credentials` (para Windows Server) quando uma instância é integrada usando a configuração de gerenciamento do host padrão na Quick Setup. As credenciais temporárias têm as permissões que você especifica para o perfil do IAM escolhido para a configuração de gerenciamento do host padrão. No Linux, só a conta raiz pode acessar essas credenciais. No Windows Server, somente a conta SYSTEM e os administradores locais podem acessar essas credenciais.

Precedência de credenciais do SSM Agent

Este tópico descreve informações importantes sobre como o SSM Agent recebe permissão para executar ações em seus recursos.

> **Note**
>
> A compatibilidade com dispositivos de borda é um pouco diferente. Você deve configurar seus dispositivos de borda para usar o software principal do AWS IoT Greengrass, configurar

um perfil de serviço do AWS Identity and Access Management (IAM) e implantar o SSM Agent para seus dispositivos usando o AWS IoT Greengrass. Para obter mais informações, consulte Gerenciar dispositivos de borda com o Systems Manager.

Quando o SSM Agent está instalado em uma máquina, ele requer permissões para se comunicar com o serviço do Systems Manager. Nas instâncias do Amazon Elastic Compute Cloud (Amazon EC2), essas permissões são fornecidas em um perfil de instância que está anexado à instância. Em uma máquina que não é do EC2, o SSM Agent normalmente obtém as permissões necessárias do arquivo de credenciais compartilhadas, localizado em `/root/.aws/credentials` (Linux e macOS) ou em `%USERPROFILE%\.aws\credentials` (Windows Server). As permissões necessárias são adicionadas a este arquivo durante o processo de ativação híbrida.

Porém, em casos raros, uma máquina pode acabar com permissões adicionadas a mais de um dos locais em que o SSM Agent verifica se há permissões para executar suas tarefas.

Por exemplo, digamos que você tenha configurado uma instância do EC2 para ser gerenciada pelo Systems Manager. Essa configuração inclui anexar um perfil de instância. Mas então você também decide usar essa instância para tarefas de desenvolvedor ou usuário final e instalar o AWS Command Line Interface (AWS CLI) nele. Esta instalação resulta em permissões adicionais sendo adicionadas a um arquivo de credenciais na instância.

Quando você executa um comando do Systems Manager na instância, o SSM Agent pode tentar usar credenciais diferentes daquelas que você espera que ele use, como de um arquivo de credenciais em vez de um perfil de instância. Isto é porque o SSM Agent procura credenciais na ordem prescrita para a Cadeia de fornecedores de credenciais padrão.

> **Note**
>
> No Linux e no macOS, o SSM Agent é executado como usuário raiz. Portanto, as variáveis de ambiente e o arquivo de credenciais que o SSM Agent procura neste processo são somente do usuário raiz (`/root/.aws/credentials`). O SSM Agent não verifica as variáveis de ambiente ou o arquivo de credenciais para quaisquer outros usuários na instância durante a pesquisa de credenciais.

A cadeia de fornecedores procura e usa credenciais nesta ordem:

1. Variáveis de ambiente, se configuradas (`AWS_ACCESS_KEY_ID` e `AWS_SECRET_ACCESS_KEY`).

2. Arquivo de credenciais compartilhado (`$HOME/.aws/credentials` para Linux e macOS ou `%USERPROFILE%\.aws\credentials` para Windows Server) com permissões fornecidas por, por exemplo, uma ativação híbrida ou uma instalação da AWS CLI.
3. Uma função do AWS Identity and Access Management (IAM) para tarefas se houver uma aplicação que usa uma definição de tarefa do Amazon Elastic Container Service (Amazon ECS) ou operação da API RunTask.
4. Um perfil de instância anexado a uma instância do Amazon EC2.
5. O perfil do IAM escolhido para a configuração de gerenciamento do host padrão.

Para obter informações mais detalhadas, consulte os seguintes tópicos relacionados:

- Perfis de instância para instâncias do EC2: Configurar permissões de instância obrigatórias para o Systems Manager
- Ativações híbridas: crie uma ativação híbrida para registrar nós no Systems Manager
- Credenciais da AWS CLI: Configuração e definições do arquivo de credenciais no Guia do usuário do AWS Command Line Interface
- Cadeia de provedores de credenciais padrão – Especificação de credenciais no Manual do desenvolvedor do AWS SDK para Go

> **Note**
>
> Este tópico no Guia do desenvolvedor do AWS SDK para Go descreve a cadeia de provedores padrão em termos do SDK para Go. No entanto, os mesmos princípios se aplicam à avaliação de credenciais para o SSM Agent.

Configurar o SSM Agent para usar com o Federal Information Processing Standard (FIPS, Padrão federal de processamento de informações)

Se precisar usar o Systems Manager com módulos criptográficos validados pelo Federal Information Processing Standard (FIPS) 140-3, você pode configurar o AWS Systems Manager Agent (SSM Agent) para usar os endpoints FIPS nas regiões compatíveis.

Para configurar o SSM Agent para se conectar aos endpoints FIPS 140-3

1. Conecte-se ao seu nó gerenciado.

2. Navegue até o diretório que contém o arquivo `amazon-ssm-agent.json`:

   - Linux: `/etc/amazon/ssm/`
   - macOS: `/opt/aws/ssm/`
   - Windows Server: `C:\Program Files\Amazon\SSM\`

3. Abra o arquivo chamado `amazon-ssm-agent.json` para edição.

> **Tip**
>
> Se ainda não existir nenhum arquivo `amazon-ssm-agent.json`, copie o conteúdo de `amazon-ssm-agent.json.template` para um novo arquivo chamado `amazon-ssm-agent.json`. Salve o `amazon-ssm-agent.json` no mesmo diretório em que `amazon-ssm-agent.json.template` está localizado.

4. Adicione o conteúdo a seguir ao arquivo. Substitua os valores do espaço reservado de *`region`* pelo código de região apropriado para sua partição:

```
{
    ---Existing file content, if any---
    "Mds": {
        "Endpoint": "ec2messages-fips.region.amazonaws.com",
    },
    "Ssm": {
        "Endpoint": "ssm-fips.region.amazonaws.com",
    },
    "Mgs": {
        "Endpoint": "ssmmessages-fips.region.amazonaws.com",
        "Region": "region"
    },
    "S3": {
        "Endpoint": "s3-fips.dualstack.region.amazonaws.com",
        "Region": region"
    },
    "Kms": {
        "Endpoint": "kms-fips.region.amazonaws.com"
    }
}
```

As regiões compatíveis incluem:

- `us-east-1` para a região Leste dos EUA (Norte da Virgínia)
- `us-east-2` para a região Leste dos EUA (Ohio)
- `us-west-1` para a região Oeste dos EUA (Norte da Califórnia)
- `us-west-2` para a região Oeste dos EUA (Oregon)
- `ca-west-1` para a região Oeste do Canadá (Calgary)

5. Salve o arquivo e reinicie o SSM Agent.

Toda vez que você alterar a configuração, reinicie o SSM Agent.

Você pode personalizar outros recursos do SSM Agent usando o mesmo procedimento. Para obter uma lista atualizada das propriedades de configuração disponíveis e seus valores padrão, consulte Config Property Definitions no repositório `amazon-ssm-agent` no GitHub.

Para obter mais informações sobre suporte da AWS para FIPS, consulte Federal Information Processing Standard (FIPS) 140-3.

Sobre a conta local ssm-user

Começando na versão 2.3.50.0 do SSM Agent, o agente cria uma conta de usuário local chamada `ssm-user` e a adiciona ao diretório `/etc/sudoers.d` (Linux e macOS) ou ao grupo de Administradores (Windows Server). Em versões do agente anteriores a 2.3.612.0, a conta é criada na primeira vez que o SSM Agent é iniciado ou reiniciado após a instalação. Na versão 2.3.612.0 e posteriores, a conta `ssm-user` é criada na primeira vez que uma sessão é iniciada em uma instância. Esse `ssm-user` é o usuário padrão do sistema operacional quando uma sessão do é iniciada no Session Manager, uma ferramenta do AWS Systems Manager. É possível alterar as permissões movendo `ssm-user` para um grupo com menos privilégios ou alterando o arquivo `sudoers`. A conta `ssm-user` não é removida do sistema quando o SSM Agent é desinstalado.

No Windows Server, o SSM Agent lida com a configuração de uma nova senha para a conta `ssm-user` quando cada sessão começa. Nenhuma senha é definida para `ssm-user` em instâncias gerenciadas do Linux.

Começando com o SSM Agent versão 2.3.612.0, a conta `ssm-user` não é criada automaticamente em máquinas Windows Server usadas como controladores de domínio. Para usar o Session Manager em um controlador de domínio do Windows Server, crie a conta `ssm-user` manualmente, caso ela ainda não esteja presente, e atribua permissões de Administrador do Domínio ao usuário.

> **Important**
>
> Para que a conta `ssm-user` seja criada, o perfil de instância anexado à instância deve fornecer as permissões necessárias. Para obter informações, consulte Etapa 2: verificar ou adicionar permissões de instância para o Session Manager.

SSM Agent e Instance Metadata Service (IMDS)

O agente do Systems Manager depende dos metadados da instância do EC2 para funcionar corretamente. O Systems Manager pode acessar metadados de instância usando a versão 1 ou a versão 2 do Instance Metadata Service (IMDSv1 e IMDSv2). Sua instância deve poder acessar o endereço IPv4 do serviço de metadados da instância: 169.254.169.254. Para obter mais informações, consulte Metadados da instância e dados do usuário no Manual do usuário do Amazon EC2.

Manter o SSM Agent atualizado

Uma versão atualizada do SSM Agent é lançada sempre que novas ferramentas são adicionadas ao Systems Manager ou sempre que atualizações são feitas nas ferramentas existentes. Deixar de usar a versão mais recente do agente pode impedir que seu nó gerenciado use várias ferramentas do Systems Manager. Por isso, recomendamos automatizar o processo de manter o SSM Agent atualizado em suas máquinas. Para mais informações, consulte Automatizar atualizações do SSM Agent. Inscreva-se na página Notas de versão do SSM Agent no GitHub para receber notificações sobre atualizações do SSM Agent.

> **Note**
>
> Uma versão atualizada do SSM Agent é lançada sempre que novas ferramentas são adicionadas ao Systems Manager ou sempre que atualizações são feitas nas ferramentas existentes. Deixar de usar a versão mais recente do agente pode impedir que seu nó gerenciado use várias ferramentas do Systems Manager. Por isso, recomendamos automatizar o processo de manter o SSM Agent atualizado em suas máquinas. Para mais informações, consulte Automatizar atualizações do SSM Agent. Inscreva-se na página Notas de versão do SSM Agent no GitHub para receber notificações sobre atualizações do SSM Agent.
>
> As Amazon Machine Images (AMIs), que incluem o SSM Agent por padrão, podem demorar até duas semanas para serem atualizadas com a versão mais recente do SSM Agent.

Recomendamos que você configure atualizações automatizadas ainda mais frequentes para o SSM Agent.

Garantir que o diretório de instalação do SSM Agent não seja modificado, movido ou excluído

O SSM Agent está instalado em `/var/lib/amazon/ssm/` (Linux e macOS) e em `%PROGRAMFILES%\Amazon\SSM\` (Windows Server). Esses diretórios de instalação contêm arquivos e pastas essenciais usados pelo SSM Agent, como um arquivo de credenciais, recursos para comunicação entre processos (IPC) e pastas de orquestração. Nada no diretório de instalação deve ser modificado, movido ou excluído. Caso contrário, o SSM Agent poderá parar de funcionar corretamente.

Atualizações contínuas do SSM Agent nas Regiões da AWS

Quando uma atualização do SSM Agent estiver disponível em seu repositório do GitHub, até duas semanas poderão ser necessárias para que a versão atualizada seja lançada para todos as Regiões da AWS em momentos diferentes. Por esse motivo, você pode receber o erro “Não compatível com a plataforma atual” ou “atualizando o amazon-ssm-agent para uma versão mais antiga, ative a permissão de downgrade para continuar” ao tentar implantar uma nova versão do SSM Agent em uma região.

Para determinar a versão do SSM Agent disponível para você, você pode executar um comando `curl`.

Para visualizar a versão do agente disponível no bucket de download global, execute o comando a seguir.

```
curl https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/VERSION
```

Para visualizar a versão do agente disponível em uma região específica, execute o comando a seguir, substituindo *`region`* pela região em que você está trabalhando, como `us-east-2` para a região Leste dos EUA (Ohio).

```
curl https://s3.region.amazonaws.com/amazon-ssm-region/latest/VERSION
```

Também é possível abrir o arquivo `VERSION` diretamente no seu navegador sem um comando `curl`.

Comunicações do SSM Agent com os buckets do S3 gerenciados pela AWS

Durante a execução de várias operações do Systems Manager, o AWS Systems Manager Agent (SSM Agent) acessa uma série de buckets do Amazon Simple Storage Service (Amazon S3). Esses buckets do S3 são acessíveis publicamente e, por padrão, SSM Agent se conecta a eles usando chamadas HTTP.

No entanto, se você estiver usando um endpoint da nuvem privada virtual (VPC) nas operações do Systems Manager, deverá fornecer permissão explícita em um perfil de instância do Amazon Elastic Compute Cloud (Amazon EC2) para o Systems Manager ou em um perfil de serviço para máquinas que não são do EC2 em um ambiente híbrido e multinuvem. Caso contrário, seus recursos não poderão acessar esses buckets públicos.

Para conceder acesso dos nós gerenciados a esses buckets quando você estiver usando um endpoint da VPC, crie uma política de permissões personalizada do Amazon S3 e anexe-a ao perfil de instância (para instâncias do EC2) ou aos perfil de serviço (para servidores nós gerenciados que não são do EC2).

Para obter informações sobre como usar um endpoint da nuvem privada virtual (VPC) em suas operações do Systems Manager, consulte Melhorar a segurança das instâncias do EC2 usando endpoints da VPC para o Systems Manager.

> **Note**
>
> Essas permissões fornecem acesso somente aos buckets gerenciados da AWS exigidos pelo SSM Agent. Elas não fornecem as permissões que são necessárias para outras operações do Amazon S3. Elas também não fornecem permissão para seus próprios buckets do S3.

Para obter mais informações, consulte os tópicos a seguir.

- Configurar permissões de instância obrigatórias para o Systems Manager
- Criar o perfil de serviço do IAM obrigatório para o Systems Manager em ambientes híbridos e multinuvem
- Referência: buckets do Amazon S3 para operações de aplicação de patches

Conteúdo

- Permissões obrigatórias do bucket
- Exemplo
- Validar máquinas ativadas para ambiente híbrido usando uma impressão digital do hardware
- SSM Agent na GitHub

Permissões obrigatórias do bucket

A tabela a seguir descreve cada um dos S3 que o SSM Agent pode precisar acessar as operações do Systems Manager.

> **Note**
>
> A *região* representa o identificador da região para uma região da Região da AWS compatível com o AWS Systems Manager, como `us-east-2` para a região Leste dos EUA (Ohio). Para ver uma lista dos valores de *região* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

Permissões do Amazon S3 exigidas pelo SSM Agent

ARN do bucket do S3	Descrição
`arn:aws:s3:::aws-windows-downloads-`*region*`/*`	Obrigatório para alguns documentos SSM que oferecem suporte somente a sistemas operacionais Windows Server, além de alguns para suporte multiplataforma, como `AWSEC2-ConfigureSTIG` .
`arn:aws:s3:::amazon-ssm-`*region*`/*`	Obrigatório para atualizar instalações do SSM Agent. Esses buckets contêm os pacotes de instalação do SSM Agent e os manifestos de instalação referenciados pelo documento e plugin do `AWS-UpdateSSMAgent` . Se essas permissões não forem fornecidas, o SSM Agent fará uma chamada HTTP para fazer download da atualização.

ARN do bucket do S3	Descrição
`arn:aws:s3:::aws-ssm-` *region*`/*`	Fornece acesso ao bucket do S3 que contém os módulos necessários para usar com documentos sem patches do Systems Manager (documentos do Command SSM). Por exemplo: `arn:aws:s3:::aws-ssm-us-east-2/*` . Veja a seguir alguns documentos do SSM comumente usados armazenados nesses buckets. • `AWS-ConfigureWindowsUpdate` • `AWS-FindWindowsUpdates` • `AWS-UpdateSSMAgent` • `AWS-UpdateEC2Config`

ARN do bucket do S3	Descrição

`arn:aws:s3:::patch-baseline-snapshot-` *region*`/*`
 	
- ou -
 	
`arn:aws:s3:::patch-baseline-snapshot-` *region*`-`*unique-suffix*` /*`
	Fornece acesso ao bucket do S3 que contém snapshots de linha de base de patches. Isso será necessário se você usar qualquer um dos seguintes documentos do Command SSM:
	
	• `AWS-RunPatchBaseline`
	• `AWS-RunPatchBaselineAssociation`
	• `AWS-RunPatchBaselineWithHooks`
	• `AWS-ApplyPatchBaseline` um documentos do SSM legado)
	
	Os buckets para a maioria das Regiões da AWS compatíveis usam o seguinte formato:
	
	`arn:aws:s3:::patch-baseline-snapshot-` *region*
	
	Para algumas regiões, um sufixo exclusivo adicional é incluído no nome do bucket. Por exemplo, o nome do bucket para a região Oriente Médio (Bahrein) (me-south-1) é o seguinte:
	
	• `patch-baseline-snapshot-me-south-1-uduvl7q8`
	
	Para obter uma lista completa dos nomes de buckets de snapshot da lista de referência de patches, consulte Buckets contendo snapshots da lista de referência de patches gerenciadas pela AWS.

ARN do bucket do S3	Descrição
	Note Se você usar um firewall on-premises e planeja usar o Patch Manager, esse firewall também deverá permitir o acesso ao endpoint da lista de referência de patches apropriado.

ARN do bucket do S3	Descrição
Para nós gerenciados do Linux e do Windows Server: `arn:aws:s3:::aws-patch-manager- region-unique-suffix /*` Em instâncias do Amazon EC2 para macOS: `arn:aws:s3:::aws-patchmanager-macos- region-unique-suffix /*`	Fornece acesso ao bucket do S3 que contém documentos do SSM Command para operações de aplicação de patches no Patch Manager. Cada nome de bucket inclui um sufixo exclusivo, como `552881074` para buckets na região Leste dos EUA (Ohio) (us-east-2): • `arn:aws:s3:::aws-patch-managerer-us-east-2-552881074/*` • `arn:aws:s3::: aws-patchmanager-macos-us-east-2-552881074/*` Documentos do SSM Veja a seguir alguns documentos SSM comumente usados armazenados nesses buckets. • `AWS-RunPatchBaseline` • `AWS-RunPatchBaselineAssociation` • `AWS-RunPatchBaselineWithHooks` • `AWS-InstanceRebootWithHooks` • `AWS-PatchAsgInstance` • `AWS-PatchInstanceWithRollback` Para obter listas completas de buckets do S3 gerenciados pela AWS para operações de aplicação de patches, consulte os seguintes tópicos:

ARN do bucket do S3	Descrição
	• Buckets contendo documentos do SSM Command para operações de aplicação de patches (Linux e Windows Server) • Buckets contendo documentos do SSM Command para operações de aplicação de patches (macOS)

Exemplo

O exemplo a seguir ilustra como fornecer acesso aos buckets do S3 necessários para as operações do Systems Manager na região Leste dos EUA (Ohio) (us-east-2). Na maioria dos casos, você precisa fornecer essas permissões explicitamente em um perfil de instância ou função de serviço somente ao usar um endpoint da VPC.

> **Important**
>
> Recomendamos que você evite usar caracteres curinga (*) no lugar das regiões específicas nessa política. Por exemplo, use `arn:aws:s3:::aws-ssm-us-east-2/*` e não use `arn:aws:s3:::aws-ssm-*/*`. O uso de curingas pode fornecer acesso a buckets do S3 aos quais você não pretende conceder acesso. Se você quiser usar o perfil de instância para mais de uma região, recomendamos repetir o primeiro bloco `Statement` para cada região.

JSON

```
{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": "s3:GetObject",
                "Resource": [
                    "arn:aws:s3:::aws-windows-downloads-us-east-2/*",
                    "arn:aws:s3:::amazon-ssm-us-east-2/*",
                    "arn:aws:s3:::aws-ssm-us-east-2/*",
                    "arn:aws:s3:::patch-baseline-snapshot-us-east-2/*",
```

```
                "arn:aws:s3:::aws-patch-manager-us-east-2-552881074/*",
                "arn:aws:s3:::aws-patchmanager-macos-us-east-2-552881074/*"
            ]
        }
    ]
}
```

Validar máquinas ativadas para ambiente híbrido usando uma impressão digital do hardware

Quando há máquinas que não são EC2 em um ambiente híbrido e multinuvem, o SSM Agent reúne uma série de atributos do sistema (referidos como hash de hardware) e usa esses atributos para computar uma impressão digital. A impressão digital é uma string opaca que o agente passa para determinadas APIs do Systems Manager. Essa impressão digital exclusiva associa o chamador a um nó gerenciado ativado para ambientes híbridos específico. O agente armazena a impressão digital e o hash de hardware no disco local em um local chamado Cofre.

O agente calcula o hash de hardware e a impressão digital quando a máquina é registrada para uso com o Systems Manager. Em seguida, a impressão digital é passada de volta para o serviço Systems Manager quando o agente envia um comando `RegisterManagedInstance`.

Posteriormente, ao enviar um `RequestManagedInstanceRoleToken`, o agente verifica a impressão digital e o hash de hardware no Cofre para se certificar de que os atributos de máquina atuais correspondam ao hash de hardware armazenado. Se os atributos atuais da máquina corresponderem ao hash de hardware armazenado no Vault, o agente passa a impressão digital do Vault para `RegisterManagedInstance`, resultando em uma chamada bem-sucedida.

Se os atributos de máquina atuais não corresponderem ao hash de hardware armazenado, o SSM Agent calcula uma nova impressão digital, armazena o novo hash de hardware e a impressão digital no Vault e passa a nova impressão digital para `RequestManagedInstanceRoleToken`. Isso faz *`RequestManagedInstanceRoleToken`* falhar e o agente não poderá obter um token de função para se conectar ao serviço do Systems Manager.

Esta falha ocorre intencionalmente e é usada como uma etapa de verificação para impedir que vários nós gerenciados se comuniquem com o serviço do Systems Manager como o mesmo nó gerenciado.

Ao comparar os atributos da máquina atual com o hash de hardware armazenado no Cofre, o agente usa a seguinte lógica para determinar se os hashes antigos e novos correspondem:

- Se o SID (ID do sistema/máquina) for diferente, não haverá nenhuma correspondência.
- Caso contrário, se o endereço IP for o mesmo, então correspondem.
- Caso contrário, a porcentagem de atributos de máquina correspondentes é calculada e comparada com o limite de similaridade configurado pelo usuário para determinar se há uma correspondência.

O limite de similaridade é armazenado no Cofre, como parte do hash de hardware.

O limite de similaridade pode ser definido depois que uma instância é registrada usando um comando como o seguinte.

Em máquinas Linux:

```
sudo amazon-ssm-agent -fingerprint -similarityThreshold 1
```

Em máquinas Windows Server que usam o PowerShell:

```
cd "C:\Program Files\Amazon\SSM\" `
    .\amazon-ssm-agent.exe -fingerprint -similarityThreshold 1
```

> **Important**
>
> Se um dos componentes usados para calcular a impressão digital mudar, isso pode fazer com que o agente hiberne. Para ajudar a evitar essa hibernação, defina o limite de similaridade para um valor baixo, como **1**.

SSM Agent na GitHub

O código-fonte para o SSM Agent está disponível no GitHub para que você possa adaptar o agente de acordo com suas necessidades. Incentivamos você a enviar solicitações pull sobre alterações que gostaria que fosse incluídas. Porém, a Amazon Web Services não oferece suporte à execução de cópias modificadas desse software.

Encontrar AMIs com o SSM Agent pré-instalado

O AWS Systems Manager Agent (SSM Agent) está pré-instalado em alguns Amazon Machine Images (AMIs) fornecidos pela AWS e por terceiros confiáveis.

Por exemplo, ao iniciar uma instância do Amazon Elastic Compute Cloud (Amazon EC2) criada de uma AMI com um dos seguintes sistemas operacionais, você provavelmente descobrirá que o SSM Agent já está instalado:

- AlmaLinux
- AMIs base do Amazon Linux 1 datadas de 9/2017 e posteriores
- Amazon Linux 2
- AMIs da Base otimizada para ECS do Amazon Linux 2
- Amazon Linux 2023 (AL2023)
- AMIs do Amazon Linux otimizadas para Amazon EKS
- macOS 10.14.x (Mojave), 10.15.x (Catalina), 11.x (Big Sur), 12.x (Monterey), 13.x (Ventura), 14.x (Sonoma) e 15.x (Sequoia)
- SUSE Linux Enterprise Server (SLES) 12 e 15
- Ubuntu Server 16.04 LTS de 64 bits (Snap), 18.04, 20.04, 22.04 LTS, 23.10, 24.04 LTS, 24.0 e 25.04
- Windows Server 2008-2012 R2 AMIs publicadas em novembro de 2016 ou mais tarde
- Windows Server 2016, 2019, 2022 (exceto versões Nano) e 2025

> **Note**
>
> A versão do SSM Agent pré-instalada em um AMI pode não ser a versão mais recente disponível. Como prática recomendada, use sempre a versão mais recente do SSM Agent em seus nós gerenciados. Para obter mais informações sobre automatização das atualizações do SSM Agent, consulte Automatizar atualizações do SSM Agent.
>
> O SSM Agent pode estar pré-instalado em AMIs gerenciadas pela AWS que não estejam nesta lista. Isso normalmente indica que não haverá compatibilidade do sistema operacional (SO) com todas as ferramentas do Systems Manager.
>
> O SSM Agent também pode estar pré-instalado nas AMIs encontradas no AWS Marketplace ou no repositório de AMIs da comunidade, mas, a AWS não oferece suporte a essas AMIs.

Verifique o status do SSM Agent

Dependendo de quando foi inicializada, uma instância criada a partir de uma AMI na lista anterior pode não ter sido pré-instalada pelo SSM Agent. Também é possível que uma instância tenha o

agente pré-instalado, mas o agente não esteja em execução. Portanto, recomendamos que você verifique o status do SSM Agent antes de tentar usar o Systems Manager em uma instância pela primeira vez.

Use o procedimento a seguir para verificar se o SSM Agent está instalado e em execução em uma instância. Se achar que o agente não está instalado, é possível instalá-lo manualmente em instâncias do Linux, macOS, e Windows Server.

Para verificar a instalação do SSM Agent em uma instância

1. Depois de iniciar uma nova instância, aguarde alguns minutos para que ela seja inicializada.
2. Conecte-se à instância usando o método preferido. Por exemplo, você pode usar o SSH para se conectar às instâncias do Linux ou usar o Remote Desktop para se conectar às instâncias do Windows Server.
3. Verifique o status do SSM Agent executando o comando para o tipo de sistema operacional da instância.

Sistema operacional	Command
Amazon Linux 1	`sudo status amazon-ssm-agent`
Amazon Linux 2 e Amazon Linux 2023	`sudo systemctl status amazon-ssm-agent`
macOS	Não há nenhum comando para verificar o status do SSM Agent no macOS. Você pode verificar o status localizando e avaliando o arquivo de log `/var/log/amazon/ssm/amazon-ssm-agent.log` do agente.
SUSE Linux Enterprise Server	`sudo systemctl status amazon-ssm-agent`
Ubuntu Server (32 bits)	`sudo status amazon-ssm-agent`
Ubuntu Server (64 bits, Deb)	`sudo systemctl status amazon-ssm-agent`

Sistema operacional	Command
Ubuntu Server (64 bits, Snap)	`sudo systemctl status snap.amazon-ssm-agent.amazon-ssm-agent.service`
Windows Server	`Get-Service AmazonSSMAgent`

> **Tip**
>
> Para visualizar os comandos para verificar o status do SSM Agent em todos os tipos de sistema operacional compatíveis com o Systems Manager, consulte Verificar o status do SSM Agent e iniciar o agente.

4. Avalie a saída do comando para saber o status do SSM Agent.

   Status: Installed and running (Instalado e sendo executado)

   Na maioria dos casos, a saída do comando indica que o agente está instalado e sendo executado.

   O exemplo a seguir mostra que o SSM Agent está instalado e sendo executado em uma instância do Amazon Linux 2.

```
amazon-ssm-agent.service - amazon-ssm-agent
Loaded: loaded (/usr/lib/systemd/system/amazon-ssm-agent.service; enabled; vendor
 preset: enabled)
Active: active (running) since Wed 2021-10-20 19:09:29 UTC; 4min 6s ago
--truncated--
```

   O exemplo a seguir mostra que o SSM Agent está instalado e sendo executado em uma instância do Windows Server.

```
Status   Name               DisplayName
------   ----               -----------
Running  AmazonSSMAgent     Amazon SSM Agent
```

Status: Installed and running (Instalado e não sendo executado)

Na alguns casos, a saída do comando indica que o agente está instalado, mas não está sendo executado.

O exemplo a seguir mostra que o SSM Agent está instalado, mas não está sendo executado em uma instância do Amazon Linux 2.

```
amazon-ssm-agent.service - amazon-ssm-agent
Loaded: loaded (/usr/lib/systemd/system/amazon-ssm-agent.service; enabled; vendor
 preset: enabled)
Active: inactive (dead) since Wed 2021-10-20 22:16:41 UTC; 18s ago
--truncated--
```

O exemplo a seguir mostra que o SSM Agent está instalado, mas não está sendo executado em uma instância do Windows Server.

```
Status   Name               DisplayName
------   ----               -----------
Stopped  AmazonSSMAgent     Amazon SSM Agent
```

Se o agente estiver instalado, mas não estiver seno executado, ative-o manualmente usando o comando para o tipo de sistema operacional em questão.

Sistema operacional	Command
Amazon Linux 1	`sudo start amazon-ssm-agent`

Amazon Linux 2 e Amazon Linux 2023	`sudo systemctl enable amazon-ssm-agent`
	
	`sudo systemctl start amazon-ssm-agent`

Sistema operacional	Command

macOS	`sudo launchctl load -w /Library/LaunchDaemons/com.amazon.aws.ssm.plist`
	
	`sudo launchctl start com.amazon.aws.ssm`

SUSE Linux Enterprise Server	`sudo systemctl enable amazon-ssm-agent`
	
	`sudo systemctl start amazon-ssm-agent`

Ubuntu Server (32 bits)	`sudo start amazon-ssm-agent`

Ubuntu Server (64 bits, Deb)	`sudo systemctl enable amazon-ssm-agent`
	
	`sudo systemctl start amazon-ssm-agent`

Ubuntu Server (64 bits, Snap)	`sudo snap start amazon-ssm-agent`
Windows Server	Execute o seguinte comando no PowerShell. `Start-Service AmazonSSMAgent`

Status: Not installed (Não instalado)

Na alguns casos, a saída do comando indica que o agente não está instalado.

O exemplo a seguir mostra que o SSM Agent não está instalado em uma instância do Amazon Linux 2.

```
Unit amazon-ssm-agent.service could not be found.
```

O exemplo a seguir mostra que o SSM Agent não está instalado em uma instância do Windows Server.

```
Get-Service : Cannot find any service with service name 'AmazonSSMAgent'.
--truncated--
```

Se o agente não estiver instalado, você poderá instalá-lo manualmente usando o procedimento para o tipo de sistema operacional:

- Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux
- Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para macOS
- Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Windows Server

Trabalhar com o SSM Agent em instâncias do EC2 para Linux

O AWS Systems Manager Agent (SSM Agent) processa solicitações do Systems Manager e configura sua máquina conforme especificado na solicitação. Use os procedimentos descritos nos tópicos a seguir para instalar, configurar ou desinstalar SSM Agent em sistemas operacionais Linux.

Tópicos

- Verificar a assinatura do SSM Agent
- Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux
- Configurar o SSM Agent para usar um proxy em nós do Linux

Verificar a assinatura do SSM Agent

Os pacotes do instalador deb e rpm do AWS Systems Manager Agent (SSM Agent) para instâncias Linux são assinados criptograficamente. Use a chave pública para verificar se o arquivo de download do agente é original e não modificado. Se os arquivos sofrerem quaisquer danos ou alterações, a verificação falhará. Você pode verificar a assinatura do pacote instalador usando RPM ou GPG. As informações a seguir são para SSM Agent versões 3.1.1141.0 ou posteriores.

Para encontrar o arquivo de assinatura correto para a arquitetura e o sistema operacional de sua instância, consulte a tabela a seguir.

A *região* representa o identificador da região para uma região da Região da AWS compatível com o AWS Systems Manager, como `us-east-2` para a região Leste dos EUA (Ohio). Para ver uma lista dos valores de *região* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

Arquitetura	Sistema operacional	URL do arquivo de assinatura	Nome do arquivo de download do agente

x86_64	AlmaLinux, Amazon Linux 1, Amazon Linux 2, Amazon Linux 2023, CentOS, CentOS Stream, RHEL, Oracle Linux, Rocky Linux, SLES	`https://s3.region.amazonaws.com/amazon-ssm-region/latest/linux_amd64/amazon-ssm-agent.rpm.sig`
		
		`https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux_amd64/amazon-ssm-agent.rpm.sig`
			`amazon-ssm-agent.rpm`

x86_64	Debian Server, Ubuntu Server	`https://s3.region.amazonaws.com/amazon-ssm-region/latest/debian_amd64/amazon-ssm-agent.deb.sig`	`amazon-ssm-agent.deb`

Arquitetura	Sistema operacional	URL do arquivo de assinatura	Nome do arquivo de download do agente
		https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/debian_amd64/amazon-ssm-agent.deb.sig	
x86	Amazon Linux 1, Amazon Linux 2, Amazon Linux 2023, CentOS, RHEL	https://s3.*region*.amazonaws.com/amazon-ssm-*region*/latest/linux_386/amazon-ssm-agent.rpm.sig https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux_386/amazon-ssm-agent.rpm.sig	amazon-ssm-agent.rpm

Arquitetura	Sistema operacional	URL do arquivo de assinatura	Nome do arquivo de download do agente

x86	Ubuntu Server	`https://s3.region.amazonaws.com/amazon-ssm-region/latest/debian_386/amazon-ssm-agent.deb.sig`
		
		`https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/debian_386/amazon-ssm-agent.deb.sig`
			`amazon-ssm-agent.deb`

Arquitetura	Sistema operacional	URL do arquivo de assinatura	Nome do arquivo de download do agente

ARM64	Amazon Linux 1, Amazon Linux 2, Amazon Linux 2023, CentOS, RHEL	`https://s3.region.amazonaws.com/amazon-ssm-region/latest/linux_arm64/amazon-ssm-agent.rpm.sig`
		
		`https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux_arm64/amazon-ssm-agent.rpm.sig`
			`amazon-ssm-agent.rpm`

Verificar o pacote do SSM Agent em um servidor Linux (v3.3.1802.0 e posteriores)

Antes de começar

Os procedimentos para GPG e RPM nesta seção se aplicam ao SSM Agent versão 3.3.1802.0 e posteriores. Antes de verificar a assinatura do SSM Agent usando o procedimento a seguir, certifique-se de ter baixado o pacote do agente mais recente para seu sistema operacional. Por exemplo, .`https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux_arm64/amazon-ssm-agent.rpm` Para obter mais informações sobre como fazer download de pacotes do SSM Agent, consulte Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux.

Se houver algum motivo para continuar usando a versão 3.3.1611.0 ou anterior do agente, siga as instruções em Verificando o pacote SSM Agent em um servidor Linux (v3.3.1611.0 e anteriores).

GPG

Para verificar o pacote do SSM Agent em um servidor Linux (v3.3.1802.0 e posteriores)

1. Copie a chave pública a seguir e salve-a em um arquivo chamado `amazon-ssm-agent.gpg`.

> **Important**
>
> A chave pública a seguir expira em 15/7/2026 (15 de julho de 2026). O Systems Manager publicará uma nova chave pública neste tópico antes que a antiga expire. Recomendamos assinar o feed RSS deste tópico para receber uma notificação quando a nova chave estiver disponível.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.22 (GNU/Linux)

mQINBGeRNq4BEACrlf5h6Pz+k+M+QCJJ2LfK7d2Tn9J8iJ9qBK2Vwvuxco1rpSO+
KEI3nTeysPuheximps8WOCADX4VlbsKxMZQLjQM4mA26m1Tiw9nAI4kod4bKjiuM
BMUTCD1wfnjH3zQi4kDUdbpfAEMiPgNLVLH85Wf+lhK+Zm+V38DYzLyVj03kX4wK
iG6RMoxzOBZa5gNsVq+j+oCUITGz/URxH713Rgo8WeoEegI0+7iCBLKg+PM0b7GV
2nzkwWJz796HdkqSg8BwXsYaLTrHxa2P1IpwPCisAkyO7gZaMd6Uj69dtMFO+V8a
Qee6b57qGuFKZw7h1Vvc85PbF1Gy/wNIpary57kUHBFUg1vYep/roJuEbJCq97r5
I2liLl4NAyrWb9r/TAVxlXvqM4iZUhxm8GAp0FywMdBr9ZECClKa5HxuVmlm0Wgl
TXoYTOZKeDg6ZoCvyhNxWneCNip74fohXymeFF5L/budhBwy5wuwSniOgTGLo/4C
VgZHWCcN+d0Q3bx/sl2QNqPg5/xzsxEtymXLdVdwLIsLdEQUnIvy8KTs5jol3Dwi
nnEEyhly6wdaw+qDOhkSOT/VnErrSMkYF8VJfa5GjhCBWKw9JVSkaP2CI/VHOgHM
MKROnulq0hRQBR7RmLYt98xu38BHJWMmF8Ga/HJuIxzD1VmkZOPvDDESUwARAQAB
tCdTU00gQWdlbnQgPHNzbS1hZ2VudC1zaWduZXJAYW1hem9uLmNvbT6JAj8EEwEC
ACkFAmeRNq4CGy8FCQLGmIAHCwkIBwMCAQYVCAIJCgsEFgIDAQIeAQIXgAAKCRBR
qOBQ0AUuXTdND/9qldQ1E3dYjBVXOnbhiUQL594bkS5VoEX7D4fZ5UMVZa5pGiz+
husnoRUS9rH1cSeq7aHJu9hSCMuMdvRpuoo0CwLB+7HtzJvAO2M01hcEkUYa6Qdj
njTzP0ZjnoenJmqF9SYmVqAI/VPa9mNQ1OJ+HQ3qh5i6w+FoWlVqEdXjZGrWijub
TqyN33i1Y26t7Os/x8I9fUeNx37y/7Kama8LTdtv9GhWiMVBg2IuVf27HCMYofrQ
m2uCGe61IhtsnhsYaYupmljl+6qgdiuCiS9BAsoIGtqTnu8lnKcGyGz6YnRszN+U
1bNE4w+UFpXWJF8ogpYcghJ06aW/LhjZnQSx3VliLdW8eOJzou41yWmiuL3ZY8eW
KAlD+7eYKS6N6fEJCeNO2VX2lcKtDfaOX+lqGIVyexKayMfpi+0frNzt/92YCpF5
3jkeS77vMMVqKIUiIp1OCGv3XsFpIr6Bt2c2throYPDoQL3zvq6vvG40BKeRQ4tT
Y+5vTc8MeNn3LdzTl9pusxTcKifrJq7f5FIsL2CpAX8uQ+Qz+XWsYQQ5PvyUDtOz
nU/MRZaP6HnqY42bzI9ZlKgXi9IE3MXIwoET9YyzFjkIDvat7SlB4uJCpeIqp/KM
```

```
OIrTMb7paGLYmBU6YqxNBkDWItNG7NeZzyhh/R/Qqb4vJaf4S+ZqD1RZXokCHAQQ
AQIABgUCZ5E2rwAKCRB90Jej2tf1/CdnD/46It+RNoE00TesZK5n2bijH5Eljw0E
4/UpMi1SV6t2zY7lIm7TcKNn18tynJNFqB6YXXOwSbBG/fbN2E9RaoUCZw23TmAv
amuHwrfsDqsHb7zzPF0bISYjqEDLQJj/gtEugUc6XY1dEpFSlWJIOvgryG04cFXI
uD2KY87ya4s1R+sEVAJ14K4RlUCiMmzJdR0NJNYJOwBi1gkLEp6jG86ttiG2U7fY
pE2ibV+c0GeIFq8PIzqqENsn9KBuRH5EcbdBwfnsj2XfM4aR3ZtRIdWXkKkdP9Rs
yU5dTF/Y7XPId5h8/gp00+DMlXFBinQ1jE7A7eDYviEFd1ba8P7dIom3Q3gzKiWu
KTGpnykShs5NvpQmvGUF6JqDHI4RK9s3kLqsNyZkhenJfRBrJ/45fQAuP4CRedkF
7PSfX0Xp7kDnKuyK6wEUEfXXrqmuLGDmigTXblO5qgdyMwkOLjiY9znBZbHoKs76
VplOoNgGnN19i3nuMcPf2npFICJv7kTIyn5Fh7pjWDCahl3U/PwoLjrrlEzpyStU
oXSZrK3kiAADEdSODXJl8KYU0Pb27JbRr1ZbWnxb+O39TOhtssstulkR0v+IDGDQ
rQE1b12sKgcNFSzInzWrNGu4S06WN8DYzlrTZ9aSHj+37ZqpXAevi8WOFXKPV3PA
E6+O8RI2451Dcg==
=aDkv
-----END PGP PUBLIC KEY BLOCK-----
```

2. Importe a chave pública em seu chaveiro e observe o valor de chave retornado.

```
gpg --import amazon-ssm-agent.gpg
```

3. Verifique a impressão digital. Substitua o *valor da chave* pelo valor da etapa anterior. Recomendamos que você use o GPG para verificar a impressão digital mesmo que você use o RPM para verificar o pacote do instalador.

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

Esse comando retorna uma saída semelhante à seguinte:

```
pub   4096R/D0052E5D 2025-01-22 [expires: 2026-07-15]
      Key fingerprint = 4855 A9E6 8332 16D6 A77D  8FE4 51A8 E050 D005 2E5D
uid                  SSM Agent <ssm-agent-signer@amazon.com>
```

A impressão digital deve corresponder ao mostrado a seguir.

```
4855 A9E6 8332 16D6 A77D 8FE4 51A8 E050 D005 2E5D
```

Se a impressão digital não coincidir, não instale o agente. Entre em contato com a AWS Support.

4. Baixe o arquivo de assinatura de acordo com a arquitetura e o sistema operacional da instância, caso ainda não tenha feito isso.

5. Verifique a assinatura do pacote do instalador. Substitua *signature-filename* e *agente-download-filename* pelos valores especificados ao baixar o arquivo de assinatura e o agente, conforme listado na tabela anteriormente neste tópico.

```
gpg --verify signature-filename agent-download-filename
```

Por exemplo, para a arquitetura x86_64 no Amazon Linux 2:

```
gpg --verify amazon-ssm-agent.rpm.sig amazon-ssm-agent.rpm
```

Esse comando retorna uma saída semelhante à seguinte:

```
gpg: Signature made Sat 08 Feb 2025 12:05:08 AM UTC using RSA key ID D0052E5D
gpg: Good signature from "SSM Agent <ssm-agent-signer@amazon.com>"
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: 4855 A9E6 8332 16D6 A77D  8FE4 51A8 E050 D005 2E5D
```

Se a saída inclui a frase `BAD signature`, verifique se você executou o procedimento corretamente. Se você continuar recebendo essa resposta, entre em contato com o Suporte e não instale o agente. A mensagem de aviso sobre a relação de confiança não significa que a assinatura não é válida, apenas que você não verificou a chave pública. Uma chave somente será confiável se você ou alguém em quem você confia a tiver assinado. Se a saída incluir a frase `Can't check signature: No public key`, verifique se você baixou SSM Agent versão 3.1.1141.0 ou posterior.

RPM

Para verificar o pacote do SSM Agent em um servidor Linux (v3.3.1802.0 e posteriores)

1. Copie a chave pública a seguir e salve-a em um arquivo chamado `amazon-ssm-agent.gpg`.

> **Important**
>
> A chave pública a seguir expira em 15/7/2026 (15 de julho de 2026). O Systems Manager publicará uma nova chave pública neste tópico antes que a antiga expire.

Recomendamos assinar o feed RSS deste tópico para receber uma notificação quando a nova chave estiver disponível.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.22 (GNU/Linux)

mQINBGeRNq4BEACrlf5h6Pz+k+M+QCJJ2LfK7d2Tn9J8iJ9qBK2Vwvuxco1rpSO+
KEI3nTeysPuheximps8WOCADX4VlbsKxMZQLjQM4mA26m1Tiw9nAI4kod4bKjiuM
BMUTCD1wfnjH3zQi4kDUdbpfAEMiPgNLVLH85Wf+lhK+Zm+V38DYzLyVj03kX4wK
iG6RMoxzOBZa5gNsVq+j+oCUITGz/URxH713Rgo8WeoEegI0+7iCBLKg+PM0b7GV
2nzkwWJz796HdkqSg8BwXsYaLTrHxa2P1IpwPCisAkyO7gZaMd6Uj69dtMFO+V8a
Qee6b57qGuFKZw7h1Vvc85PbF1Gy/wNIpary57kUHBFUg1vYep/roJuEbJCq97r5
I2liLl4NAyrWb9r/TAVxlXvqM4iZUhxm8GAp0FywMdBr9ZECClKa5HxuVmlm0Wgl
TXoYTOZKeDg6ZoCvyhNxWneCNip74fohXymeFF5L/budhBwy5wuwSniOgTGLo/4C
VgZHWCcN+d0Q3bx/sl2QNqPg5/xzsxEtymXLdVdwLIsLdEQUnIvy8KTs5jol3Dwi
nnEEyhly6wdaw+qDOhkSOT/VnErrSMkYF8VJfa5GjhCBWKw9JVSkaP2CI/VHOgHM
MKROnulq0hRQBR7RmLYt98xu38BHJWMmF8Ga/HJuIxzD1VmkZOPvDDESUwARAQAB
tCdTU00gQWdlbnQgPHNzbS1hZ2VudC1zaWduZXJAYW1hem9uLmNvbT6JAj8EEwEC
ACkFAmeRNq4CGy8FCQLGmIAHCwkIBwMCAQYVCAIJCgsEFgIDAQIeAQIXgAAKCRBR
qOBQ0AUuXTdND/9qldQ1E3dYjBVXOnbhiUQL594bkS5VoEX7D4fZ5UMVZa5pGiz+
husnoRUS9rH1cSeq7aHJu9hSCMuMdvRpuoo0CwLB+7HtzJvAO2M01hcEkUYa6Qdj
njTzP0ZjnoenJmqF9SYmVqAI/VPa9mNQ1OJ+HQ3qh5i6w+FoWlVqEdXjZGrWijub
TqyN33i1Y26t7Os/x8I9fUeNx37y/7Kama8LTdtv9GhWiMVBg2IuVf27HCMYofrQ
m2uCGe61IhtsnhsYaYupmljl+6qgdiuCiS9BAsoIGtqTnu8lnKcGyGz6YnRszN+U
1bNE4w+UFpXWJF8ogpYcghJ06aW/LhjZnQSx3VliLdW8eOJzou41yWmiuL3ZY8eW
KAlD+7eYKS6N6fEJCeNO2VX2lcKtDfaOX+lqGIVyexKayMfpi+0frNzt/92YCpF5
3jkeS77vMMVqKIUiIp1OCGv3XsFpIr6Bt2c2throYPDoQL3zvq6vvG40BKeRQ4tT
Y+5vTc8MeNn3LdzTl9pusxTcKifrJq7f5FIsL2CpAX8uQ+Qz+XWsYQQ5PvyUDtOz
nU/MRZaP6HnqY42bzI9ZlKgXi9IE3MXIwoET9YyzFjkIDvat7SlB4uJCpeIqp/KM
OIrTMb7paGLYmBU6YqxNBkDWItNG7NeZzyhh/R/Qqb4vJaf4S+ZqD1RZXokCHAQQ
AQIABgUCZ5E2rwAKCRB90Jej2tf1/CdnD/46It+RNoE00TesZK5n2bijH5Eljw0E
4/UpMi1SV6t2zY7lIm7TcKNn18tynJNFqB6YXXOwSbBG/fbN2E9RaoUCZw23TmAv
amuHwrfsDqsHb7zzPF0bISYjqEDLQJj/gtEugUc6XY1dEpFSlWJIOvgryG04cFXI
uD2KY87ya4s1R+sEVAJ14K4RlUCiMmzJdR0NJNYJOwBi1gkLEp6jG86ttiG2U7fY
pE2ibV+c0GeIFq8PIzqqENsn9KBuRH5EcbdBwfnsj2XfM4aR3ZtRIdWXkKkdP9Rs
yU5dTF/Y7XPId5h8/gp00+DMlXFBinQ1jE7A7eDYviEFd1ba8P7dIom3Q3gzKiWu
KTGpnykShs5NvpQmvGUF6JqDHI4RK9s3kLqsNyZkhenJfRBrJ/45fQAuP4CRedkF
7PSfX0Xp7kDnKuyK6wEUEfXXrqmuLGDmigTXblO5qgdyMwkOLjiY9znBZbHoKs76
VplOoNgGnN19i3nuMcPf2npFICJv7kTIyn5Fh7pjWDCahl3U/PwoLjrrlEzpyStU
oXSZrK3kiAADEdSODXJl8KYU0Pb27JbRr1ZbWnxb+O39TOhtssstulkR0v+IDGDQ
rQE1b12sKgcNFSzInzWrNGu4S06WN8DYzlrTZ9aSHj+37ZqpXAevi8WOFXKPV3PA
```

```
E6+08RI2451Dcg==
=aDkv
-----END PGP PUBLIC KEY BLOCK-----
```

2. Importe a chave pública em seu chaveiro e observe o valor de chave retornado.

```
rpm --import amazon-ssm-agent.gpg
```

3. Verifique a impressão digital. Recomendamos que você use o GPG para verificar a impressão digital mesmo que você use o RPM para verificar o pacote do instalador.

```
rpm -qa gpg-pubkey --qf '%{Description}' | gpg --with-fingerprint | grep -A 1
 "ssm-agent-signer@amazon.com"
```

Esse comando retorna uma saída semelhante à seguinte:

```
pub  4096R/D0052E5D 2025-01-22 SSM Agent <ssm-agent-signer@amazon.com>
      Key fingerprint = 4855 A9E6 8332 16D6 A77D  8FE4 51A8 E050 D005 2E5D
```

A impressão digital deve corresponder ao mostrado a seguir.

```
4855 A9E6 8332 16D6 A77D 8FE4 51A8 E050 D005 2E5D
```

Se a impressão digital não coincidir, não instale o agente. Entre em contato com a AWS Support.

4. Verifique a assinatura do pacote do instalador. Substitua *agent-download-filename* pelos valores especificados ao baixar o agente, conforme listado anteriormente na tabela.

```
rpm --checksig agent-download-filename
```

Por exemplo, para a arquitetura x86_64 no Amazon Linux 2:

```
rpm --checksig amazon-ssm-agent.rpm
```

Esse comando retorna uma saída semelhante à seguinte:

```
amazon-ssm-agent.rpm: rsa sha1 md5 OK
```

Se o pgp estiver ausente da saída e você tiver importado a chave pública, então o agente não será assinado. Se a saída contiver a frase `NOT OK (MISSING KEYS: (MD5) key-id)`, verifique se você executou o procedimento corretamente e verifique se você baixou SSM Agent versão 3.1.1141.0 ou posterior. Se você continuar recebendo essa resposta, entre em contato com o Suporte e não instale o atendente.

Verificando o pacote SSM Agent em um servidor Linux (v3.3.1611.0 e anteriores)

Antes de começar

Os procedimentos para GPG e RPM nesta seção se aplicam à versão SSM Agent 3.3.1611.0 e anteriores. Recomendamos sempre usar a versão mais recente do agente. Para mais informações, consulte Verificar o pacote do SSM Agent em um servidor Linux (v3.3.1802.0 e posteriores). No entanto, se houver algum motivo para continuar usando a versão 3.3.1611.0 ou anterior do agente, siga as instruções em um dos procedimentos a seguir.

GPG

Para verificar o pacote SSM Agent em um servidor Linux (v3.3.1611.0 e anteriores)

1. Copie as chaves públicas a seguir e salve-as em um arquivo chamado `amazon-ssm-agent.gpg`.

> **Important**
>
> A chave pública mostrada abaixo expirou em 17/2/2025 (17 de fevereiro de 2025) e funcionará para a versão 3.3.1611.0 e versões anteriores até 3.2.1542.0, mas somente se tiver sido usada anteriormente para verificar a assinatura do agente. O Systems Manager publicará uma nova chave pública neste tópico antes que a antiga expire. Recomendamos assinar o feed RSS deste tópico para receber uma notificação quando a nova chave estiver disponível.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.22 (GNU/Linux)

mQENBGTtIoIBCAD2M1aoGIE0FXynAHM/jtuvdAVVaX3Q4ZejTqrX+Jq8ElAMhxyO
GzHu2CDtCYxtVxXK3unptLVt2kGgJwNbhYC393jDeZx5dCda4Nk2YXX1UK3P461i
```

```
axuuXRzMYvfM4RZn+7bJTu635tA07q9Xm6MGD4TCTvsjBfViOxbrxOg5ozWbJdSw
fSR8MwUrRfmFpAefRlYfCEuZ8FHywa9U6jLeWt2O/kqrZliJOAGjGzXtB7EZkqKb
faCCxikjjvhF1awdEqSK4DQorC/OvQc4I5kP5y2CJbtXvXO73QH2yE75JMDIIx9x
rOsIRUoSfK3UrWaOVuAnEEn5ueKzZNqGG1J1ABEBAAG0J1NTTSBBZ2VudCA8c3Nt
LWFnZW50LXNpZ25lckBhbWF6b24uY29tPokBPwQTAQIAKQUCZO0iggIbLwUJAsaY
gAcLCQgHAwIBBhUIAgkKCwQWAgMBAh4BAheAAAoJELwfSVyX3QTt+icH/A//tJsW
I+7Ay8FGJh8dJPNy++HIBjVSFdGNJFWNbw1Z8uZcazHEcUCH3FhW4CLQLTZ3OVPz
qvFwzDtRDVIN/Y9EGDhLMFvimrE+/z4olWsJ5DANf6BnX8I5UNIcRt5d8SWH1BEJ
2FWIBZFgKyTDI6XzRC5x4ahtgpOVAGeeKDehs+wh6Ga4W0/K4GsviP1Kyr+Ic2br
NAIq0q0IHyN1q9zam3Y0+jKwEuNmTj+Bjyzshyv/X8S0JWWoXJhkexkOvWeBYNNt
5wI4QcSteyfIzp6KlQF8q11Hzz9D9WaPfcBEYyhq7vLEARobkbQMBzpkmaZua241
0RaWG50HRvrgm4aJAhwEEAECAAYFAmTtIoMACgkQfdCXo9rX9fwwqBAAzkTgYJ38
sWgxpn7Ux/81F2BWR1sVkmP79i++fXyJlKI8xtcJFQZhzeUos69KBUCy7mgx5bYU
P7NA5o9DUbwz/QS0i1Cqm4+jtFlX0MXe4FikXcqfDPnnzN8mVB2H+fa43iHR1PuH
GgUWuNdxzSoIYRmLZXWmeN5YXPcmixlhLzcE2TOQn1mOKcu2fKdLtBQ8KiEkmjiu
naoLxnUcyk1zMhaha+LzEkQdOyasix0ggylN2ViWVnlmfy0niuXDxW0qZWPdLStF
OODiX3iqGmkH3rDfy6nvxxBR4GIs+MGD72fpWzzrINDgkGI2i2t1+0AX/mps3aTy
+ftlgrim8stYWB58XXDAb0vad06sNye5/zDzfr0I9HupJrTzFhaYJQjWPaSlINto
LDJnBXohiUIPRYRcy/k012oFHDWZHT3H6CyjK9UD5UlxA9H7dsJurANs6FOVRe+7
34uJyxDZ/W7zLG4AVG0zxibrUSoaJxwcOjVPVsQAlrwG/GTs7tcAccsJqbJ1Py/w
9AgJl8VU2qc8POsHNXk348gjP7C8PDnGMpZFzr9f5INctRushpiv7onX+aWJVX7T
n2uX/TP3LCyH/MsrNJrJOQnMYFRLQitciP0E+F+eA3v9CY6mDuyb8JSx5HuGGUsG
S4bKBOcA8vimEpwPoT8CE7fdsZ3Qkwdu+pw=
=zr5w
-----END PGP PUBLIC KEY BLOCK-----
```

2. Importe a chave pública em seu chaveiro e observe o valor de chave retornado.

```
gpg --import amazon-ssm-agent.gpg
```

3. Verifique a impressão digital. Substitua o *valor da chave* pelo valor da etapa anterior. Recomendamos que você use o GPG para verificar a impressão digital mesmo que você use o RPM para verificar o pacote do instalador.

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

Esse comando retorna uma saída semelhante à seguinte:

```
pub   2048R/97DD04ED 2023-08-28 [expired: 2025-02-17]
      Key fingerprint = DE92 C7DA 3E56 E923 31D6 2A36 BC1F 495C 97DD 04ED
uid                  SSM Agent <ssm-agent-signer@amazon.com>
```

A impressão digital deve corresponder ao mostrado a seguir.

```
DE92 C7DA 3E56 E923 31D6 2A36 BC1F 495C 97DD 04ED
```

Se a impressão digital não coincidir, não instale o agente. Entre em contato com a AWS Support.

4. Baixe o arquivo de assinatura de acordo com a arquitetura e o sistema operacional da instância, caso ainda não tenha feito isso.
5. Verifique a assinatura do pacote do instalador. Substitua *signature-filename* e *agente-download-filename* pelos valores especificados ao baixar o arquivo de assinatura e o agente, conforme listado na tabela anteriormente neste tópico.

```
gpg --verify signature-filename agent-download-filename
```

Por exemplo, para a arquitetura x86_64 no Amazon Linux 2:

```
gpg --verify amazon-ssm-agent.rpm.sig amazon-ssm-agent.rpm
```

Esse comando retorna uma saída semelhante à seguinte:

```
gpg: Signature made Fri 10 Jan 2025 01:54:18 AM UTC using RSA key ID 97DD04ED
gpg: Good signature from "SSM Agent <ssm-agent-signer@amazon.com>"
gpg: Note: This key has expired!
Primary key fingerprint: DE92 C7DA 3E56 E923 31D6  2A36 BC1F 495C 97DD 04ED
```

Se a saída inclui a frase `BAD signature`, verifique se você executou o procedimento corretamente. Se você continuar recebendo essa resposta, entre em contato com o Suporte e não instale o agente. A mensagem de aviso sobre a relação de confiança não significa que a assinatura não é válida, apenas que você não verificou a chave pública. Uma chave somente será confiável se você ou alguém em quem você confia a tiver assinado. Se a saída incluir a frase `Can't check signature: No public key`, verifique se você baixou SSM Agent versão 3.1.1141.0 ou posterior.

RPM

Para verificar o pacote SSM Agent em um servidor Linux (v3.3.1611.0 e anteriores)

1. Copie a chave pública a seguir e salve-a em um arquivo chamado `amazon-ssm-agent.gpg`.

> **Important**
>
> A chave pública mostrada abaixo expirou em 17/2/2025 (17 de fevereiro de 2025) e funcionará para a versão 3.3.1611.0 e versões anteriores até 3.2.1542.0, mas somente se tiver sido usada anteriormente para verificar a assinatura do agente. O Systems Manager publicará uma nova chave pública neste tópico antes que a antiga expire. Recomendamos assinar o feed RSS deste tópico para receber uma notificação quando a nova chave estiver disponível.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.22 (GNU/Linux)

mQENBGTtIoIBCAD2M1aoGIE0FXynAHM/jtuvdAVVaX3Q4ZejTqrX+Jq8ElAMhxyO
GzHu2CDtCYxtVxXK3unptLVt2kGgJwNbhYC393jDeZx5dCda4Nk2YXX1UK3P461i
axuuXRzMYvfM4RZn+7bJTu635tA07q9Xm6MGD4TCTvsjBfViOxbrxOg5ozWbJdSw
fSR8MwUrRfmFpAefRlYfCEuZ8FHywa9U6jLeWt2O/kqrZliJOAGjGzXtB7EZkqKb
faCCxikjjvhF1awdEqSK4DQorC/OvQc4I5kP5y2CJbtXvXO73QH2yE75JMDIIx9x
rOsIRUoSfK3UrWaOVuAnEEn5ueKzZNqGG1J1ABEBAAG0J1NTTSBBZ2VudCA8c3Nt
LWFnZW50LXNpZ25lckBhbWF6b24uY29tPokBPwQTAQIAKQUCZO0iggIbLwUJAsaY
gAcLCQgHAwIBBhUIAgkKCwQWAgMBAh4BAheAAAoJELwfSVyX3QTt+icH/A//tJsW
I+7Ay8FGJh8dJPNy++HIBjVSFdGNJFWNbw1Z8uZcazHEcUCH3FhW4CLQLTZ3OVPz
qvFwzDtRDVIN/Y9EGDhLMFvimrE+/z4olWsJ5DANf6BnX8I5UNIcRt5d8SWH1BEJ
2FWIBZFgKyTDI6XzRC5x4ahtgpOVAGeeKDehs+wh6Ga4W0/K4GsviP1Kyr+Ic2br
NAIq0q0IHyN1q9zam3Y0+jKwEuNmTj+Bjyzshyv/X8S0JWWoXJhkexkOvWeBYNNt
5wI4QcSteyfIzp6KlQF8q11Hzz9D9WaPfcBEYyhq7vLEARobkbQMBzpkmaZua241
0RaWG50HRvrgm4aJAhwEEAECAAYFAmTtIoMACgkQfdCXo9rX9fwwqBAAzkTgYJ38
sWgxpn7Ux/81F2BWR1sVkmP79i++fXyJlKI8xtcJFQZhzeUos69KBUCy7mgx5bYU
P7NA5o9DUbwz/QS0i1Cqm4+jtFlX0MXe4FikXcqfDPnnzN8mVB2H+fa43iHR1PuH
GgUWuNdxzSoIYRmLZXWmeN5YXPcmixlhLzcE2TOQn1mOKcu2fKdLtBQ8KiEkmjiu
naoLxnUcyk1zMhaha+LzEkQdOyasix0ggylN2ViWVnlmfy0niuXDxW0qZWPdLStF
OODiX3iqGmkH3rDfy6nvxxBR4GIs+MGD72fpWzzrINDgkGI2i2t1+0AX/mps3aTy
+ftlgrim8stYWB58XXDAb0vad06sNye5/zDzfr0I9HupJrTzFhaYJQjWPaSlINto
LDJnBXohiUIPRYRcy/k012oFHDWZHT3H6CyjK9UD5UlxA9H7dsJurANs6FOVRe+7
```

```
34uJyxDZ/W7zLG4AVG0zxibrUSoaJxwcOjVPVsQAlrwG/GTs7tcAccsJqbJ1Py/w
9AgJl8VU2qc8POsHNXk348gjP7C8PDnGMpZFzr9f5INctRushpiv7onX+aWJVX7T
n2uX/TP3LCyH/MsrNJrJOQnMYFRLQitciP0E+F+eA3v9CY6mDuyb8JSx5HuGGUsG
S4bKBOcA8vimEpwPoT8CE7fdsZ3Qkwdu+pw=
=zr5w
-----END PGP PUBLIC KEY BLOCK-----
```

2. Importe a chave pública em seu chaveiro e observe o valor de chave retornado.

```
rpm --import amazon-ssm-agent.gpg
```

3. Verifique a impressão digital. Recomendamos que você use o GPG para verificar a impressão digital mesmo que você use o RPM para verificar o pacote do instalador.

```
rpm -qa gpg-pubkey --qf '%{Description}' | gpg --with-fingerprint | grep -A 1
 "ssm-agent-signer@amazon.com"
```

Esse comando retorna uma saída semelhante à seguinte:

```
pub  2048R/97DD04ED 2023-08-28 SSM Agent <ssm-agent-signer@amazon.com>
      Key fingerprint = DE92 C7DA 3E56 E923 31D6 2A36 BC1F 495C 97DD 04ED
```

A impressão digital deve corresponder ao mostrado a seguir.

```
DE92 C7DA 3E56 E923 31D6 2A36 BC1F 495C 97DD 04ED
```

Se a impressão digital não coincidir, não instale o agente. Entre em contato com a AWS Support.

4. Verifique a assinatura do pacote do instalador. Substitua *agent-download-filename* pelos valores especificados ao baixar o agente, conforme listado anteriormente na tabela.

```
rpm --checksig agent-download-filename
```

Por exemplo, para a arquitetura x86_64 no Amazon Linux 2:

```
rpm --checksig amazon-ssm-agent.rpm
```

Esse comando retorna uma saída semelhante à seguinte:

```
amazon-ssm-agent.rpm: rsa sha1 md5 OK
```

Se o pgp estiver ausente da saída e você tiver importado a chave pública, então o agente não será assinado. Se a saída contiver a frase `NOT OK (MISSING KEYS: (MD5) key-id)`, verifique se você executou o procedimento corretamente e verifique se você baixou SSM Agent versão 3.1.1141.0 ou posterior. Se você continuar recebendo essa resposta, entre em contato com o Suporte e não instale o atendente.

Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux

Antes de instalar manualmente o agente do AWS Systems Manager (SSM Agent) em um sistema operacional Linux do Amazon Elastic Compute Cloud (Amazon EC2), consulte as informações a seguir.

URLs do arquivo de instalação do SSM Agent

É possível acessar os arquivos de instalação do SSM Agent que estão armazenados em qualquer Região da AWS comercial. Também fornecemos arquivos de instalação em um bucket do Amazon Simple Storage Service (Amazon S3) disponível globalmente que pode ser usado como alternativa ou fonte de backup de arquivos.

Se você estiver instalando manualmente o agente em uma ou duas instâncias, poderá usar os comandos nos procedimentos de Instalação rápida que fornecemos para ajudar a economizar tempo. Os comandos fornecidos nestes procedimentos também podem ser passados para instâncias do Amazon EC2 como scripts por meio de dados do usuário.

Se você estiver criando um script ou modelo a ser usado para instalar o agente em várias instâncias, recomendamos utilizar os arquivos de instalação na Região da AWS em que você está localizado geograficamente ou próximo a ela. Para instalações em massa, isso pode aumentar a velocidade dos downloads e reduzir a latência. Nesses casos, recomendamos usar os procedimentos descritos em Criar comandos de instalação personalizados nos tópicos de instalação.

Amazon Machine Images com o agente pré-instalado

O SSM Agent é pré-instalado em algumas Amazon Machine Images (AMIs) fornecidas pela AWS. Para mais informações, consulte Encontrar AMIs com o SSM Agent pré-instalado.

Instalação em outros tipos de máquinas

Se for necessário instalar o agente em um servidor ou em uma máquina virtual (VM) on-premises para que ele possa ser usado com o Systems Manager, consulte Como instalar o SSM Agent para nós híbridos do Linux. Para mais informações sobre como instalar o agente em dispostivos de borda, consulte Gerenciar dispositivos de borda com o Systems Manager.

Manter o agente atualizado

Uma versão atualizada do SSM Agent é lançada sempre que novas ferramentas são adicionadas ao Systems Manager ou sempre que atualizações são feitas nas ferramentas existentes. Deixar de usar a versão mais recente do agente pode impedir que seu nó gerenciado use várias ferramentas do Systems Manager. Por isso, recomendamos automatizar o processo de manter o SSM Agent atualizado em suas máquinas. Para mais informações, consulte Automatizar atualizações do SSM Agent. Inscreva-se na página Notas de versão do SSM Agent no GitHub para receber notificações sobre atualizações do SSM Agent.

Escolher o sistema operacional

Para visualizar o procedimento de instalação manual do SSM Agent no sistema operacional especificado, escolha um link na seguinte lista:

> **Note**
>
> Para obter uma lista das versões com suporte de cada um dos sistemas operacionais a seguir, consulte Sistemas operacionais compatíveis com o Systems Manager.

- AlmaLinux
- Amazon Linux 2 e Amazon Linux 2023
- Amazon Linux 1
- CentOS
- CentOS Stream
- Debian Server
- Oracle Linux
- Red Hat Enterprise Linux
- Rocky Linux

- SUSE Linux Enterprise Server
- Ubuntu Server

Desinstalar o SSM Agent de instâncias do Linux

Use o gerenciador de pacotes do seu sistema operacional para desinstalar o SSM Agent das instâncias do Linux. Dependendo do sistema operacional, o comando de desinstalação será semelhante ao seguinte exemplo:

```
sudo dpkg -r amazon-ssm-agent
```

Instalar o SSM Agent manualmente em instâncias do AlmaLinux

Use as informações desta seção para obter a ajuda para instalar ou reinstalar o SSM Agent manualmente em uma instância do AlmaLinux.

Antes de começar

Antes de instalar o SSM Agent em uma instância do AlmaLinux, observe o seguinte:

- Verifique se o Python 3 está instalado na instância do AlmaLinux. Isso é necessário para que o SSM Agent funcione corretamente.
- Para obter informações importantes aplicáveis à instalação do SSM Agent em todos os sistemas operacionais baseados em Linux, consulte Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux.

Tópicos

- Comandos de instalação rápida do SSM Agent no AlmaLinux
- Criar comandos de instalação do agente personalizados para o AlmaLinux em sua região

Comandos de instalação rápida do SSM Agent no AlmaLinux

Use as etapas a seguir para instalar o SSM Agent manualmente em uma única instância. Este procedimento usa arquivos de instalação disponíveis globalmente.

Antes de começar

Antes de instalar o SSM Agent em uma instância do AlmaLinux, observe o seguinte:

- Verifique se o Python 3 está instalado na instância do AlmaLinux. Isso é necessário para que o SSM Agent funcione corretamente.

Para instalar o SSM Agent no AlmaLinux

1. Conecte à sua instância do AlmaLinux usando o método de sua preferência, como SSH.
2. Copie o comando correspondente à arquitetura da instância e execute-o na instância.

> **Note**
>
> Mesmo que os URLs nos comandos a seguir incluam um diretório `ec2-downloads-windows`, estes são os arquivos de instalação global corretos para o AlmaLinux.

Instâncias x86_64

```
sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_amd64/amazon-ssm-agent.rpm
```

Instâncias ARM64

```
sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_arm64/amazon-ssm-agent.rpm
```

3. (Recomendado) Execute o comando a seguir para verificar se o agente está em execução.

```
sudo systemctl status amazon-ssm-agent
```

Na maioria dos casos, o comando informa que o agente está em execução, conforme mostrado no exemplo a seguir.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendo>
   Active: active (running) since Tue 2022-04-19 16:40:41 UTC; 9s ago
 Main PID: 4898 (amazon-ssm-agen)
    Tasks: 14 (limit: 4821)
   Memory: 34.6M
   CGroup: /system.slice/amazon-ssm-agent.service
           ##4898 /usr/bin/amazon-ssm-agent
```

```
           ##4954 /usr/bin/ssm-agent-worker
            --truncated--
```

Em casos raros, o comando informa que o agente está instalado, mas não está em execução, conforme mostrado no exemplo a seguir.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendo>
   Active: inactive (dead) since Tue 2022-04-19 16:42:05 UTC; 2s ago
            --truncated--
```

Para ativar o agente nesses casos, execute os comandos a seguir.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Criar comandos de instalação do agente personalizados para o AlmaLinux em sua região

Para instalar o SSM Agent em várias instâncias usando um script ou modelo, recomendamos usar arquivos de instalação armazenados no Região da AWS em que você está trabalhando.

Para os comandos a seguir, fornecemos exemplos que usam um bucket do S3 acessível ao público na região Leste dos EUA (Ohio) (`us-east-2`).

> **Tip**
>
> Também é possível substituir um URL global no procedimento Comandos de instalação rápida do SSM Agent no AlmaLinux descrito anteriormente neste tópico por um URL regional personalizado construído por você.

No comando a seguir, substitua *`region`* por suas próprias informações. Para ver uma lista dos valores de *`região`* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

x86_64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_amd64/amazon-ssm-agent.rpm
```

ARM64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Instalar o SSM Agent manualmente em instâncias do Amazon Linux 2 e do Amazon Linux 2023

> **Important**
>
> Este tópico fornece comandos para trabalhar com o SSM Agent em instâncias do Amazon Linux 2 e do Amazon Linux 2023. Alguns desses comandos não são compatíveis com instâncias do Amazon Linux 1. Antes de continuar, verifique se você está visualizando o tópico correto para seu tipo de instância. Para executar comandos nas instâncias do Amazon Linux 1, consulte Instalar o SSM Agent manualmente em instâncias do Amazon Linux 1.

Na maioria dos casos, as Amazon Machine Images (AMIs) para Amazon Linux 2 e Amazon Linux 2023 fornecidas pela AWS têm o AWS Systems Manager Agent (SSM Agent) pré-instalado por padrão. Para obter mais informações, consulte Encontrar AMIs com o SSM Agent pré-instalado.

Se o SSM Agent não estiver pré-instalado em uma nova instância do Amazon Linux 2 ou do Amazon Linux 2023, ou se for necessário reinstalar o agente manualmente, use as informações desta página para obter ajuda.

Antes de começar

Antes de instalar o SSM Agent em uma instância do Amazon Linux 2 e do Amazon Linux 2023, observe o seguinte:

- Para obter informações importantes aplicáveis à instalação do SSM Agent em todos os sistemas operacionais baseados em Linux, consulte Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux.
- Se você usar um comando `yum` para atualizar o SSM Agent em um nó gerenciado após o agente ter sido instalado ou atualizado usando o documento `AWS-UpdateSSMAgent` do SSM, você poderá ver a seguinte mensagem: `"Warning: RPMDB altered outside of yum."` Esta mensagem é esperada e pode ser ignorada com segurança.

Tópicos

- Comandos de instalação rápida do SSM Agent no Amazon Linux 2 ou Amazon Linux 2023
- Criar comandos de instalação do agente personalizados para Amazon Linux 2 ou Amazon Linux 2023 em sua região

Comandos de instalação rápida do SSM Agent no Amazon Linux 2 ou Amazon Linux 2023

Use as etapas a seguir para instalar o SSM Agent manualmente em uma única instância. Este procedimento usa arquivos de instalação disponíveis globalmente.

Para instalar o SSM Agent no Amazon Linux 2 ou no Amazon Linux 2023 usando comandos rápidos de copiar e colar

1. Conecte à sua instância do Amazon Linux 2 ou do Amazon Linux 2023 usando o método de sua preferência, como SSH.
2. Copie o comando correspondente à arquitetura da instância e execute-o na instância.

> **Note**
>
> Mesmo que os URLs nos comandos a seguir incluam um diretório `ec2-downloads-windows`, estes são os arquivos de instalação global corretos para o Amazon Linux 2 e Amazon Linux 2023.

x86_64

```
sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_amd64/amazon-ssm-agent.rpm
```

ARM64

```
sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_arm64/amazon-ssm-agent.rpm
```

3. (Recomendado) Execute o comando a seguir para verificar se o agente está em execução.

```
sudo systemctl status amazon-ssm-agent
```

Na maioria dos casos, o comando informa que o agente está em execução, conforme mostrado no exemplo a seguir.

```
amazon-ssm-agent.service - amazon-ssm-agent
Loaded: loaded (/usr/lib/systemd/system/amazon-ssm-agent.service; enabled; vendor
 preset: enabled)
Active: active (running) since Wed 2021-10-20 19:09:29 UTC; 4min 6s ago
            --truncated--
```

Em casos raros, o comando informa que o agente está instalado, mas não está em execução, conforme mostrado no exemplo a seguir.

```
amazon-ssm-agent.service - amazon-ssm-agent
Loaded: loaded (/usr/lib/systemd/system/amazon-ssm-agent.service; enabled; vendor
 preset: enabled)
Active: inactive (dead) since Wed 2021-10-20 22:16:41 UTC; 18s ago
            --truncated--
```

Para ativar o agente nesses casos, execute o comando a seguir.

```
sudo systemctl start amazon-ssm-agent
```

Criar comandos de instalação do agente personalizados para Amazon Linux 2 ou Amazon Linux 2023 em sua região

Para instalar o SSM Agent em várias instâncias usando um script ou modelo, recomendamos usar arquivos de instalação armazenados no Região da AWS em que você está trabalhando.

Para os comandos a seguir, fornecemos exemplos que usam um bucket do S3 acessível ao público na região Leste dos EUA (Ohio) (`us-east-2`).

> **Tip**
>
> Também é possível substituir um URL global no procedimento Comandos de instalação rápida do SSM Agent no Amazon Linux 1 descrito anteriormente neste tópico por um URL regional personalizado construído por você.

No comando a seguir, substitua *`region`* por suas próprias informações. Para ver uma lista dos valores de *`região`* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

x86_64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_amd64/amazon-ssm-agent.rpm
```

ARM64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Instalar o SSM Agent manualmente em instâncias do Amazon Linux 1

> **Important**
>
> O Amazon Linux 1 chegou ao fim do ciclo de suporte padrão em 31 de dezembro de 2020 e atingiu o fim da vida útil em 31 de dezembro de 2023, conforme anunciado em Atualização sobre o fim da vida útil do Amazon Linux AMI no Blog de notícias da AWS. A AWS não fornece mais Amazon Machine Images (AMIs) para esse sistema operacional. No entanto, a AWS Systems Manager continua fornecendo suporte para instâncias existentes do Amazon Linux 1.
>
> Este tópico fornece comandos para trabalhar com o SSM Agent em instâncias do Amazon Linux 1. Alguns desses comandos não são compatíveis com instâncias do Amazon Linux 2 e do Amazon Linux 2023. Antes de continuar, verifique se você está visualizando o tópico correto para seu tipo de instância. Para executar comandos nas instâncias do Amazon Linux 2 ou Amazon Linux 2023, consulte Instalar o SSM Agent manualmente em instâncias do Amazon Linux 2 e do Amazon Linux 2023.

Na maioria dos casos, as Amazon Machine Images (AMIs) para Amazon Linux 1 fornecidas pela AWS têm o AWS Systems Manager Agent (SSM Agent) pré-instalado por padrão. Para obter mais informações, consulte Encontrar AMIs com o SSM Agent pré-instalado.

Se for necessário reinstalar o agente manualmente no Amazon Linux 1, use as informações nesta página para obter ajuda.

Antes de começar

Antes de instalar o SSM Agent em uma instância do Amazon Linux 1, observe o seguinte:

- Para obter informações importantes aplicáveis à instalação do SSM Agent em todos os sistemas operacionais baseados em Linux, consulte Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux.

- Se você usar um comando `yum` para atualizar o SSM Agent em um nó gerenciado após o agente ter sido instalado ou atualizado usando o documento `AWS-UpdateSSMAgent` do SSM, você poderá ver a seguinte mensagem: `"Warning: RPMDB altered outside of yum."` Esta mensagem é esperada e pode ser ignorada com segurança.

Tópicos

- Comandos de instalação rápida do SSM Agent no Amazon Linux 1
- Criar comandos de instalação do agente personalizados para o Amazon Linux 1 em sua região

Comandos de instalação rápida do SSM Agent no Amazon Linux 1

Use as etapas a seguir para instalar o SSM Agent manualmente em uma única instância. Este procedimento usa arquivos de instalação disponíveis globalmente.

Para instalar o SSM Agent no Amazon Linux 1 usando comandos rápidos de copiar e colar

1. Conecte à sua instância do Amazon Linux 1 usando o método preferido, como SSH.
2. Copie o comando correspondente à arquitetura da instância e execute-o na instância.

> **Note**
>
> Mesmo que os URLs nos comandos a seguir incluam um diretório `ec2-downloads-windows`, estes são os arquivos de instalação global corretos para o Amazon Linux 1.

x86_64

```
sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_amd64/amazon-ssm-agent.rpm
```

x86

```
sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_386/amazon-ssm-agent.rpm
```

ARM64

```
sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_arm64/amazon-ssm-agent.rpm
```

3. (Recomendado) Execute o comando para arquitetura da sua instância para verificar se o agente está em execução.

   x86_64 e x86

   ```
   sudo status amazon-ssm-agent
   ```

   ARM64

   ```
   sudo systemctl status amazon-ssm-agent
   ```

   Na maioria dos casos, o comando informa que o agente está em execução, conforme mostrado nos exemplos a seguir.

   x86_64 e x86

   ```
   amazon-ssm-agent start/running, process 12345
   ```

   ARM64

   ```
   amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/usr/lib/systemd/system/amazon-ssm-agent.service; enabled;
    vendor preset: enabled)
   Active: active (running) since Wed 2021-10-20 19:09:29 UTC; 4min 6s ago
              --truncated--
   ```

   Na casos raros, o comando informa que o agente está instalado, mas não está em execução, conforme mostrado nos exemplos a seguir.

   x86_64 e x86

   ```
   amazon-ssm-agent stop/waiting
   ```

ARM64

```
amazon-ssm-agent.service - amazon-ssm-agent
Loaded: loaded (/usr/lib/systemd/system/amazon-ssm-agent.service; enabled;
 vendor preset: enabled)
Active: inactive (dead) since Wed 2021-10-20 22:16:41 UTC; 18s ago
            --truncated--
```

Para ativar o agente nesses casos, execute o comando correspondente à arquitetura da instância.

x86_64 e x86

```
sudo start amazon-ssm-agent
```

ARM64

```
sudo systemctl start amazon-ssm-agent
```

Criar comandos de instalação do agente personalizados para o Amazon Linux 1 em sua região

Para instalar o SSM Agent em várias instâncias usando um script ou modelo, recomendamos usar arquivos de instalação armazenados no Região da AWS em que você está trabalhando.

Para os comandos a seguir, fornecemos exemplos que usam um bucket do S3 acessível ao público na região Leste dos EUA (Ohio) (`us-east-2`).

> **Tip**
>
> Também é possível substituir um URL global no procedimento Comandos de instalação rápida do SSM Agent no Amazon Linux 1 descrito anteriormente neste tópico por um URL regional personalizado construído por você.

No comando a seguir, substitua *`region`* por suas próprias informações. Para ver uma lista dos valores de *`região`* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

x86_64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_amd64/amazon-ssm-agent.rpm
```

x86

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_386/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_386/amazon-ssm-agent.rpm
```

ARM64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Instalar o SSM Agent manualmente em instâncias do CentOS

As Amazon Machine Images (AMIs) para CentOS fornecidas pela AWS não têm o AWS Systems Manager Agent (SSM Agent) pré-instalado por padrão. Para ver uma lista de AMIs gerenciadas pela AWS nas quais o agente pode estar pré-instalado, consulte Encontrar AMIs com o SSM Agent pré-instalado.

Use as informações nesta seção para obter ajuda para instalar ou reinstalar o SSM Agent manualmente em uma instância do CentOS.

Antes de começar

Antes de instalar o SSM Agent em uma instância do CentOS, observe o seguinte:

- Para obter informações importantes aplicáveis à instalação do SSM Agent em todos os sistemas operacionais baseados em Linux, consulte Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux.
- Se você usar um comando `yum` para atualizar o SSM Agent em um nó gerenciado após o agente ter sido instalado ou atualizado usando o documento `AWS-UpdateSSMAgent` do SSM, você poderá ver a seguinte mensagem: `"Warning: RPMDB altered outside of yum."` Esta mensagem é esperada e pode ser ignorada com segurança.

Tópicos

- Instalar o SSM Agent no CentOS 8.x
- Instalar o SSM Agent no CentOS 7.x
- Instalar o SSM Agent no CentOS 6.x

Instalar o SSM Agent no CentOS 8.x

As Amazon Machine Images (AMIs) para CentOS 8 fornecidas pela AWS não têm o AWS Systems Manager Agent (SSM Agent) pré-instalado por padrão. Use as informações nesta página para obter a ajuda para instalar ou reinstalar o agente em uma instância do CentOS 8.

Antes de começar

Antes de instalar o SSM Agent em uma instância do CentOS 8, observe o seguinte:

- Verifique se o Python 2 ou o Python 3 está instalado na instância do CentOS 8. Isso é necessário para que o SSM Agent funcione corretamente.

Tópicos

- Comandos de instalação rápida do SSM Agent no Cent OS 8
- Criar comandos de instalação do agente personalizados para o CentOS 8 em sua região

Comandos de instalação rápida do SSM Agent no Cent OS 8

Use as etapas a seguir para instalar o SSM Agent manualmente em uma única instância. Este procedimento usa arquivos de instalação disponíveis globalmente.

Como instalar o SSM Agent no CentOS 8.x

1. Conecte à sua instância do CentOS 8 usando o método preferido, como SSH.
2. Copie o comando correspondente à arquitetura da instância e execute-o na instância.

> **Note**
>
> Mesmo que os URLs nos comandos a seguir incluam um diretório `ec2-downloads-windows`, estes são os arquivos de instalação global corretos para o CentOS 8.

x86_64Instâncias do

```
sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_amd64/amazon-ssm-agent.rpm
```

Instâncias ARM64

```
sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_arm64/amazon-ssm-agent.rpm
```

3. (Recomendado) Execute o comando a seguir para verificar se o agente está em execução.

```
sudo systemctl status amazon-ssm-agent
```

Na maioria dos casos, o comando informa que o agente está em execução, conforme mostrado no exemplo a seguir.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendo>
   Active: active (running) since Tue 2022-04-19 15:48:54 UTC; 19s ago
            --truncated--
```

Em casos raros, o comando informa que o agente está instalado, mas não está em execução, conforme mostrado no exemplo a seguir.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; disabled; vend>
   Active: inactive (dead)
            --truncated--
```

Para ativar o agente nesses casos, execute os comandos a seguir.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Criar comandos de instalação do agente personalizados para o CentOS 8 em sua região

Para instalar o SSM Agent em várias instâncias usando um script ou modelo, recomendamos usar arquivos de instalação armazenados no Região da AWS em que você está trabalhando.

Para os comandos a seguir, fornecemos exemplos que usam um bucket do S3 acessível ao público na região Leste dos EUA (Ohio) (`us-east-2`).

> **Tip**
>
> Também é possível substituir um URL global no procedimento Comandos de instalação rápida do SSM Agent no Cent OS 8 descrito anteriormente neste tópico por um URL regional personalizado construído por você.

No comando a seguir, substitua *`region`* por suas próprias informações. Para ver uma lista dos valores de *`região`* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

x86_64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_amd64/amazon-ssm-agent.rpm
```

ARM64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Instalar o SSM Agent no CentOS 7.x

As Amazon Machine Images (AMIs) para CentOS 7 fornecidas pela AWS não têm o AWS Systems Manager Agent (SSM Agent) pré-instalado por padrão. Use as informações nesta página para obter a ajuda para instalar ou reinstalar o agente em uma instância do CentOS 7.

Tópicos

- Comandos de instalação rápida do SSM Agent no Cent OS 7
- Criar comandos de instalação do agente personalizados para o CentOS 7 em sua região

Comandos de instalação rápida do SSM Agent no Cent OS 7

Use as etapas a seguir para instalar o SSM Agent manualmente em uma única instância. Este procedimento usa arquivos de instalação disponíveis globalmente.

Como instalar o SSM Agent no CentOS 7.x

1. Conecte à sua instância do CentOS 7 usando o método preferido, como SSH.
2. Copie o comando correspondente à arquitetura da instância e execute-o na instância.

> **Note**
>
> Mesmo que os URLs nos comandos a seguir incluam um diretório `ec2-downloads-windows`, estes são os arquivos de instalação global corretos para o CentOS 7.

Instâncias x86_64

```
sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_amd64/amazon-ssm-agent.rpm
```

Instâncias ARM64

```
sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_arm64/amazon-ssm-agent.rpm
```

3. (Recomendado) Execute o comando a seguir para verificar se o agente está em execução.

```
sudo systemctl status amazon-ssm-agent
```

Na maioria dos casos, o comando informa que o agente está em execução, conforme mostrado no exemplo a seguir.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendor
 preset: disabled)
   Active: active (running) since Tue 2022-04-19 15:57:27 UTC; 6s ago
           --truncated--
```

Em casos raros, o comando informa que o agente está instalado, mas não está em execução, conforme mostrado no exemplo a seguir.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendor
 preset: disabled)
   Active: inactive (dead) since Tue 2022-04-19 15:58:44 UTC; 2s ago
           --truncated--
```

Para ativar o agente nesses casos, execute os comandos a seguir.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Criar comandos de instalação do agente personalizados para o CentOS 7 em sua região

Para instalar o SSM Agent em várias instâncias usando um script ou modelo, recomendamos usar arquivos de instalação armazenados no Região da AWS em que você está trabalhando.

Para os comandos a seguir, fornecemos exemplos que usam um bucket do S3 acessível ao público na região Leste dos EUA (Ohio) (`us-east-2`).

> **Tip**
>
> Também é possível substituir um URL global no procedimento Comandos de instalação rápida do SSM Agent no Cent OS 7 descrito anteriormente neste tópico por um URL regional personalizado construído por você.

No comando a seguir, substitua *`region`* por suas próprias informações. Para ver uma lista dos valores de *`região`* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

x86_64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_amd64/amazon-ssm-agent.rpm
```

ARM64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Instalar o SSM Agent no CentOS 6.x

As Amazon Machine Images (AMIs) para CentOS 6 fornecidas pela AWS não têm o AWS Systems Manager Agent (SSM Agent) pré-instalado por padrão. Use as informações nesta página para obter a ajuda para instalar ou reinstalar o agente em uma instância do CentOS 6.

Tópicos

- Comandos de instalação rápida do SSM Agent no Cent OS 6
- Criar comandos de instalação do agente personalizados para o CentOS 6 em sua região

Comandos de instalação rápida do SSM Agent no Cent OS 6

Use as etapas a seguir para instalar o SSM Agent manualmente em uma única instância. Este procedimento usa arquivos de instalação disponíveis globalmente.

Como instalar o SSM Agent no CentOS 6.x

1. Conecte à sua instância do CentOS 6 usando o método preferido, como SSH.
2. Copie o comando correspondente à arquitetura da instância e execute-o na instância.

> **Note**
>
> Mesmo que os URLs nos comandos a seguir incluam um diretório `ec2-downloads-windows`, estes são os arquivos de instalação global corretos para o CentOS 6.
> Os comandos a seguir especificam o diretório da versão `3.0.1479.0` em vez de um diretório `latest`. Isso ocorre porque o SSM Agent versão 3.1 e posteriores não é compatível com o CentOS 6.

Instâncias x86_64

```
sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/
SSMAgent/3.0.1479.0/linux_amd64/amazon-ssm-agent.rpm
```

Instâncias x86

```
sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/
SSMAgent/3.0.1479.0/linux_386/amazon-ssm-agent.rpm
```

3. (Recomendado) Execute o comando a seguir para verificar se o agente está em execução.

```
sudo status amazon-ssm-agent
```

Na maioria dos casos, o comando informa que o agente está em execução, conforme mostrado no exemplo a seguir.

```
amazon-ssm-agent start/running, process 1744
```

Em casos raros, o comando informa que o agente está instalado, mas não está em execução, conforme mostrado no exemplo a seguir.

```
amazon-ssm-agent stop/waiting
```

Para ativar o agente nesses casos, execute o comando a seguir.

```
sudo start amazon-ssm-agent
```

Criar comandos de instalação do agente personalizados para o CentOS 6 em sua região

Para instalar o SSM Agent em várias instâncias usando um script ou modelo, recomendamos usar arquivos de instalação armazenados no Região da AWS em que você está trabalhando.

Para os comandos a seguir, fornecemos exemplos que usam um bucket do S3 acessível ao público na região Leste dos EUA (Ohio) (`us-east-2`).

> **Tip**
>
> Também é possível substituir um URL global no procedimento Comandos de instalação rápida do SSM Agent no Cent OS 6 descrito anteriormente neste tópico por um URL regional personalizado construído por você.

No comando a seguir, substitua *region* por suas próprias informações. Para ver uma lista dos valores de *região* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

> **Note**
>
> Os comandos a seguir especificam o diretório da versão `3.0.1479.0` em vez de um diretório `latest`. Isso ocorre porque o SSM Agent versão 3.1 e posteriores não é compatível com o CentOS 6.

x86_64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/3.0.1479.0/
linux_amd64/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-
east-2/3.0.1479.0/linux_amd64/amazon-ssm-agent.rpm
```

x86

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/3.0.1479.0/
linux_386/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-
east-2/3.0.1479.0/linux_386/amazon-ssm-agent.rpm
```

Instalar manualmente o SSM Agent em instâncias do CentOS Stream

As Amazon Machine Images (AMIs) para CentOS Stream fornecidas pela AWS não têm o AWS Systems Manager Agent (SSM Agent) pré-instalado por padrão. Para ver uma lista de AMIs gerenciadas pela AWS nas quais o agente pode estar pré-instalado, consulte Encontrar AMIs com o SSM Agent pré-instalado.

Use as informações nesta seção para obter a ajuda para instalar ou reinstalar o SSM Agent manualmente em uma instância do CentOS Stream.

Antes de começar

Antes de instalar o SSM Agent em uma instância do CentOS Stream, observe o seguinte:

- Para obter informações importantes aplicáveis à instalação do SSM Agent em todos os sistemas operacionais baseados em Linux, consulte Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux.

Tópicos

- Comandos de instalação rápida do SSM Agent no CentOS Stream
- Criar comandos de instalação do agente personalizados para o CentOS Stream em sua região

Comandos de instalação rápida do SSM Agent no CentOS Stream

Use as etapas a seguir para instalar o SSM Agent manualmente em uma única instância. Este procedimento usa arquivos de instalação disponíveis globalmente.

Antes de começar

Antes de instalar o SSM Agent em uma instância do CentOS Stream, observe o seguinte:

- Verifique se o Python 2 ou o Python 3 está instalado na instância do CentOS Stream 8. Isso é necessário para que o SSM Agent funcione corretamente.

Para instalar o SSM Agent no CentOS Stream

1. Conecte à sua instância do CentOS Stream usando o método preferido, como SSH.
2. Copie o comando correspondente à arquitetura da instância e execute-o na instância.

> **Note**
>
> Mesmo que os URLs nos comandos a seguir incluam um diretório `ec2-downloads-windows`, estes são os arquivos de instalação global corretos para o CentOS Stream.

Instâncias x86_64

```
sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_amd64/amazon-ssm-agent.rpm
```

Instâncias ARM64

```
sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_arm64/amazon-ssm-agent.rpm
```

3. (Recomendado) Execute o comando a seguir para verificar se o agente está em execução.

```
sudo systemctl status amazon-ssm-agent
```

Na maioria dos casos, o comando informa que o agente está em execução, conforme mostrado no exemplo a seguir.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendo>
   Active: active (running) since Tue 2022-04-19 16:40:41 UTC; 9s ago
 Main PID: 4898 (amazon-ssm-agen)
    Tasks: 14 (limit: 4821)
   Memory: 34.6M
   CGroup: /system.slice/amazon-ssm-agent.service
           ##4898 /usr/bin/amazon-ssm-agent
           ##4954 /usr/bin/ssm-agent-worker
            --truncated--
```

Em casos raros, o comando informa que o agente está instalado, mas não está em execução, conforme mostrado no exemplo a seguir.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendo>
```

```
   Active: inactive (dead) since Tue 2022-04-19 16:42:05 UTC; 2s ago
           --truncated--
```

Para ativar o agente nesses casos, execute os comandos a seguir.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Criar comandos de instalação do agente personalizados para o CentOS Stream em sua região

Para instalar o SSM Agent em várias instâncias usando um script ou modelo, recomendamos usar arquivos de instalação armazenados no Região da AWS em que você está trabalhando.

Para os comandos a seguir, fornecemos exemplos que usam um bucket do S3 acessível ao público na região Leste dos EUA (Ohio) (`us-east-2`).

> **Tip**
>
> Também é possível substituir um URL global no procedimento Comandos de instalação rápida do SSM Agent no CentOS Stream descrito anteriormente neste tópico por um URL regional personalizado construído por você.

No comando a seguir, substitua *`region`* por suas próprias informações. Para ver uma lista dos valores de *`região`* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

x86_64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_amd64/amazon-ssm-agent.rpm
```

ARM64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Instalar o SSM Agent manualmente em instâncias do Debian Server

As Amazon Machine Images (AMIs) para Debian Server fornecidas pela AWS não têm o AWS Systems Manager Agent (SSM Agent) pré-instalado por padrão. Para ver uma lista de AMIs gerenciadas pela AWS nas quais o agente pode estar pré-instalado, consulte Encontrar AMIs com o SSM Agent pré-instalado.

Use as informações nesta seção para obter a ajuda para instalar ou reinstalar o SSM Agent manualmente em uma instância do Debian Server.

Antes de começar

Antes de instalar o SSM Agent em uma instância do Debian Server, observe o seguinte:

- Para obter informações importantes aplicáveis à instalação do SSM Agent em todos os sistemas operacionais baseados em Linux, consulte Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux.

Tópicos

- Comandos de instalação rápida do SSM Agent no Debian Server
- Criar comandos de instalação do agente personalizados para o Debian Server em sua região

Comandos de instalação rápida do SSM Agent no Debian Server

Use as etapas a seguir para instalar o SSM Agent manualmente em uma única instância. Este procedimento usa arquivos de instalação disponíveis globalmente.

Para instalar o SSM Agent no Debian Server

1. Conecte à sua instância do Debian Server usando o método preferido, como SSH.
2. Insira o comando a seguir para criar um diretório temporário na instância.

```
mkdir /tmp/ssm
```

3. Execute o comando a seguir para mudar para o diretório temporário.

```
cd /tmp/ssm
```

4. Copie o comando correspondente à arquitetura da instância e execute-o na instância.

> **Note**
>
> Mesmo que os URLs nos comandos a seguir incluam um diretório `ec2-downloads-windows`, estes são os arquivos de instalação global corretos para o Debian Server. Somente a arquitetura x86_64 é compatível com o Debian Server 8.

Instâncias x86_64

```
wget https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/
debian_amd64/amazon-ssm-agent.deb
```

Instâncias ARM64

```
wget https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/
debian_arm64/amazon-ssm-agent.deb
```

5. Execute o seguinte comando:

```
sudo dpkg -i amazon-ssm-agent.deb
```

6. (Recomendado) Execute o comando a seguir para verificar se o agente está em execução.

```
sudo systemctl status amazon-ssm-agent
```

Na maioria dos casos, o comando informa que o agente está em execução, conforme mostrado no exemplo a seguir.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/lib/systemd/system/amazon-ssm-agent.service; enabled; vendor
   Active: active (running) since Tue 2022-04-19 16:25:03 UTC; 4s ago
 Main PID: 628 (amazon-ssm-agen)
   CGroup: /system.slice/amazon-ssm-agent.service
           ##628 /usr/bin/amazon-ssm-agent
           ##650 /usr/bin/ssm-agent-worker
            --truncated--
```

Em casos raros, o comando informa que o agente está instalado, mas não está em execução, conforme mostrado no exemplo a seguir.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/lib/systemd/system/amazon-ssm-agent.service; enabled; vendor
   Active: inactive (dead) since Tue 2022-04-19 16:26:30 UTC; 5s ago
 Main PID: 628 (code=exited, status=0/SUCCESS)
            --truncated--
```

Para ativar o agente nesses casos, execute os comandos a seguir.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Criar comandos de instalação do agente personalizados para o Debian Server em sua região

Para instalar o SSM Agent em várias instâncias usando um script ou modelo, recomendamos usar arquivos de instalação armazenados no Região da AWS em que você está trabalhando.

Para os comandos a seguir, fornecemos exemplos que usam um bucket do S3 acessível ao público na região Leste dos EUA (Ohio) (`us-east-2`).

> **Tip**
>
> Também é possível substituir um URL global no procedimento Comandos de instalação rápida do SSM Agent no Debian Server descrito anteriormente neste tópico por um URL regional personalizado construído por você.

No comando a seguir, substitua *region* por suas próprias informações. Para ver uma lista dos valores de *região* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

> **Note**
>
> Somente a arquitetura x86_64 é compatível com o Debian Server 8.

x86_64

```
wget https://s3.region.amazonaws.com/amazon-ssm-region/latest/debian_amd64/amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

Veja o exemplo a seguir.

```
wget https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/debian_amd64/amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

ARM64

```
wget https://s3.region.amazonaws.com/amazon-ssm-region/latest/debian_arm64/amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

Veja o exemplo a seguir.

```
wget https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/debian_arm64/
amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

Instalar o SSM Agent manualmente em instâncias do Oracle Linux

As Amazon Machine Images (AMIs) para Oracle Linux fornecidas pela AWS não têm o AWS Systems Manager Agent (SSM Agent) pré-instalado por padrão. Para ver uma lista de AMIs gerenciadas pela AWS nas quais o agente pode estar pré-instalado, consulte Encontrar AMIs com o SSM Agent pré-instalado.

Use as informações nesta seção para obter a ajuda para instalar ou reinstalar o SSM Agent manualmente em uma instância do Oracle Linux.

Antes de começar

Antes de instalar o SSM Agent em uma instância do Oracle Linux, observe o seguinte:

- Para obter informações importantes aplicáveis à instalação do SSM Agent em todos os sistemas operacionais baseados em Linux, consulte Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux.
- Se você usar um comando `yum` para atualizar o SSM Agent em um nó gerenciado após o agente ter sido instalado ou atualizado usando o documento `AWS-UpdateSSMAgent` do SSM, você poderá ver a seguinte mensagem: `"Warning: RPMDB altered outside of yum."` Esta mensagem é esperada e pode ser ignorada com segurança.

Tópicos

- Comandos de instalação rápida do SSM Agent no Oracle Linux
- Criar comandos de instalação do agente personalizados para o Oracle Linux em sua região

Comandos de instalação rápida do SSM Agent no Oracle Linux

Use as etapas a seguir para instalar o SSM Agent manualmente em uma única instância. Este procedimento usa arquivos de instalação disponíveis globalmente.

Para instalar o SSM Agent no Oracle Linux usando comandos rápidos de copiar e colar

1. Conecte à sua instância do Oracle Linux usando o método preferido, como SSH.
2. Copie o comando a seguir e execute-o na instância.

> **Note**
>
> Mesmo que o URL nos comandos a seguir inclua um diretório `ec2-downloads-windows`, estes são os arquivos de instalação global corretos para o Oracle Linux.

x86_64

```
sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_amd64/amazon-ssm-agent.rpm
```

3. (Recomendado) Execute o comando a seguir para verificar se o agente está em execução.

```
sudo systemctl status amazon-ssm-agent
```

Na maioria dos casos, o comando informa que o agente está em execução, conforme mostrado no exemplo a seguir.

```
amazon-ssm-agent.service - amazon-ssm-agent
Loaded: loaded (/usr/lib/systemd/system/amazon-ssm-agent.service; enabled; vendor
 preset: enabled)
Active: active (running) since Wed 2021-10-20 19:09:29 UTC; 4min 6s ago
            --truncated--
```

Em casos raros, o comando informa que o agente está instalado, mas não está em execução, conforme mostrado no exemplo a seguir.

```
amazon-ssm-agent.service - amazon-ssm-agent
Loaded: loaded (/usr/lib/systemd/system/amazon-ssm-agent.service; enabled; vendor
 preset: enabled)
Active: inactive (dead) since Wed 2021-10-20 22:16:41 UTC; 18s ago
            --truncated--
```

Para ativar o agente nesses casos, execute os comandos a seguir.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Criar comandos de instalação do agente personalizados para o Oracle Linux em sua região

Para instalar o SSM Agent em várias instâncias usando um script ou modelo, recomendamos usar arquivos de instalação armazenados no Região da AWS em que você está trabalhando.

Para os comandos a seguir, fornecemos exemplos que usam um bucket do S3 acessível ao público na região Leste dos EUA (Ohio) (`us-east-2`).

> **Tip**
>
> Também é possível substituir um URL global no procedimento Comandos de instalação rápida do SSM Agent no Oracle Linux descrito anteriormente neste tópico por um URL regional personalizado construído por você.

No comando a seguir, substitua *`region`* por suas próprias informações. Para ver uma lista dos valores de *`região`* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

x86_64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Instalar o SSM Agent manualmente em instâncias do Red Hat Enterprise Linux

As Amazon Machine Images (AMIs) para Red Hat Enterprise Linux (RHEL) fornecidas pela AWS não têm o AWS Systems Manager Agent (SSM Agent) pré-instalado por padrão. Para ver uma lista de

AMIs gerenciadas pela AWS nas quais o agente pode estar pré-instalado, consulte Encontrar AMIs com o SSM Agent pré-instalado.

Use as informações nesta seção para obter a ajuda para instalar ou reinstalar o SSM Agent manualmente em uma instância do RHEL.

Antes de começar

Antes de instalar o SSM Agent em uma instância do RHEL, observe o seguinte:

- Para obter informações importantes aplicáveis à instalação do SSM Agent em todos os sistemas operacionais baseados em Linux, consulte Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux.
- Se você usar um comando `yum` para atualizar o SSM Agent em um nó gerenciado após o agente ter sido instalado ou atualizado usando o documento `AWS-UpdateSSMAgent` do SSM, você poderá ver a seguinte mensagem: `"Warning: RPMDB altered outside of yum."` Esta mensagem é esperada e pode ser ignorada com segurança.

Tópicos

- Instalar o SSM Agent no RHEL 8.x, 9.x e 10.x
- Instalar o SSM Agent no RHEL 7.x
- Instalar o SSM Agent no RHEL 6.x

Instalar o SSM Agent no RHEL 8.x, 9.x e 10.x

As Amazon Machine Images (AMIs) para RHEL 8 e 9 fornecidas pela AWS não têm o AWS Systems Manager Agent (SSM Agent) pré-instalado por padrão. Use as informações nesta página para obter a ajuda para instalar ou reinstalar o agente em instâncias do RHEL 8 e 9.

Antes de começar

Antes de instalar o SSM Agent em uma instância do RHEL 8, 9 ou 10, observe o seguinte:

- Verifique se o Python 2 ou o Python 3 está instalado na instância do RHEL 8, 9 ou 10. Isso é necessário para que o SSM Agent funcione corretamente.

Tópicos

- Comandos de instalação rápida do SSM Agent no RHEL 8, 9 e 10

- Criar comandos de instalação do agente personalizados para o RHEL 8, 9 e 10 em sua região

Comandos de instalação rápida do SSM Agent no RHEL 8, 9 e 10

Use as etapas a seguir para instalar o SSM Agent manualmente em uma única instância. Este procedimento usa arquivos de instalação disponíveis globalmente.

Para instalar o SSM Agent no RHEL 8.x, 9.x ou 10.x

1. Conecte à instância do RHEL 8, 9 ou 10 usando o método de sua preferência, como SSH.
2. Copie o comando correspondente à arquitetura da instância e execute-o na instância.

> **Note**
>
> Mesmo que os URLs nos comandos a seguir incluam um diretório `ec2-downloads-windows`, estes são os arquivos de instalação global corretos para o RHEL 8, 9 e 10.

Instâncias x86_64

```
sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_amd64/amazon-ssm-agent.rpm
```

Instâncias ARM64

```
sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_arm64/amazon-ssm-agent.rpm
```

3. (Recomendado) Execute o comando a seguir para verificar se o agente está em execução.

```
sudo systemctl status amazon-ssm-agent
```

Na maioria dos casos, o comando informa que o agente está em execução, conforme mostrado no exemplo a seguir.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendo>
   Active: active (running) since Tue 2022-04-19 16:40:41 UTC; 9s ago
 Main PID: 4898 (amazon-ssm-agen)
```

```
    Tasks: 14 (limit: 4821)
   Memory: 34.6M
   CGroup: /system.slice/amazon-ssm-agent.service
           ##4898 /usr/bin/amazon-ssm-agent
           ##4954 /usr/bin/ssm-agent-worker
            --truncated--
```

Em casos raros, o comando informa que o agente está instalado, mas não está em execução, conforme mostrado no exemplo a seguir.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendo>
   Active: inactive (dead) since Tue 2022-04-19 16:42:05 UTC; 2s ago
            --truncated--
```

Para ativar o agente nesses casos, execute os comandos a seguir.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Criar comandos de instalação do agente personalizados para o RHEL 8, 9 e 10 em sua região

Para instalar o SSM Agent em várias instâncias usando um script ou modelo, recomendamos usar arquivos de instalação armazenados no Região da AWS em que você está trabalhando.

Para os comandos a seguir, fornecemos exemplos que usam um bucket do S3 acessível ao público na região Leste dos EUA (Ohio) (`us-east-2`).

> **Tip**
>
> Também é possível substituir um URL global no procedimento Comandos de instalação rápida do SSM Agent no RHEL 8, 9 e 10 descrito anteriormente neste tópico por um URL regional personalizado construído por você.

No comando a seguir, substitua *region* por suas próprias informações. Para ver uma lista dos valores de *região* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

x86_64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_amd64/amazon-ssm-agent.rpm
```

ARM64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Instalar o SSM Agent no RHEL 7.x

As Amazon Machine Images (AMIs) para RHEL 7 fornecidas pela AWS não têm o AWS Systems Manager Agent (SSM Agent) pré-instalado por padrão. Use as informações nesta página para obter a ajuda para instalar ou reinstalar o agente em instâncias do RHEL 7.

Tópicos

- Comandos de instalação rápida do SSM Agent no RHEL 7
- Criar comandos de instalação do agente personalizados para o RHEL 7 em sua região

Comandos de instalação rápida do SSM Agent no RHEL 7

Use as etapas a seguir para instalar o SSM Agent manualmente em uma única instância. Este procedimento usa arquivos de instalação disponíveis globalmente.

Como instalar o SSM Agent no RHEL 7.x

1. Conecte à sua instância do RHEL 7 usando o método preferido, como SSH.
2. Copie o comando correspondente à arquitetura da instância e execute-o na instância.

> **Note**
>
> Mesmo que os URLs nos comandos a seguir incluam um diretório `ec2-downloads-windows`, estes são os arquivos de instalação global corretos para o RHEL 7.

Instâncias x86_64

```
sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_amd64/amazon-ssm-agent.rpm
```

Instâncias ARM64

```
sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_arm64/amazon-ssm-agent.rpm
```

3. (Recomendado) Execute o comando a seguir para verificar se o agente está em execução.

```
sudo systemctl status amazon-ssm-agent
```

Na maioria dos casos, o comando informa que o agente está em execução, conforme mostrado no exemplo a seguir.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendor
 preset: disabled)
   Active: active (running) since Tue 2022-04-19 16:47:36 UTC; 22s ago
 Main PID: 1342 (amazon-ssm-agen)
   CGroup: /system.slice/amazon-ssm-agent.service
           ##1342 /usr/bin/amazon-ssm-agent
           ##1362 /usr/bin/ssm-agent-worker
            --truncated--
```

Em casos raros, o comando informa que o agente está instalado, mas não está em execução, conforme mostrado no exemplo a seguir.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendor
 preset: disabled)
   Active: inactive (dead) since Tue 2022-04-19 16:48:56 UTC; 5s ago
  Process: 1342 ExecStart=/usr/bin/amazon-ssm-agent (code=exited, status=0/SUCCESS)
 Main PID: 1342 (code=exited, status=0/SUCCESS)
            --truncated--
```

Para ativar o agente nesses casos, execute os comandos a seguir.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Criar comandos de instalação do agente personalizados para o RHEL 7 em sua região

Para instalar o SSM Agent em várias instâncias usando um script ou modelo, recomendamos usar arquivos de instalação armazenados no Região da AWS em que você está trabalhando.

Para os comandos a seguir, fornecemos exemplos que usam um bucket do S3 acessível ao público na região Leste dos EUA (Ohio) (`us-east-2`).

> **Tip**
>
> Também é possível substituir um URL global no procedimento Comandos de instalação rápida do SSM Agent no RHEL 7 descrito anteriormente neste tópico por um URL regional personalizado construído por você.

No comando a seguir, substitua *`region`* por suas próprias informações. Para ver uma lista dos valores de *`região`* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

x86_64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_amd64/amazon-ssm-agent.rpm
```

ARM64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Instalar o SSM Agent no RHEL 6.x

As Amazon Machine Images (AMIs) para RHEL 6 fornecidas pela AWS não têm o AWS Systems Manager Agent (SSM Agent) pré-instalado por padrão. Use as informações nesta página para obter a ajuda para instalar ou reinstalar o agente em instâncias do RHEL 6.

Tópicos

- Comandos de instalação rápida do SSM Agent no RHEL 6
- Criar comandos de instalação do agente personalizados para o RHEL 6 em sua região

Comandos de instalação rápida do SSM Agent no RHEL 6

Use as etapas a seguir para instalar o SSM Agent manualmente em uma única instância. Este procedimento usa arquivos de instalação disponíveis globalmente.

Como instalar o SSM Agent no RHEL 6.x

1. Conecte à sua instância do RHEL 6 usando o método preferido, como SSH.

2. Copie o comando correspondente à arquitetura da instância e execute-o na instância.

> **Note**
>
> Mesmo que os URLs nos comandos a seguir incluam um diretório `ec2-downloads-windows`, estes são os arquivos de instalação global corretos para o RHEL 6.
> Os comandos a seguir especificam o diretório da versão `3.0.1479.0` em vez de um diretório `latest`. Isso ocorre porque o SSM Agent versão 3.1 e posteriores não é compatível com o RHEL 6.

Instâncias x86_64

```
sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/
SSMAgent/3.0.1479.0/linux_amd64/amazon-ssm-agent.rpm
```

Instâncias x86

```
sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/
SSMAgent/3.0.1479.0/linux_386/amazon-ssm-agent.rpm
```

3. (Recomendado) Execute o comando a seguir para verificar se o agente está em execução.

```
sudo status amazon-ssm-agent
```

Na maioria dos casos, o comando informa que o agente está em execução, conforme mostrado no exemplo a seguir.

```
amazon-ssm-agent start/running, process 1788
```

Em casos raros, o comando informa que o agente está instalado, mas não está em execução, conforme mostrado no exemplo a seguir.

```
amazon-ssm-agent stop/waiting
```

Para ativar o agente nesses casos, execute o comando a seguir.

```
sudo start amazon-ssm-agent
```

Criar comandos de instalação do agente personalizados para o RHEL 6 em sua região

Para instalar o SSM Agent em várias instâncias usando um script ou modelo, recomendamos usar arquivos de instalação armazenados no Região da AWS em que você está trabalhando.

Para os comandos a seguir, fornecemos exemplos que usam um bucket do S3 acessível ao público na região Leste dos EUA (Ohio) (`us-east-2`).

> **Tip**
>
> Também é possível substituir um URL global no procedimento Comandos de instalação rápida do SSM Agent no RHEL 6 descrito anteriormente neste tópico por um URL regional personalizado construído por você.

No comando a seguir, substitua *`region`* por suas próprias informações. Para ver uma lista dos valores de *`região`* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

> **Note**
>
> Os comandos a seguir especificam o diretório da versão `3.0.1479.0` em vez de um diretório `latest`. Isso ocorre porque o SSM Agent versão 3.1 e posteriores não é compatível com o RHEL 6.

x86_64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/3.0.1479.0/
linux_amd64/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-
east-2/3.0.1479.0/linux_amd64/amazon-ssm-agent.rpm
```

x86

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/3.0.1479.0/
linux_386/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-
east-2/3.0.1479.0/linux_386/amazon-ssm-agent.rpm
```

Instalar manualmente o SSM Agent em instâncias do Rocky Linux

As Amazon Machine Images (AMIs) para Rocky Linux fornecidas pela AWS não têm o AWS Systems Manager Agent (SSM Agent) pré-instalado por padrão. Para ver uma lista de AMIs gerenciadas pela AWS nas quais o agente pode estar pré-instalado, consulte Encontrar AMIs com o SSM Agent pré-instalado.

Use as informações nesta seção para obter a ajuda para instalar ou reinstalar o SSM Agent manualmente em uma instância do Rocky Linux.

Antes de começar

Antes de instalar o SSM Agent em uma instância do Rocky Linux, observe o seguinte:

- Para obter informações importantes aplicáveis à instalação do SSM Agent em todos os sistemas operacionais baseados em Linux, consulte Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux.

Tópicos

- Comandos de instalação rápida do SSM Agent no Rocky Linux
- Criar comandos de instalação do agente personalizados para o Rocky Linux em sua região

Comandos de instalação rápida do SSM Agent no Rocky Linux

Use as etapas a seguir para instalar o SSM Agent manualmente em uma única instância. Este procedimento usa arquivos de instalação disponíveis globalmente.

Antes de começar

Antes de instalar o SSM Agent em uma instância do Rocky Linux, observe o seguinte:

- Verifique se o Python 2 ou o Python 3 está instalado na instância do Rocky Linux. Isso é necessário para que o SSM Agent funcione corretamente.

Para instalar o SSM Agent no Rocky Linux

1. Conecte à sua instância do Rocky Linux usando o método preferido, como SSH.
2. Copie o comando correspondente à arquitetura da instância e execute-o na instância.

> **Note**
>
> Mesmo que os URLs nos comandos a seguir incluam um diretório `ec2-downloads-windows`, estes são os arquivos de instalação global corretos para o Rocky Linux.

Instâncias x86_64

```
sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_amd64/amazon-ssm-agent.rpm
```

Instâncias ARM64

```
sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_arm64/amazon-ssm-agent.rpm
```

3. (Recomendado) Execute o comando a seguir para verificar se o agente está em execução.

```
sudo systemctl status amazon-ssm-agent
```

Na maioria dos casos, o comando informa que o agente está em execução, conforme mostrado no exemplo a seguir.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendo>
   Active: active (running) since Tue 2022-04-19 16:40:41 UTC; 9s ago
 Main PID: 4898 (amazon-ssm-agen)
    Tasks: 14 (limit: 4821)
   Memory: 34.6M
```

```
   CGroup: /system.slice/amazon-ssm-agent.service
           ##4898 /usr/bin/amazon-ssm-agent
           ##4954 /usr/bin/ssm-agent-worker
            --truncated--
```

Em casos raros, o comando informa que o agente está instalado, mas não está em execução, conforme mostrado no exemplo a seguir.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendo>
   Active: inactive (dead) since Tue 2022-04-19 16:42:05 UTC; 2s ago
            --truncated--
```

Para ativar o agente nesses casos, execute os comandos a seguir.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Criar comandos de instalação do agente personalizados para o Rocky Linux em sua região

Para instalar o SSM Agent em várias instâncias usando um script ou modelo, recomendamos usar arquivos de instalação armazenados no Região da AWS em que você está trabalhando.

Para os comandos a seguir, fornecemos exemplos que usam um bucket do S3 acessível ao público na região Leste dos EUA (Ohio) (`us-east-2`).

> **Tip**
>
> Também é possível substituir um URL global no procedimento Comandos de instalação rápida do SSM Agent no Rocky Linux descrito anteriormente neste tópico por um URL regional personalizado construído por você.

No comando a seguir, substitua *`region`* por suas próprias informações. Para ver uma lista dos valores de *`região`* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

x86_64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_amd64/amazon-ssm-agent.rpm
```

ARM64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Instalar manualmente o SSM Agent em instâncias do SUSE Linux Enterprise Server

Na maioria dos casos, as Amazon Machine Images (AMIs) para SUSE Linux Enterprise Server (SLES) fornecidas pela AWS têm o AWS Systems Manager Agent (SSM Agent) pré-instalado por padrão. Para obter mais informações, consulte Encontrar AMIs com o SSM Agent pré-instalado.

Se o SSM Agent não estiver pré-instalado em uma nova instância do SLES, ou se for necessário reinstalar o agente manualmente, use as informações nesta página para obter ajuda.

Antes de começar

Antes de instalar o SSM Agent em uma instância do SLES, observe o seguinte:

- Para obter informações importantes aplicáveis à instalação do SSM Agent em todos os sistemas operacionais baseados em Linux, consulte Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux.

Tópicos

- Comandos de instalação rápida do SSM Agent no SLES
- Criar comandos de instalação do agente personalizados para o SLES em sua região

Comandos de instalação rápida do SSM Agent no SLES

Use as etapas a seguir para instalar o SSM Agent manualmente em uma única instância. Este procedimento usa arquivos de instalação disponíveis globalmente.

Para instalar o SSM Agent no SLES usando comandos rápidos de copiar e colar

1. Conecte à sua instância do SLES usando o método preferido, como SSH.
2. Opção 1: use um comando `zypper`:

   - Execute o seguinte comando:

     ```
     sudo zypper install amazon-ssm-agent
     ```

   - Insira y em resposta a qualquer prompt.

   Opção 2: use um comando `rpm`.

   - Crie um diretório temporário na instância.

     ```
     mkdir /tmp/ssm
     ```

   - Altere para o diretório temporário.

     ```
     cd /tmp/ssm
     ```

   - Execute os seguintes comandos um de cada vez para fazer download do instalador do SSM Agent e executá-lo.

     > **Note**
     >
     > Mesmo que os URLs nos comandos a seguir incluam um diretório `ec2-downloads-windows`, estes são os arquivos de instalação global corretos para o SLES.

     Instâncias do x86_64:

```
wget https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux_amd64/
amazon-ssm-agent.rpm
```

Instâncias do ARM64:

```
wget https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux_arm64/
amazon-ssm-agent.rpm
```

- Execute o seguinte comando:

```
sudo rpm --install amazon-ssm-agent.rpm
```

- (Recomendado) Execute o comando a seguir para verificar se o agente está em execução.

```
sudo systemctl status amazon-ssm-agent
```

Na maioria dos casos, o comando informa que o agente está em execução, conforme mostrado no exemplo a seguir.

```
# amazon-ssm-agent.service - amazon-ssm-agent
 Loaded: loaded (/usr/lib/systemd/system/amazon-ssm-agent.service; enabled;
 vendor preset: disabled)
 Active: active (running) since Mon 2022-02-21 23:13:28 UTC; 7s ago
 Main PID: 2102 (amazon-ssm-agen)
 Tasks: 15 (limit: 512)
 CGroup: /system.slice/amazon-ssm-agent.service
 ##2102 /usr/sbin/amazon-ssm-agent
 ##2107 /usr/sbin/ssm-agent-worker
           --truncated--
```

Em casos raros, o comando informa que o agente está instalado, mas não está em execução, conforme mostrado no exemplo a seguir.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/usr/lib/systemd/system/amazon-ssm-agent.service; disabled;
 vendor preset: disabled)
   Active: inactive (dead)
           --truncated--
```

Para ativar o agente nesses casos, execute os comandos a seguir.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Criar comandos de instalação do agente personalizados para o SLES em sua região

Para instalar o SSM Agent em várias instâncias usando um script ou modelo, recomendamos usar arquivos de instalação armazenados no Região da AWS em que você está trabalhando.

Para os comandos a seguir, fornecemos exemplos que usam um bucket do S3 acessível ao público na região Leste dos EUA (Ohio) (`us-east-2`).

> **Tip**
>
> Também é possível substituir um URL global no procedimento Comandos de instalação rápida do SSM Agent no Amazon Linux 1 descrito anteriormente neste tópico por um URL regional personalizado construído por você.

No comando a seguir, substitua *`region`* por suas próprias informações. Para ver uma lista dos valores de *`região`* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

x86_64

```
wget https://s3.region.amazonaws.com/amazon-ssm-region/latest/linux_amd64/amazon-
ssm-agent.rpm
```

```
sudo rpm --install amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
wget https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/linux_amd64/
amazon-ssm-agent.rpm
```

```
sudo rpm --install amazon-ssm-agent.rpm
```

ARM64

```
wget https://s3.region.amazonaws.com/amazon-ssm-region/latest/linux_arm64/amazon-ssm-agent.rpm
```

```
sudo rpm --install amazon-ssm-agent.rpm
```

Veja o exemplo a seguir.

```
wget https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/linux_arm64/amazon-ssm-agent.rpm
```

```
sudo rpm --install amazon-ssm-agent.rpm
```

Instalar o SSM Agent manualmente em instâncias do Ubuntu Server

> **Important**
>
> Antes de instalar o SSM Agent em uma versão de 64 bits do Ubuntu Server, certifique-se de estar usando as ferramentas de instalação corretas. Começando com imagens de máquina da Amazon (AMIs) que são identificadas com 20180627, o SSM Agent é pré-instalado na versão 16.04 usando pacotes Snap. Em instâncias criadas de AMIs anteriores, o SSM Agent deve ser instalado usando pacotes do instalador deb. Para ter mais informações, consulte Determinar a versão do SSM Agent correta a ser instalada nas instâncias do Ubuntu Server 16.04 de 64 bits.

Na maioria dos casos, as Amazon Machine Images (AMIs) para Ubuntu Server fornecidas pela AWS têm o AWS Systems Manager Agent (SSM Agent) pré-instalado por padrão. Para ter mais informações, consulte Encontrar AMIs com o SSM Agent pré-instalado.

Se o SSM Agent não estiver pré-instalado em uma nova instância do Ubuntu Server, ou se for necessário reinstalar o agente manualmente, use as informações nesta seção para obter ajuda.

Antes de começar

Antes de instalar o SSM Agent em uma instância do Ubuntu Server, observe o seguinte:

- Para obter informações importantes aplicáveis à instalação do SSM Agent em todos os sistemas operacionais baseados em Linux, consulte Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux.

Tópicos

- Instalar o SSM Agent no Ubuntu Server 16.04 LTS de 64 bits (Snap), 18.04, 20.04, 20.10 STR, 22.04 LTS, 23.10, 24.04 LTS, 24.0 e 25.04
- Instalar o SSM Agent no Ubuntu Server 16.04 e 14.04 de 64 bits (deb)
- Instalar o SSM Agent no Ubuntu Server 16.04 e 14.04 de 32 bits
- Determinar a versão do SSM Agent correta a ser instalada nas instâncias do Ubuntu Server 16.04 de 64 bits

Instalar o SSM Agent no Ubuntu Server 16.04 LTS de 64 bits (Snap), 18.04, 20.04, 20.10 STR, 22.04 LTS, 23.10, 24.04 LTS, 24.0 e 25.04

Antes de começar

Antes de instalar o SSM Agent em um Ubuntu Server 16.04 LTS de 64 bits (Snap), 18.04, 20.04, 20.10 STR, 22.04 LTS, 23.10, 24.04 LTS, 24.0 e 25.04, observe o seguinte:

Instalação da versão 16.04 por instaladores Snaps ou deb

No Ubuntu Server 16.04, o SSM Agent é instalado usando os pacotes de instalação Snaps ou deb, dependendo da versão da AMI 16.04.

Locais de arquivos do instalador do SSM Agent

No Ubuntu Server 16,04 LTS de 64 bits (Snap), 18,04, 20,04, 20,10 STR, 22,04 LTS, 23,10, 24,04 LTS, 24,0 e 25,04, os arquivos do instalador do SSM Agent, incluindo arquivos binários do agente e arquivos de configuração, são armazenados no seguinte diretório: `/snap/amazon-ssm-agent/current/`. Se você fizer alterações em qualquer arquivo de configuração nesse diretório, copie esses arquivos do diretório `/snap` para o `/etc/amazon/ssm/`. Os arquivos de log e biblioteca não foram alterados (`/var/lib/amazon/ssm`, `/var/log/amazon/ssm`).

Usar o canal `candidate` do Snap

O canal candidato na loja Snap contém a versão mais recente do SSM Agent (incluindo todas as correções de bugs mais recentes), não o canal estável. Para saber mais sobre as diferenças entre os canais candidatos e estáveis, consulte Níveis de risco em https://snapcraft.io/docs/channels.

Se você quiser acompanhar as informações de versão do SSM Agent no canal candidato, execute o comando a seguir nas instâncias de 64 bits do Ubuntu Server 20.10 STR e 20.04, 18.04 e 16.04 LTS.

```
sudo snap switch --channel=candidate amazon-ssm-agent
```

Snaps recomendados nas versões 18.04 e posteriores

No Ubuntu Server 18.04, 20.04, 20.10 STR, 22.04 LTS, 23.10, 24.04 LTS, 24.0 w 25.04, recomendamos usar apenas Snaps. Além disso, verifique se apenas uma instância do agente está instalada e em execução nas suas instâncias. Se você quiser usar o SSM Agent sem Snaps, desinstale o SSM Agent. Então instale o SSM Agent como um pacote debian usando as instruções para instalar o SSM Agent no Ubuntu Server 16.04 e 14.04 de 64 bits (deb). Antes de instalar, certifique-se de não haja nenhum Snap instalado que se sobreponha à lista de pacotes que você deseja gerenciar como pacotes debian.

Mensagem de erro do `Maximum timeout exceeded`

Devido a um problema conhecido com o Snap, você pode ver um erro `Maximum timeout exceeded` com os comandos `snap`. Se este erro for exibido, execute os seguintes comandos, um de cada vez, para iniciar o agente, pará-lo e verificar seu status:

```
sudo systemctl start snap.amazon-ssm-agent.amazon-ssm-agent.service
```

```
sudo systemctl stop snap.amazon-ssm-agent.amazon-ssm-agent.service
```

```
sudo systemctl status snap.amazon-ssm-agent.amazon-ssm-agent.service
```

Para instalar o SSM Agent no Ubuntu Server 16.04 LTS de 64 bits (Snap), 18.04, 20.04, 20.10 STR, 22.04 LTS, 23.10, 24.04 LTS, 24.0 e 25.04 (com pacote Snap)

1. O SSM Agent é instalado por padrão no Ubuntu Server 16.04 LTS de 64 bits (Snap), 18.04, 20.04, 20.10 STR, 22.04 LTS, 23.10, 24.04 LTS, 24.0 e 25.04 AMIs com um identificador de `20180627` ou posterior.

   Você pode usar o seguinte script se precisar instalar o SSM Agent em um servidor on-premises ou se precisar reinstalar o agente. Você não precisa especificar um URL para download porque o comando `snap` faz download automático do agente da loja de aplicativos Snap em https://snapcraft.io.

   ```
   sudo snap install amazon-ssm-agent --classic
   ```

2. Execute o comando a seguir para determinar se o SSM Agent está em execução.

   ```
   sudo snap list amazon-ssm-agent
   ```

3. Execute o comando a seguir para iniciar o serviço se o comando anterior retornar `amazon-ssm-agent is stopped`, `inactive` ou `disabled`.

   ```
   sudo snap start amazon-ssm-agent
   ```

4. Verifique o status do agente.

   ```
   sudo snap services amazon-ssm-agent
   ```

Instalar o SSM Agent no Ubuntu Server 16.04 e 14.04 de 64 bits (deb)

> **Important**
>
> Antes de instalar o SSM Agent em uma versão de 64 bits do Ubuntu Server, certifique-se de estar usando as ferramentas de instalação corretas. Começando com imagens de máquina da Amazon (AMIs) que são identificadas com 20180627, o SSM Agent é pré-instalado na versão 16.04 usando pacotes Snap. Em instâncias criadas de AMIs anteriores, o SSM Agent deve ser instalado usando pacotes do instalador deb. Para obter mais informações, consulte Determinar a versão do SSM Agent correta a ser instalada nas instâncias do Ubuntu Server 16.04 de 64 bits. Se o SSM Agent estiver instalado na instância juntamente com um Snap e

você instalar ou atualizar o SSM Agent usando um pacote de instalação deb, a instalação ou operações do SSM Agent poderão falhar.

Na maioria dos casos, as Amazon Machine Images (AMIs) para Ubuntu Server 16.04 fornecidas pela AWS têm o AWS Systems Manager Agent (SSM Agent) pré-instalado por padrão. Para obter mais informações, consulte Encontrar AMIs com o SSM Agent pré-instalado.

Se o SSM Agent não estiver pré-instalado em uma nova instância do Ubuntu Server 16.04 anterior à versão 20180627, se você estiver instalando no Ubuntu Server 14.04 ou se for necessário reinstalar o agente manualmente, use as informações nesta página para obter ajuda.

Comandos de instalação rápida para SSM Agent no Ubuntu Server 16.04 e 14.04 de 64 bits (deb)

Use as etapas a seguir para instalar o SSM Agent manualmente em uma única instância. Este procedimento usa arquivos de instalação disponíveis globalmente.

Para instalar o SSM Agent no Ubuntu Server 16.04 e 14.04 de 64 bits (deb) usando comandos rápidos de copiar e colar

1. Conecte à sua instância do Ubuntu Server usando o método preferido, como SSH.
2. Insira o comando a seguir para criar um diretório temporário na instância.

```
mkdir /tmp/ssm
```

3. Altere para o diretório temporário.

```
cd /tmp/ssm
```

4. Execute os seguintes comandos.

> **Note**
>
> Mesmo que os URLs nos comandos a seguir incluam um diretório `ec2-downloads-windows`, estes são os arquivos de instalação global corretos para o Ubuntu Server 16.04 e 14.04 de 64 bits.

```
wget https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/debian_amd64/
amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

5. (Recomendado) Execute um dos comandos a seguir para determinar se o SSM Agent está em execução.

   Ubuntu Server 16.04

   ```
   sudo systemctl status amazon-ssm-agent
   ```

   Ubuntu Server 14.04

   ```
   sudo status amazon-ssm-agent
   ```

   Na maioria dos casos, o comando informa que o agente está em execução.

   Em casos raros, o comando informa que o agente está instalado, mas não está em execução, conforme mostrado no exemplo a seguir.

6. Execute um dos comandos a seguir para iniciar o serviço se o comando anterior retornar `amazon-ssm-agent is stopped`, `inactive` ou `disabled`.

   Ubuntu Server 16.04:

   ```
   sudo systemctl enable amazon-ssm-agent
   ```

   Ubuntu Server 14.04:

   ```
   sudo start amazon-ssm-agent
   ```

Criar comandos de instalação personalizados para o SSM Agent no Ubuntu Server 16.04 e 14.04 de 64 bits (deb) em sua região

Para instalar o SSM Agent em várias instâncias usando um script ou modelo, recomendamos usar arquivos de instalação armazenados no Região da AWS em que você está trabalhando.

Para os comandos a seguir, fornecemos exemplos que usam um bucket do S3 acessível ao público na região Leste dos EUA (Ohio) (`us-east-2`).

> **Tip**
>
> Também é possível substituir um URL global no procedimento Comandos de instalação rápida para SSM Agent no Ubuntu Server 16.04 e 14.04 de 64 bits (deb) descrito anteriormente neste tópico por um URL regional personalizado construído por você.

No comando a seguir, substitua *`region`* por suas próprias informações. Para ver uma lista dos valores de *`região`* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

```
wget https://s3.region.amazonaws.com/amazon-ssm-region/latest/debian_amd64/amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

Veja o exemplo a seguir.

```
wget https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/debian_amd64/amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

Instalar o SSM Agent no Ubuntu Server 16.04 e 14.04 de 32 bits

Na maioria dos casos, as Amazon Machine Images (AMIs) para Ubuntu Server 16.04 fornecidas pela AWS têm o AWS Systems Manager Agent (SSM Agent) pré-instalado por padrão. Para obter mais informações, consulte Encontrar AMIs com o SSM Agent pré-instalado.

Se o SSM Agent não estiver pré-instalado em uma nova instância do Ubuntu Server 16.04, se você estiver instalando no Ubuntu Server 14.04 ou se for necessário reinstalar o agente manualmente, use as informações nesta página para obter ajuda.

Comandos de instalação rápida para o SSM Agent no Ubuntu Server 16.04 e 14.04 de 32 bits (deb)

Use as etapas a seguir para instalar o SSM Agent manualmente em uma única instância. Este procedimento usa arquivos de instalação disponíveis globalmente.

Para instalar o SSM Agent no Ubuntu Server 16.04 e 14.04 de 32 bits (deb) usando comandos rápidos de copiar e colar

1. Conecte à sua instância do Ubuntu Server usando o método preferido, como SSH.
2. Insira o comando a seguir para criar um diretório temporário na instância.

```
mkdir /tmp/ssm
```

3. Altere para o diretório temporário.

```
cd /tmp/ssm
```

4. Execute os seguintes comandos.

> **Note**
>
> Mesmo que os URLs nos comandos a seguir incluam um diretório `ec2-downloads-windows`, estes são os arquivos de instalação global corretos para o Ubuntu Server 16.04 e 14.04 de 32 bits.

```
wget https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/debian_386/
amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

5. (Recomendado) Execute um dos comandos a seguir para determinar se o SSM Agent está em execução.

Ubuntu Server 16.04

```
sudo systemctl status amazon-ssm-agent
```

Ubuntu Server 14.04

```
sudo status amazon-ssm-agent
```

Na maioria dos casos, o comando informa que o agente está em execução.

Em casos raros, o comando informa que o agente está instalado, mas não está em execução, conforme mostrado no exemplo a seguir.

6. Execute um dos comandos a seguir para iniciar o serviço se o comando anterior retornar `amazon-ssm-agent is stopped`, `inactive` ou `disabled`.

   Ubuntu Server 16.04:

   ```
   sudo systemctl enable amazon-ssm-agent
   ```

   Ubuntu Server 14.04:

   ```
   sudo start amazon-ssm-agent
   ```

Criar comandos de instalação personalizados para o SSM Agent no Ubuntu Server 16.04 e 14.04 de 32 bits (deb) em sua região

Para instalar o SSM Agent em várias instâncias usando um script ou modelo, recomendamos usar arquivos de instalação armazenados no Região da AWS em que você está trabalhando.

Para os comandos a seguir, fornecemos exemplos que usam um bucket do S3 acessível ao público na região Leste dos EUA (Ohio) (`us-east-2`).

> **Tip**
>
> Também é possível substituir um URL global no procedimento Comandos de instalação rápida para o SSM Agent no Ubuntu Server 16.04 e 14.04 de 32 bits (deb) descrito anteriormente neste tópico por um URL regional personalizado construído por você.

No comando a seguir, substitua *region* por suas próprias informações. Para ver uma lista dos valores de *região* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

```
wget https://s3.region.amazonaws.com/amazon-ssm-region/latest/debian_386/amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

Veja o exemplo a seguir.

```
wget https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/debian_386/amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

Determinar a versão do SSM Agent correta a ser instalada nas instâncias do Ubuntu Server 16.04 de 64 bits

> **Important**
>
> Antes de instalar o SSM Agent em uma versão de 64 bits do Ubuntu Server, certifique-se de estar usando as ferramentas de instalação corretas. Começando com imagens de máquina da Amazon (AMIs) que são identificadas com 20180627, o SSM Agent é pré-instalado na versão 16.04 usando pacotes Snap. Em instâncias criadas de AMIs anteriores, o SSM Agent deve ser instalado usando pacotes do instalador deb. Para obter mais informações, consulte Determinar a versão do SSM Agent correta a ser instalada nas instâncias do Ubuntu Server 16.04 de 64 bits.

Lembre-se de que, se uma instância tem mais de uma instalação do SSM Agent (por exemplo, uma instalada usando um Snap, outra usando um instalador deb), as operações do agente não funcionarão corretamente.

Você pode verificar a data de criação de ID da AMI da fonte para uma instância usando um dos métodos a seguir. Esses procedimentos aplicam-se somente a AMIs gerenciadas pela AWS.

Verificar a data de criação de um ID de AMI de origem (console)

1. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. No painel de navegação à esquerda, escolha Instances (Instâncias).
3. Selecione uma instância.
4. Na guia Detalhes, verifique se há um identificador YYYYMMDD no valor existente no campo Nome da AMI. Por exemplo: `ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20180627`.

Verificar a data de criação de um ID de AMI de origem (AWS CLI)

- Execute o seguinte comando .

  ```
  aws ec2 describe-images --image-ids ami-id
  ```

  ami-id representa o ID de uma AMI fornecido pela AWS, por exemplo, `ami-07c8bc5c1ce9598c3`.

  Se houver êxito, o comando retornará informações da maneira a seguir e você poderá verificar as informações nos campos `CreationDate` e `Name`.

  ```
  {
      "Images": [
          {
              "Architecture": "x86_64",
              "CreationDate": "2020-07-24T20:40:27.000Z",
              "ImageId": "ami-07c8bc5c1ce9598c3",
  -- truncated --
              "ImageOwnerAlias": "amazon",
              "Name": "amzn2-ami-hvm-2.0.20200722.0-x86_64-gp2",
              "RootDeviceName": "/dev/xvda",
  ```

```
            "RootDeviceType": "ebs",
            "SriovNetSupport": "simple",
            "VirtualizationType": "hvm"
        }
    ]
}
```

Configurar o SSM Agent para usar um proxy em nós do Linux

É possível configurar o AWS Systems Manager para se comunicar por meio de um proxy HTTP adicionando as configurações SSM Agent, `http_proxy` e `https_proxy` a um arquivo de configuração `no_proxy`. Um arquivo de substituição também preserva as configurações de proxy se você instalar versões mais recentes ou mais antigas do SSM Agent. Esta seção inclui procedimentos para ambientes upstart e systemd. Se você pretende usar o Session Manager, observe que os servidores proxy HTTPS não são compatíveis.

Tópicos

- Configurar o SSM Agent para usar um proxy (upstart)
- Configurar o SSM Agent para usar um proxy (systemd)

Configurar o SSM Agent para usar um proxy (upstart)

Use o procedimento a seguir para criar um arquivo de configuração de substituição para um ambiente `upstart`.

Configurar o SSM Agent para usar um proxy (upstart)

1. Conecte-se à instância gerenciada na qual você instalou o SSM Agent.
2. Abra um editor simples como o VIM e, caso esteja usando um servidor de proxy HTTP ou um servidor de proxy HTTPS, especifique uma das opções de configuração a seguir.

   Para um servidor proxy HTTP:

   ```
   env http_proxy=http://hostname:port
   env https_proxy=http://hostname:port
   env no_proxy=IP address for instance metadata services (IMDS)
   ```

   Para um servidor proxy HTTP:

```
env http_proxy=http://hostname:port
env https_proxy=https://hostname:port
env no_proxy=IP address for instance metadata services (IMDS)
```

> **Important**
>
> Adicione a configuração `no_proxy` ao arquivo e especifique o endereço IP. O endereço IP para `no_proxy` é o endpoint dos serviços de metadados de instância (IMDS) do Systems Manager. Se você não especificar `no_proxy`, as chamadas para o Systems Manager assumirão a identidade do serviço de proxy (se o fallback do IMDSv1 estiver ativado) ou falharão (se o IMDSv2 for imposto).
>
> - Para IPv4, especifique `no_proxy=169.254.169.254`.
> - Para IPv6, especifique `no_proxy=[fd00:ec2::254]`. O endereço IPv6 do serviço de metadados da instância é compatível com comandos IMDSv2. O endereço IPv6 pode ser acessado somente em instâncias desenvolvidas no AWS Nitro System. Para obter mais informações, consulte Como o serviço de metadados de instância versão 2 funciona no Guia do usuário do Amazon EC2.

3. Salve o arquivo com o nome `amazon-ssm-agent.override` no seguinte local: `/etc/init/`
4. Encerre e reinicie o SSM Agent usando os seguintes comandos:

```
sudo service stop amazon-ssm-agent
sudo service start amazon-ssm-agent
```

> **Note**
>
> Para obter mais informações sobre como trabalhar com arquivos `.override` em ambientes Upstart, consulte init: Upstart init daemon job configuration.

Configurar o SSM Agent para usar um proxy (systemd)

Use o procedimento a seguir para configurar o SSM Agent para usar um proxy em um ambiente do `systemd`

> **Note**
>
> Algumas das etapas neste procedimento contêm instruções explícitas para instâncias do Ubuntu Server onde o SSM Agent foi instalado usando o Snap.

1. Conecte-se à instância na qual você instalou o SSM Agent.
2. Dependendo do tipo de sistema operacional em sua máquina local, execute um dos comandos a seguir.

   - Em instâncias do Ubuntu Server em que o SSM Agent é instalado usando um snap:

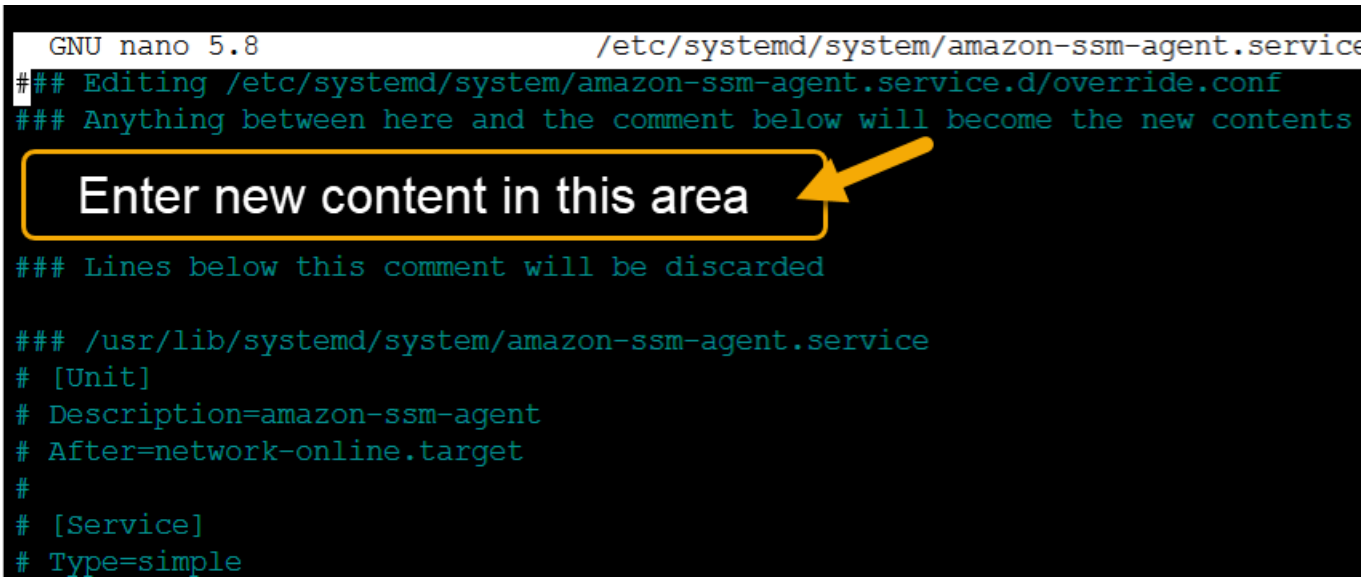
     ```
     sudo systemctl edit snap.amazon-ssm-agent.amazon-ssm-agent
     ```

     Em outros sistemas operacionais

     ```
     sudo systemctl edit amazon-ssm-agent
     ```

3. Abra um editor simples como o VIM e, caso esteja usando um servidor de proxy HTTP ou um servidor de proxy HTTPS, especifique uma das opções de configuração a seguir.

   Certifique-se de inserir as informações acima do comentário que diz "`### Lines below this comment will be discarded`", conforme indicado na imagem a seguir.

   Para um servidor proxy HTTP:

```
[Service]
Environment="http_proxy=http://hostname:port"
Environment="https_proxy=http://hostname:port"
Environment="no_proxy=IP address for instance metadata services (IMDS)"
```

Para um servidor proxy HTTP:

```
[Service]
Environment="http_proxy=http://hostname:port"
Environment="https_proxy=https://hostname:port"
Environment="no_proxy=IP address for instance metadata services (IMDS)"
```

> **Important**
>
> Adicione a configuração `no_proxy` ao arquivo e especifique o endereço IP. O endereço IP para `no_proxy` é o endpoint dos serviços de metadados de instância (IMDS) do Systems Manager. Se você não especificar `no_proxy`, as chamadas para o Systems Manager assumirão a identidade do serviço de proxy (se o fallback do IMDSv1 estiver ativado) ou falharão (se o IMDSv2 for imposto).
>
> - Para IPv4, especifique `no_proxy=169.254.169.254`.
> - Para IPv6, especifique `no_proxy=[fd00:ec2::254]`. O endereço IPv6 do serviço de metadados da instância é compatível com comandos IMDSv2. O endereço IPv6 pode ser acessado somente em instâncias desenvolvidas no AWS Nitro System. Para obter mais informações, consulte Como o serviço de metadados de instância versão 2 funciona no Guia do usuário do Amazon EC2.

4. Salve as alterações. O sistema cria automaticamente um dos arquivos a seguir, de acordo com o tipo de sistema operacional.

   - Em instâncias do Ubuntu Server em que o SSM Agent é instalado usando um snap:

     `/etc/systemd/system/snap.amazon-ssm-agent.amazon-ssm-agent.service.d/override.conf`

   - Em instâncias do Amazon Linux 2 e Amazon Linux 2023:

     `/etc/systemd/system/amazon-ssm-agent.service.d/override.conf`

- Em outros sistemas operacionais

  `/etc/systemd/system/amazon-ssm-agent.service.d/`**`amazon-ssm-agent.override`**

5. Reiniciar o SSM Agent usando um dos comandos a seguir, de acordo com o tipo de sistema operacional.

   - Em instâncias do Ubuntu Server instaladas usando um snap:

     ```
     sudo systemctl daemon-reload && sudo systemctl restart snap.amazon-ssm-agent.amazon-ssm-agent
     ```

   - Em outros sistemas operacionais

     ```
     sudo systemctl daemon-reload && sudo systemctl restart amazon-ssm-agent
     ```

> **Note**
>
> Para obter mais informações sobre como trabalhar com arquivos `.override` em ambientes systemd, consulte Modificar arquivos da unidade existente no Guia do administrador do sistema Red Hat Enterprise Linux 7.

Trabalhar com o SSM Agent em instâncias do EC2 para macOS

O AWS Systems Manager (SSM Agent) processa solicitações do Systems Manager e configura sua máquina conforme especificado na solicitação. Use os procedimentos a seguir para instalar, configurar ou desinstalar o SSM Agent para macOS.

> **Note**
>
> O SSM Agent é pré-instalado por padrão no Amazon Machine Images (AMIs) para macOS. Você não precisa instalar o SSM Agent em uma instância do Amazon Elastic Compute Cloud (Amazon EC2) para o macOS, a menos que o tenha desinstalado.

O código-fonte para o SSM Agent está disponível no GitHub para que você possa adaptar o agente de acordo com suas necessidades. Incentivamos você a enviar solicitações pull sobre alterações que gostaria que fosse incluídas. Porém, a AWS não oferece suporte à execução de cópias modificadas desse software.

> **Note**
>
> Para visualizar detalhes sobre as diferentes versões do SSM Agent, consulte as notas de release.

Antes de instalar manualmente o SSM Agent em um sistema operacional macOS, consulte as informações a seguir.

- O SSM Agent é instalado por padrão nas instâncias do EC2 e as Amazon Machine Images suportadas pelo Systems Manager. Para obter mais informações, consulte a lista de sistemas operacionais suportados para o macOS.

  O SSM Agent não precisa ser instalado manualmente nas instâncias do EC2 do macOS, a menos que ele tenha sido desinstalado.
- Não há suporte para instâncias do EC2 a macOS em todas as Regiões da AWS. Para obter listas de regiões em que há suporte a macOS para instâncias baseadas em x86 e M1 EC2, consulte Workloads macOS nas perguntas frequentes do Amazon EC2.
- Uma versão atualizada do SSM Agent é lançada sempre que novas ferramentas são adicionadas ao Systems Manager ou sempre que atualizações são feitas nas ferramentas existentes. Deixar de usar a versão mais recente do agente pode impedir que seu nó gerenciado use várias ferramentas do Systems Manager. Por isso, recomendamos automatizar o processo de manter o SSM Agent atualizado em suas máquinas. Para mais informações, consulte Automatizar atualizações do SSM Agent. Inscreva-se na página Notas de versão do SSM Agent no GitHub para receber notificações sobre atualizações do SSM Agent.

Tópicos

- Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para macOS

Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para macOS

Conecte-se à instância do macOS e execute as seguintes etapas para instalar o AWS Systems Manager Agent (SSM Agent): Realize essas etapas em cada instância que executará comandos usando o Systems Manager. Os comandos fornecidos neste procedimento também podem ser passados para instâncias do Amazon EC2 como scripts por meio de dados do usuário.

Antes de começar

Instale o `wget` usando o Homebrew.

Para instalar o SSM Agent no macOS

1. Baixe o arquivo do instalador do agente para instâncias x86_64 usando o comando a seguir.

   No comando a seguir, substitua *`region`* por suas próprias informações. Para ver uma lista dos valores de *`região`* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

   ```
   sudo wget https://s3.region.amazonaws.com/amazon-ssm-region/latest/darwin_amd64/
   amazon-ssm-agent.pkg
   ```

   Para instâncias Apple silicon use o comando a seguir.

   ```
   sudo wget https://s3.region.amazonaws.com/amazon-ssm-region/latest/darwin_arm64/
   amazon-ssm-agent.pkg
   ```

   Aqui está um exemplo.

   ```
   sudo wget https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
   darwin_amd64/amazon-ssm-agent.pkg
   ```

2. Use o seguinte comando para fazer download e executar o instalador do SSM Agent.

   x86_64:

   ```
   sudo installer -pkg amazon-ssm-agent.pkg -target /
   ```

3. Verifique o status do agente.

Para determinar se o SSM Agent está em execução, verifique o log do agente em `/var/log/amazon/ssm/amazon-ssm-agent.log`.

4. Execute o comando a seguir para iniciar o serviço se o log do agente indicar “amazon-ssm-agent is stopped”.

```
sudo launchctl load -w /Library/LaunchDaemons/com.amazon.aws.ssm.plist && sudo
 launchctl start com.amazon.aws.ssm
```

> **Important**
>
> Uma versão atualizada do SSM Agent é lançada sempre que novas ferramentas são adicionadas ao Systems Manager ou sempre que atualizações são feitas nas ferramentas existentes. Deixar de usar a versão mais recente do agente pode impedir que seu nó gerenciado use várias ferramentas do Systems Manager. Por isso, recomendamos automatizar o processo de manter o SSM Agent atualizado em suas máquinas. Para mais informações, consulte Automatizar atualizações do SSM Agent. Inscreva-se na página Notas de versão do SSM Agent no GitHub para receber notificações sobre atualizações do SSM Agent.

Desinstalar o SSM Agent de instâncias macOS

O macOS não oferece suporte de forma nativa à desinstalação dos arquivos PKG. Para desinstalar o AWS Systems Manager Agent (SSM Agent) de uma instância do Amazon Elastic Compute Cloud (Amazon EC2) para o macOS, use o script gerenciado pela AWS a partir do local a seguir.

https://github.com/aws/amazon-ssm-agent/blob/mainline/Tools/src/update/darwin/uninstall.sh

Trabalhar com o SSM Agent em instâncias do EC2 para Windows Server

AWS Systems Manager Agent (SSM Agent) é pré-instalado, por padrão, nas Amazon Machine Images (AMIs) para Windows Server fornecidas pela AWS. Há suporte para as opções de sistema operacional (SO) a seguir.

- Windows Server 2008-2012 R2 AMIs publicadas em novembro de 2016 ou mais tarde

- Windows Server 2016, 2019, 2022 (exceto versões Nano) e 2025

Notas de suporte para versões anteriores

As AMIs do Windows Server publicadas antes de novembro de 2016 usam o serviço EC2Config para processar solicitações e configurar instâncias.

A menos que você tenha um motivo específico para usar o serviço EC2Config ou uma versão anterior do SSM Agent para processar solicitações do Systems Manager, é recomendável baixar e instalar a versão mais recente do SSM Agent em cada uma de suas instâncias do Amazon Elastic Compute Cloud (Amazon EC2) ou de máquinas que não são do EC2 que estejam configuradas para o Systems Manager em um ambiente híbrido e multinuvem.

A partir de 14 de janeiro de 2020, o Windows Server 2008 não é mais compatível para obter recursos ou atualizações de segurança da Microsoft. As Amazon Machine Images (AMIs) herdadas para Windows Server 2008 e 2008 R2 ainda incluem a versão 2 do SSM Agent pré-instalada, mas o Systems Manager não é oficialmente compatível com as versões 2008 e não atualiza mais o agente para essas versões do Windows Server. Além disso, o SSM Agent versão 3 pode não ser compatível com todas as operações no Windows Server 2008 e 2008 R2. A versão final do SSM Agent oficialmente compatível com as versões 2008 do Windows Server é a 2.3.1644.0.

Manter o SSM Agent atualizado

Uma versão atualizada do SSM Agent é lançada sempre que novas ferramentas são adicionadas ao Systems Manager ou sempre que atualizações são feitas nas ferramentas existentes. Deixar de usar a versão mais recente do agente pode impedir que seu nó gerenciado use várias ferramentas do Systems Manager. Por isso, recomendamos automatizar o processo de manter o SSM Agent atualizado em suas máquinas. Para mais informações, consulte Automatizar atualizações do SSM Agent. Inscreva-se na página Notas de versão do SSM Agent no GitHub para receber notificações sobre atualizações do SSM Agent.

Para visualizar detalhes sobre as diferentes versões do SSM Agent, consulte as notas de release.

Tópicos

- Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Windows Server
- Configurar o SSM Agent para usar um proxy para instâncias do Windows Server

Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Windows Server

O AWS Systems Manager Agent (SSM Agent) é pré-instalado, por padrão, nas seguintes Amazon Machine Images (AMIs) para Windows Server, fornecidas pela Amazon:

- Windows Server 2008-2012 R2 AMIs publicadas em novembro de 2016 ou mais tarde
- Windows Server 2016, 2019, 2022 (exceto versões Nano) e 2025

Instalar o SSM Agent em instâncias do EC2 para Windows Server

Se necessário, você pode fazer download e instalar manualmente a versão mais recente do SSM Agent na instância do Amazon Elastic Compute Cloud (Amazon EC2 para Windows Server usando o procedimento a seguir. Os comandos fornecidos neste procedimento também podem ser passados para instâncias do Amazon EC2 como scripts por meio de dados do usuário.

O SSM Agent requer o Windows PowerShell 3.0 ou posterior para executar determinados documentos do AWS Systems Manager (documentos SSM) em instâncias Windows Server (por exemplo, o documento `AWS-ApplyPatchBaseline` herdado). Verifique se as suas instâncias do Windows Server estão executando o Windows Management Framework 3.0 ou posterior. Esse framework inclui o Windows PowerShell. Para obter mais informações, consulte o Windows Management Framework 3.0.

> **Note**
>
> Este procedimento se aplica à instalação ou reinstalação do SSM Agent em uma instância do EC2 para o Windows Server. Se for necessário instalar o agente em um servidor ou em uma máquina virtual (VM) on-premises para que ele possa ser usado com o Systems Manager, consulte Como instalar o SSM Agent para nós híbridos do Windows.

Para instalar manualmente a versão mais recente do SSM Agent em instâncias do EC2 para o Windows Server

1. Conecte-se à sua instância usando a área de trabalho remota ou o Windows PowerShell. Para obter mais informações, consulte Conectar-se à sua instância no Guia do usuário do Amazon EC2.

2. Faça download da versão mais recente do SSM Agent para sua instância. Você pode fazer download usando os comandos do PowerShell ou um link de download direto.

> **Note**
>
> Os URLs nesta etapa permitem que você baixe o SSM Agent de qualquer Região da AWS. Para fazer download do agente de uma região específica, em vez disso, use um URL específico da região: `https://amazon-ssm-region.s3.region.amazonaws.com/latest/windows_amd64/AmazonSSMAgentSetup.exe`
> A *região* representa o identificador da região para uma região da Região da AWS compatível com o AWS Systems Manager, como `us-east-2` para a região Leste dos EUA (Ohio). Para ver uma lista dos valores de *região* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

PowerShell

Execute os três comandos do PowerShell a seguir em ordem. Esses comandos permitem que você faça download do SSM Agent sem ajustar as configurações de Segurança Avançada do Internet Explorer (IE) e, depois, instale o agente e remova o arquivo de instalação.

64-bit

```
[System.Net.ServicePointManager]::SecurityProtocol = 'TLS12'
$progressPreference = 'silentlyContinue'
Invoke-WebRequest `
    https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/windows_amd64/AmazonSSMAgentSetup.exe `
    -OutFile $env:USERPROFILE\Desktop\SSMAgent_latest.exe
```

32-bit

```
[System.Net.ServicePointManager]::SecurityProtocol = 'TLS12'
$progressPreference = 'silentlyContinue'
Invoke-WebRequest `
    https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/windows_386/AmazonSSMAgentSetup.exe `
```

```
    -OutFile $env:USERPROFILE\Desktop\SSMAgent_latest.exe
```

```
Start-Process `
    -FilePath $env:USERPROFILE\Desktop\SSMAgent_latest.exe `
    -ArgumentList "/S" `
    -Wait
```

```
rm -Force $env:USERPROFILE\Desktop\SSMAgent_latest.exe
```

Download direto

Faça download da versão mais recente do SSM Agent para sua instância usando o link a seguir. Se preferir, substitua esse URL por um URL específico da Região da AWS.

https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/windows_amd64/AmazonSSMAgentSetup.exe

Execute o arquivo `AmazonSSMAgentSetup.exe` obtido por download para instalar o SSM Agent.

3. Inicie ou reinicie o SSM Agent enviando o seguinte comando no PowerShell:

```
Restart-Service AmazonSSMAgent
```

> **Note**
>
> Para desinstalar o SSM Agent de uma instância do Windows Server abra o Painel de Controle, Programas. Escolha a opção Uninstall a program (Desinstalar um programa). Abra o menu contextual (clique com o botão direito) para o Amazon SSM Agent e escolha Uninstall (Desinstalar).

Configurar o SSM Agent para usar um proxy para instâncias do Windows Server

As informações neste tópico se aplicam a instâncias do Windows Server criadas a partir de novembro de 2016 que não usam a opção de instalação Nano. Se você pretende usar o Session Manager, observe que os servidores proxy HTTPS não são compatíveis.

> **Note**
>
> A partir de 14 de janeiro de 2020, o Windows Server 2008 não é mais compatível para obter recursos ou atualizações de segurança da Microsoft. As Amazon Machine Images (AMIs) herdadas para Windows Server 2008 e 2008 R2 ainda incluem a versão 2 do SSM Agent pré-instalada, mas o Systems Manager não é oficialmente compatível com as versões 2008 e não atualiza mais o agente para essas versões do Windows Server. Além disso, o SSM Agent versão 3 pode não ser compatível com todas as operações no Windows Server 2008 e 2008 R2. A versão final do SSM Agent oficialmente compatível com as versões 2008 do Windows Server é a 2.3.1644.0.

Antes de começar

Antes de configurar o SSM Agent para usar um proxy, observe as informações importantes a seguir.

No procedimento a seguir, execute um comando para configurar o SSM Agent para usar um proxy. O comando inclui uma configuração `no_proxy` com um endereço IP. O endereço IP é o endpoint dos serviços de metadados de instância (IMDS) do Systems Manager. Se você não especificar `no_proxy`, as chamadas para o Systems Manager assumirão a identidade do serviço de proxy (se o fallback do IMDSv1 estiver ativado) ou falharão (se o IMDSv2 for imposto).

- Para IPv4, especifique `no_proxy=169.254.169.254`.
- Para IPv6, especifique `no_proxy=[fd00:ec2::254]`. O endereço IPv6 do serviço de metadados da instância é compatível com comandos IMDSv2. O endereço IPv6 pode ser acessado somente em instâncias desenvolvidas no AWS Nitro System. Para obter mais informações, consulte Como o serviço de metadados de instância versão 2 funciona no Guia do usuário do Amazon EC2.

Para configurar o SSM Agent para usar um proxy

1. Usando o Desktop Remoto ou o Windows PowerShell, conecte-se à instância que você deseja configurar para usar um proxy.
2. Execute o seguinte bloco de comandos no PowerShell. Substitua o *`nome do host`* e a *`porta`* por informações sobre o seu proxy:

```
$serviceKey = "HKLM:\SYSTEM\CurrentControlSet\Services\AmazonSSMAgent"
$keyInfo = (Get-Item -Path $serviceKey).GetValue("Environment")
```

```
$proxyVariables = @("http_proxy=hostname:port", "https_proxy=hostname:port",
 "no_proxy=IP address for instance metadata services (IMDS)")

if ($keyInfo -eq $null) {
    New-ItemProperty -Path $serviceKey -Name Environment -Value $proxyVariables -
PropertyType MultiString -Force
} else {
    Set-ItemProperty -Path $serviceKey -Name Environment -Value $proxyVariables
}

Restart-Service AmazonSSMAgent
```

Depois que executar o comando anterior, você poderá examinar os logs do SSM Agent para confirmar se as configurações de proxy foram aplicadas. As entradas nos logs são semelhantes ao seguinte: Para obter mais informações sobre logs do SSM Agent, consulte Visualizar logs do SSM Agent.

```
2020-02-24 15:31:54 INFO Getting IE proxy configuration for current user: The operation
 completed successfully.
2020-02-24 15:31:54 INFO Getting WinHTTP proxy default configuration: The operation
 completed successfully.
2020-02-24 15:31:54 INFO Proxy environment variables:
2020-02-24 15:31:54 INFO http_proxy: hostname:port
2020-02-24 15:31:54 INFO https_proxy: hostname:port
2020-02-24 15:31:54 INFO no_proxy: IP address for instance metadata services (IMDS)
2020-02-24 15:31:54 INFO Starting Agent: amazon-ssm-agent - v2.3.871.0
2020-02-24 15:31:54 INFO OS: windows, Arch: amd64
```

Para redefinir a configuração de proxy do SSM Agent

1. Usando o Desktop Remoto ou o Windows PowerShell, conecte-se à instância a ser configurada.
2. Se você se conectou usando o Desktop Remoto, execute o PowerShell como administrador.
3. Execute o seguinte bloco de comandos no PowerShell.

```
Remove-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Services\AmazonSSMAgent -
Name Environment
Restart-Service AmazonSSMAgent
```

Precedência de configuração de proxy do SSM Agent

Ao definir as configurações de proxy para o SSM Agent nas instâncias do Windows Server, é importante entender que essas configurações são avaliadas e aplicadas à configuração do agente quando o SSM Agent é iniciado. A maneira como você define as configurações de proxy para uma instância do Windows Server pode determinar se outras configurações podem prevalecer sobre as configurações pretendidas. O agente usa as primeiras configurações de proxy encontradas.

> **Important**
>
> O SSM Agent comunica-se usando o protocolo HTTPS. Por esse motivo, você deve configurar o parâmetro `HTTPS proxy` usando uma das opções de configurações a seguir.

As configurações de proxy do SSM Agent são avaliadas na ordem a seguir.

1. Configurações do registro do `AmazonSSMAgent` (`HKLM:\SYSTEM\CurrentControlSet\Services\AmazonSSMAgent`)
2. Variáveis de ambiente de `System` (`http_proxy`, `https_proxy`, `no_proxy`)
3. Variáveis de ambiente da conta de usuário `LocalSystem` (`http_proxy`, `https_proxy` e `no_proxy`)
4. Configurações do navegador (`HTTP`, `secure`, `exceptions`)
5. Configuração de proxy do `WinHTTP` (`http=`, `https=`, `bypass-list=`)

As configurações de proxy do SSM Agent e serviços do Systems Manager

Se você tiver configurado o SSM Agent para usar um proxy e estiver usando ferramentas do AWS Systems Manager, como o Run Command e o Patch Manager, que usam o PowerShell ou o cliente do Windows Update durante sua execução em instâncias Windows Server, defina também as configurações adicionais de proxy. Caso contrário, pode haver falha na operação porque as configurações de proxy usadas pelo PowerShell e o cliente do Windows Update não são herdadas da configuração de proxy do SSM Agent.

Para o Run Command, defina as configurações de proxy do `WinINet` nas instâncias do Windows Server. Os comandos `[System.Net.WebRequest]` fornecidos são por sessão. Para aplicar essas configurações a comandos de rede subsequentes que são executados no Run Command,

esses comandos devem preceder outros comandos do PowerShell na mesma entrada do plugin `aws:runPowershellScript`.

Os comandos do PowerShell a seguir retornam as configurações de proxy atuais do `WinINet` e aplicam suas configurações de proxy ao `WinINet`.

```
[System.Net.WebRequest]::DefaultWebProxy

$proxyServer = "http://hostname:port"
$proxyBypass = "169.254.169.254"
$WebProxy = New-Object System.Net.WebProxy($proxyServer,$true,$proxyBypass)

[System.Net.WebRequest]::DefaultWebProxy = $WebProxy
```

Para o Patch Manager, defina as configurações de proxy de todo o sistema para que o cliente do Windows Update possa verificar e baixar as atualizações. Recomendamos usar o Run Command para executar os seguintes comandos porque eles são executados na conta SYSTEM e as configurações se aplicam a todo o sistema. Os comandos do `netsh` a seguir retornam as configurações de proxy atuais e aplicam suas configurações de proxy ao sistema local.

```
netsh winhttp show proxy

netsh winhttp set proxy proxy-server="hostname:port" bypass-list="169.254.169.254"
```

Para obter mais informações sobre o uso de Run Command, consulte AWS Systems Manager Run Command.

Verificar o status do SSM Agent e iniciar o agente

Este tópico lista os comandos para verificar se o AWS Systems Manager Agent (SSM Agent) está em execução em cada sistema operacional compatível. Ele também fornece os comandos para iniciar o agente se ele não estiver em execução.

Sistema operacional	Comando para verificar o status do SSM Agent	Comando para iniciar o SSM Agent
Amazon Linux 1	`sudo status amazon-ssm-agent`	`sudo start amazon-ssm-agent`

Sistema operacional	Comando para verificar o status do SSM Agent	Comando para iniciar o SSM Agent
Amazon Linux 2, Amazon Linux 2022 e Amazon Linux 2023	`sudo systemctl status amazon-ssm-agent`	`sudo systemctl enable amazon-ssm-agent` `sudo systemctl start amazon-ssm-agent`
CentOS 6.x	`sudo status amazon-ssm-agent`	`sudo start amazon-ssm-agent`
CentOS 7.x e CentOS 8.x	`sudo systemctl status amazon-ssm-agent`	`sudo systemctl enable amazon-ssm-agent` `sudo systemctl start amazon-ssm-agent`
Debian Server 8, 9 e 10	`sudo systemctl status amazon-ssm-agent`	`sudo systemctl enable amazon-ssm-agent` `sudo systemctl start amazon-ssm-agent`
macOS	Verifique o arquivo de log do agente em `/var/log/amazon/ssm/amazon-ssm-agent.log`	`sudo launchctl load -w /Library/LaunchDaemons/com.amazon.aws.ssm.plist` `sudo launchctl start com.amazon.aws.ssm`
Oracle Linux	`sudo systemctl status amazon-ssm-agent`	`sudo systemctl enable amazon-ssm-agent` `sudo systemctl start amazon-ssm-agent`

Sistema operacional	Comando para verificar o status do SSM Agent	Comando para iniciar o SSM Agent
Red Hat Enterprise Linux (RHEL) 6.x	`sudo status amazon-ssm-agent`	`sudo start amazon-ssm-agent`

Red Hat Enterprise Linux (RHEL) 7.x, 8.x, 9.x e 10.x	`sudo systemctl status amazon-ssm-agent`	`sudo systemctl enable amazon-ssm-agent`
		
		`sudo systemctl start amazon-ssm-agent`

SUSE Linux Enterprise Server (SLES)	`sudo systemctl status amazon-ssm-agent`	`sudo systemctl enable amazon-ssm-agent`
		
		`sudo systemctl start amazon-ssm-agent`

Ubuntu Server 14.04 (todos) e 16.04 (32 bits)	`sudo status amazon-ssm-agent`	`sudo start amazon-ssm-agent`

Instâncias do Ubuntu Server 16.04 de 64 bits (instalação do pacote deb)	`sudo systemctl status amazon-ssm-agent`	`sudo systemctl enable amazon-ssm-agent`
		
		`sudo systemctl start amazon-ssm-agent`

Ubuntu Server 16.04 LTS, 18.04, 20.04, 20.10 STR, 22.04 LTS, 23.10, 24.04 LTS, 24.0 e 25.04	`sudo systemctl status snap.amazon-ssm-agent.amazon-ssm-agent.service`	`sudo snap start amazon-ssm-agent`
Windows Server	Executar no PowerShell: `Get-Service AmazonSSMAgent`	Executar no modo Administrador do PowerShell: `Start-Service AmazonSSMAgent`

Mais informações

- Trabalhar com o SSM Agent em instâncias do EC2 para Linux
- Trabalhar com o SSM Agent em instâncias do EC2 para Windows Server
- Verificar o número de versão do SSM Agent

Verificar o número de versão do SSM Agent

Determinadas funcionalidades do AWS Systems Manager têm pré-requisitos que incluem uma versão mínima do Systems Manager Agent (SSM Agent) instalada nos nós gerenciados. É possível obter a versão do SSM Agent instalada no momento nos nós gerenciados usando o console do Systems Manager ou fazendo login neles.

> **Note**
>
> Para ver detalhes sobre versões anteriores, consulte as Notas de lançamento do SSM Agent no GitHub.

Os procedimentos a seguir descrevem como obter a versão do SSM Agent instalada atualmente em seus nós gerenciados.

Para verificar o número de versão do SSM Agent instalado em um nó gerenciado

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Na coluna Versão do SSM Agent, anote o número da Versão do agente.

Como obter a versão do SSM Agent instalada no momento no sistema operacional

Escolha uma das guias a seguir para obter do sistema operacional a versão do SSM Agent instalada atualmente.

Amazon Linux 1, Amazon Linux 2, and Amazon Linux 2023

> **Note**
>
> Esse comando varia de acordo com o gerenciador de pacotes do sistema operacional.

1. Faça login em seu nó gerenciado.
2. Execute o seguinte comando:

```
yum info amazon-ssm-agent
```

Esse comando retorna uma saída semelhante à seguinte:

```
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Installed Packages
Name        : amazon-ssm-agent
Arch        : x86_64
Version     : 3.0.655.0
```

CentOS

1. Faça login em seu nó gerenciado.
2. Execute o seguinte comando para 6 e 7.

```
yum info amazon-ssm-agent
```

Esse comando retorna uma saída semelhante à seguinte:

```
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Installed Packages
Name        : amazon-ssm-agent
Arch        : x86_64
Version     : 3.0.655.0
```

Debian Server

1. Faça login em seu nó gerenciado.
2. Execute o seguinte comando:

```
apt list amazon-ssm-agent
```

Esse comando retorna uma saída semelhante à seguinte:

```
apt list amazon-ssm-agent
Listing... Done
amazon-ssm-agent/now 3.0.655.0-1 amd64 [installed,local]

3.0.655.0 is the version of SSM agent
```

macOS

1. Faça login em seu nó gerenciado.
2. Execute o seguinte comando:

```
pkgutil --pkg-info com.amazon.aws.ssm
```

RHEL

1. Faça login em seu nó gerenciado.
2. Execute o seguinte comando para RHEL 6, 7, 8 e 9.

```
yum info amazon-ssm-agent
```

Esse comando retorna uma saída semelhante à seguinte:

```
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Installed Packages
Name        : amazon-ssm-agent
Arch        : x86_64
Version     : 3.0.655.0
```

Execute o comando a seguir para o utilitário de pacote do DNF.

```
dnf info amazon-ssm-agent
```

SLES

1. Faça login em seu nó gerenciado.

2. Execute o comando a seguir para SLES 12 e 15.

```
zypper info amazon-ssm-agent
```

Esse comando retorna uma saída semelhante à seguinte:

```
Loading repository data...
Reading installed packages...
Information for package amazon-ssm-agent:
----------------------------------------
Repository : @System
Name : amazon-ssm-agent
Version : 3.0.655.0-1
```

Ubuntu Server

> **Note**
>
> Para verificar se a instância do Ubuntu Server 16.04 usa pacotes deb ou Snap, consulte Instalar o SSM Agent manualmente em instâncias do Ubuntu Server.

1. Faça login em seu nó gerenciado.
2. Execute o comando a seguir para o Ubuntu Server 16.04 e 14.04 de 64 bits (com pacote instalador deb).

```
apt list amazon-ssm-agent
```

Esse comando retorna uma saída semelhante à seguinte:

```
apt list amazon-ssm-agent
Listing... Done
amazon-ssm-agent/now 3.0.655.0-1 amd64 [installed,local]

3.0.655.0 is the version of SSM agent
```

Execute o comando a seguir para o Ubuntu Server 16.04 LTS de 64 bits (Snap), 18.04, 20.04, 20.10 STR, 22.04 LTS, 23.10, 24.04 LTS, 24.0 e 25.04.

```
sudo snap list amazon-ssm-agent
```

Esse comando retorna uma saída semelhante à seguinte:

```
snap list amazon-ssm-agent
Name Version Rev Tracking Publisher Notes
amazon-ssm-agent 3.0.529.0 3552 latest/stable/… aws# classic-

3.0.529.0 is the version of SSM agent
```

Windows

1. Faça login em seu nó gerenciado.
2. Execute o seguinte comando do PowerShell.

```
& "C:\Program Files\Amazon\SSM\amazon-ssm-agent.exe" -version
```

Esse comando retorna uma saída semelhante à seguinte:

```
SSM Agent version: 3.1.804.0
```

Recomendamos o uso da versão mais recente do SSM Agent para que você possa se beneficiar dos recursos novos ou atualizados. Para garantir que as instâncias gerenciadas estejam sempre executando a versão mais atualizada do SSM Agent, você pode automatizar o processo de atualização do SSM Agent. Para obter mais informações, consulte Automatizar atualizações do SSM Agent.

Visualizar logs do SSM Agent

O AWS Systems Manager Agent (SSM Agent) grava informações sobre execuções, comandos, ações programadas, erros e status de integridade nos arquivos de log de cada nó gerenciado. Você pode visualizar arquivos de log conectando-se manualmente a um nó gerenciado ou pode enviar logs automaticamente para o Amazon CloudWatch Logs. Para obter mais informações sobre o envio de logs ao CloudWatch Logs, consulte Registrar em log e monitorar no AWS Systems Manager.

Você pode visualizar logs do SSM Agent em nós gerenciados nos locais a seguir.

Linux and macOS

```
/var/log/amazon/ssm/
```

Windows

```
%PROGRAMDATA%\Amazon\SSM\Logs\
```

Para nós gerenciados do Linux, os arquivos SSM Agent, `stderr` e `stdout` são gravados no seguinte diretório: `/var/lib/amazon/ssm/`.

Para nós gerenciados do Windows, os arquivos SSM Agent, `stderr` e `stdout` são gravados no seguinte diretório: `%PROGRAMDATA%\Amazon\SSM\InstanceData\`.

Para obter informações sobre como habilitar o log de depuração do SSM Agent, consulte Permitir o registro em log de depuração do SSM Agent.

Para obter mais informações sobre a configuração de `cihub/seelog`, consulte Seelog Wiki no GitHub. Para obter exemplos de configurações de `cihub/seelog`, consulte o repositório de exemplos de cihub/seelog no GitHub.

Permitir o registro em log de depuração do SSM Agent

Use o procedimento a seguir para permitir o registro em log da depuração do SSM Agent em seus nós gerenciados.

Linux and macOS

Para permitir o registro de depuração do SSM Agent em nós gerenciados do Linux e macOS

1. Use o Session Manager, uma ferramenta do AWS Systems Manager, para conectar-se ao nó gerenciado no qual você deseja permitir o log de depuração ou faça login em seu nó gerenciado. Para ter mais informações, consulte Trabalhar com o Session Manager.
2. Localize o arquivo seelog.xml.template.

   Linux

   Na maioria dos tipos de nós gerenciados do Linux, o arquivo está localizado no diretório `/etc/amazon/ssm/seelog.xml.template`.

   No Ubuntu Server 20.10 STR e 20.04, 18.04 e 16.04 LTS, o arquivo está localizado no diretório `/snap/amazon-ssm-agent/current/seelog.xml.template`. Copie esse

arquivo do diretório `/snap/amazon-ssm-agent/current/` no diretório `/etc/amazon/ssm/` antes de fazer quaisquer alterações.

macOS:

Nos tipos de instância do macOS, o arquivo está localizado no diretório `/opt/aws/ssm/seelog.xml.template`.

3. Altere o nome do arquivo de `seelog.xml.template` para `seelog.xml`.

> **Note**
>
> No Ubuntu Server 20.10 STR & 20.04, 18.04 e 16.04 LTS, o arquivo `seelog.xml` deve ser criado no diretório `/etc/amazon/ssm/`. Você pode criar esse diretório e arquivo executando os comandos a seguir.
>
> ```
> sudo mkdir -p /etc/amazon/ssm
> ```
>
> ```
> sudo cp -p /snap/amazon-ssm-agent/current/seelog.xml.template /etc/amazon/ssm/seelog.xml
> ```

4. Edite o arquivo `seelog.xml` para alterar o comportamento de registro em log padrão. Altere o valor do minlevel de info para depurar, conforme mostrado no exemplo a seguir.

```
<seelog type="adaptive" mininterval="2000000" maxinterval="100000000" critmsgcount="500" minlevel="debug">
```

5. (Opcional) Reinicie o SSM Agent usando o comando a seguir.

Linux

```
sudo service amazon-ssm-agent restart
```

macOS:

```
sudo /opt/aws/ssm/bin/amazon-ssm-agent restart
```

Windows

Para permitir o registro de depuração do SSM Agent em nós gerenciados do Windows Server

1. Use o Session Manager para conectar-se ao nó gerenciado no qual você deseja permitir o log de depuração ou faça login nos nós gerenciados. Para ter mais informações, consulte Trabalhar com o Session Manager.
2. Faça uma cópia do arquivo seelog.xml.template. Altere o nome da cópia para seelog.xml. O arquivo está localizado no diretório a seguir.

   ```
   %PROGRAMFILES%\Amazon\SSM\seelog.xml.template
   ```
3. Edite o arquivo `seelog.xml` para alterar o comportamento de registro em log padrão. Altere o valor do minlevel de info para depurar, conforme mostrado no exemplo a seguir.

   ```
   <seelog type="adaptive" mininterval="2000000" maxinterval="100000000"
   critmsgcount="500" minlevel="debug">
   ```
4. Localize a seguinte entrada:

   ```
   filename="{{LOCALAPPDATA}}\Amazon\SSM\Logs\{{EXECUTABLENAME}}.log"
   ```

   Altere essa entrada para usar o seguinte caminho:

   ```
   filename="C:\ProgramData\Amazon\SSM\Logs\{{EXECUTABLENAME}}.log"
   ```
5. Localize a seguinte entrada:

   ```
   filename="{{LOCALAPPDATA}}\Amazon\SSM\Logs\errors.log"
   ```

   Altere essa entrada para usar o seguinte caminho:

   ```
   filename="C:\ProgramData\Amazon\SSM\Logs\errors.log"
   ```
6. Reinicie o SSM Agent usando o comando do PowerShell a seguir no modo Administrador.

   ```
   Restart-Service AmazonSSMAgent
   ```

Restringir o acesso aos comandos em nível raiz por meio do SSM Agent

O AWS Systems Manager Agent (SSM Agent) é executado em instâncias do Amazon Elastic Compute Cloud (Amazon EC2) e em outros tipos de máquina em ambientes híbridos e multinuvem usando permissões raiz (Linux) ou permissões SYSTEM (Windows Server). Como há o nível mais alto de permissões de acesso ao sistema, qualquer entidade confiável que tenha recebido permissão para enviar comandos ao SSM Agent terá permissões raiz ou SYSTEM. (Na AWS, uma entidade confiável que pode executar ações e acessar recursos na AWS é chamada de entidade principal. Uma entidade principal pode ser um Usuário raiz da conta da AWS, um usuário ou um perfil.)

Esse nível de acesso é necessário para uma entidade principal enviar comandos autorizados do Systems Manager ao SSM Agent, mas também possibilita que uma entidade principal execute código mal-intencionado explorando vulnerabilidades potenciais no SSM Agent.

Especificamente, as permissões para executar os comandos SendCommand e StartSession devem ser cuidadosamente restritas. Um bom primeiro passo é conceder permissões para cada comando apenas a entidades principais selecionadas em sua organização. No entanto, recomendamos reforçar ainda mais seu procedimento de segurança restringindo em quais nós gerenciados uma entidade principal pode executar esses comandos. Isso pode ser feito na política do IAM atribuída à entidade principal. Na política do IAM, você pode incluir uma condição que limita o usuário a executar comandos apenas em nós gerenciados marcados com tags específicas ou combinações de tags.

Por exemplo, digamos que você tenha duas frotas de servidores, uma para teste e outra para produção. Na política do IAM aplicada a engenheiros júniores, você especifica que eles podem executar comandos apenas em instâncias marcadas com `ssm:resourceTag/testServer`. Mas, para um grupo menor de engenheiros líderes, que devem ter acesso a todas as instâncias, você concede acesso às instâncias marcadas com `ssm:resourceTag/testServer` e `ssm:resourceTag/productionServer`.

Usando essa abordagem, se os engenheiros júniores tentarem executar um comando em uma instância de produção, eles terão o acesso negado, porque a política do IAM atribuída não fornece acesso explícito a instâncias marcadas com `ssm:resourceTag/productionServer`.

Para obter mais informações e exemplos, veja estes tópicos:

- Restringir o acesso ao Run Command com base em etiquetas
- Restringir o acesso à sessão com base em tags de instância

Automatizar atualizações do SSM Agent

A AWS lança uma nova versão do AWS Systems Manager Agent (SSM Agent) quando adicionamos ou atualizamos ferramentas do Systems Manager. Se os nós gerenciados usarem uma versão mais antiga do agente, você não poderá usar as novas ferramentas nem se beneficiar das ferramentas atualizadas. Por esses motivos, recomendamos que você automatize o processo de atualização do SSM Agent em seus nós gerenciados, usando qualquer um dos seguintes métodos.

Atualizações do agente no sistema operacional Bottlerocket

Não é possível atualizar o SSM Agent no sistema operacional Bottlerocket usando o documento Systems Manager Command `AWS-UpdateSSMAgent`. As atualizações são gerenciadas dentro do contêiner de controle do Bottlerocket. Para obter mais informações, consulte Contêiner de controle do Bottlerocket e Operador de atualização do Bottlerocket no GitHub.

Requisitos de versão do macOS

Se uma instância estiver sendo executada no macOS versão 11.0 (Big Sur) ou posterior, a instância deverá ter o SSM Agent versão 3.1.941.0 ou posterior para executar o documento AWS-UpdateSSMAgent. Se a instância estiver executando uma versão do SSM Agent lançada antes da 3.1.941.0, atualize o SSM Agent para executar o documento AWS-UpdateSSMAgent executando os comandos `brew upgrade amazon-ssm-agent` e `brew update`.

Método	Detalhes
Atualização automatizada com um clique em todas os nós gerenciados (recomendado)	Você pode configurar todos os nós gerenciad os em sua Conta da AWS para verificar e baixar automaticamente novas versões do SSM Agent. Para isso, escolha Atualização automática do SSM Agent na guia Configura ções no Fleet Manager, conforme descrito posteriormente neste tópico.
Atualização global ou seletiva	Você pode usar o State Manager, uma ferramenta do AWS Systems Manager, para criar uma associação que baixe e instale automaticamente o SSM Agent em seus nós gerenciados. Se quiser limitar a interrupç

Método	Detalhes
	ão às cargas de trabalho, crie uma janela de manutenção do Systems Manager para executar a instalação durante períodos designados. Ambos os métodos permitem que você crie uma configuração de atualizaç ão global para todos os nós gerenciados ou escolha seletivamente quais instâncias serão atualizadas. Para obter informações sobre como criar uma associação do State Manager, consulte Walkthrough: Automatically update SSM Agent with the AWS CLI. Para obter informações sobre como criar uma janela de manutenção, consulte Tutorial: Create a maintenance window for patching using the console.
Atualização global ou seletiva para novos ambientes	Se você estiver começando a usar o Systems Manager, recomendamos usar a opção Atualizar o Systems Manager (SSM) Agent a cada duas semanas no Quick Setup, uma ferramenta do AWS Systems Manager. O Quick Setup permite que você crie uma configuração de atualização global para todas os nós gerenciados ou escolha seletivamente quais nós gerenciados serão atualizados. Para obter mais informações, consulte Configura r o gerenciamento de host do Amazon EC2 usando a Quick Setup.

Se você preferir atualizar o SSM Agent em seus nós gerenciados manualmente, poderá assinar as notificações que a AWS publica quando uma nova versão do agente é lançada. Para mais informações, consulte Assinar as notificações do SSM Agent. Depois de assinar notificações, você pode usar o Run Command para atualizar manualmente uma ou mais nós gerenciados com a versão

mais recente. Para obter mais informações, consulte Atualização do SSM Agent por meio de Run Command.

Atualizar automaticamente o SSM Agent

Você pode configurar o Systems Manager para atualizar automaticamente o SSM Agent em todos os nós gerenciados com base em Linux e Windows em sua Conta da AWS. Se você ativar essa opção, o Systems Manager verificará automaticamente a cada duas semanas se há uma nova versão do agente. Se houver uma nova versão, o Systems Manager atualizará automaticamente o agente para a última versão lançada usando o documento SSM `AWS-UpdateSSMAgent`. Recomendamos escolher essa opção para garantir que os nós gerenciados estejam sempre executando a versão mais atualizada do SSM Agent.

> **Note**
>
> Se você usar um comando `yum` para atualizar o SSM Agent em um nó gerenciado após o agente ter sido instalado ou atualizado usando o documento `AWS-UpdateSSMAgent` do SSM, você poderá ver a seguinte mensagem: `"Warning: RPMDB altered outside of yum."` Esta mensagem é esperada e pode ser ignorada com segurança.

Como atualizar automaticamente o SSM Agent

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Fleet Manager.
3. Escolha a guia Configurações.
4. Na área Atualização automática do agente, escolha Atualização automática do SSM Agent.

Para alterar a versão do SSM Agent para a qual a frota é atualizada, escolha Edit (Editar) em Agent auto update (Atualização automática do agente) na guia Settings (Configurações). Em seguida, insira o número da versão do SSM Agent para a qual você deseja atualizar na opção Version (Versão) em Parameters (Parâmetros). Se não for especificado, o agente será atualizado para a versão mais recente.

Para interromper a implantação automática de versões atualizadas do SSM Agent em todos os nós gerenciados da conta, escolha Delete (Excluir) em Agent auto update (Atualização automática do

agente) na guia Settings (Configurações). Essa ação exclui a associação do State Manager que atualiza automaticamente o SSM Agent em seus nós gerenciados.

Assinar as notificações do SSM Agent

O Amazon Simple Notification Service (Amazon SNS) notifica você quando novas versões do Agente do AWS Systems Manager (SSM Agent) forem lançadas. Use o procedimento a seguir para se inscrever nessas notificações.

> **Tip**
>
> Você também pode se inscrever em notificações observando a página Notas de versão do SSM Agent no GitHub.

Para assinar as notificações do SSM Agent

1. Abra o console do Amazon SNS em https://console.aws.amazon.com/sns/v3/home.
2. No seletor de regiões na barra de navegação, selecione Leste dos EUA (Norte da Virgínia), caso a região ainda não esteja selecionada. Selecione essa Região da AWS, pois as notificações do Amazon SNS para o SSM Agent em que você está se inscrevendo são geradas somente nessa região.
3. No painel de navegação, escolha Subscriptions.
4. Selecione Create subscription.
5. Em Create subscription (Criar inscrição), faça o seguinte:
   a. Para Topic ARN (ARN do tópico), use o seguinte nome do recurso da Amazon (ARN):

      ```
      arn:aws:sns:us-east-1:720620558202:SSM-Agent-Update
      ```

   b. Para Protocol (Protocolo), escolha `Email` ou `SMS`.
   c. Para Endpoint, dependendo se você escolheu `Email` ou `SMS` na etapa anterior, insira um endereço de e-mail ou um código de área e número para receber notificações.
   d. Selecione Criar assinatura.
6. Se selecionar `Email`, você receberá um e-mail solicitando que você confirme sua inscrição. Abra o e-mail e siga as instruções para concluir a sua assinatura.

Sempre que uma nova versão do SSM Agent for lançada, enviaremos notificações aos inscritos. Se não deseja mais receber essas notificações, use o procedimento a seguir para cancelar a assinatura.

Para cancelar a assinatura de notificações do SSM Agent

1. Abra o console do Amazon SNS.
2. No painel de navegação, escolha Subscriptions.
3. Selecione a assinatura e, em seguida, Delete (Excluir). Quando a confirmação for solicitada, escolha Excluir.

Solução de problemas do SSM Agent

Se você tiver problemas ao executar operações em seus nós gerenciados, poderá haver um problema com o AWS Systems Manager Agent (SSM Agent). Use as seguintes informações para ajudá-lo a visualizar os arquivos de log do SSM Agent e solucionar o problema com o agente.

Tópicos

- O SSM Agent está desatualizado
- Solucionar problemas usando arquivos de log do SSM Agent
- Arquivos de log do agente não alternam (Windows)
- Unable to connect to endpoint
- Verificar sua configuração de VPC
- Verificar seus atributos relacionados a DNS da VPC
- Verificar regras de entrada e saída em grupos de segurança de endpoints
- Usa a ssm-cli para solucionar problemas de disponibilidade do nó gerenciado

O SSM Agent está desatualizado

Uma versão atualizada do SSM Agent é lançada sempre que novas ferramentas são adicionadas ao Systems Manager ou sempre que atualizações são feitas nas ferramentas existentes. Deixar de usar a versão mais recente do agente pode impedir que seu nó gerenciado use várias ferramentas do Systems Manager. Por isso, recomendamos automatizar o processo de manter o SSM Agent atualizado em suas máquinas. Para mais informações, consulte Automatizar atualizações do SSM Agent. Inscreva-se na página Notas de versão do SSM Agent no GitHub para receber notificações sobre atualizações do SSM Agent.

Solucionar problemas usando arquivos de log do SSM Agent

SSM Agent registra informações nos arquivos a seguir. As informações nesses arquivos podem ajudar a solucionar problemas. Para obter mais informações sobre os arquivos de log do SSM Agent, incluindo como ativar o log de depuração, consulte Visualizar logs do SSM Agent.

> **Note**
>
> Se você optar por visualizar esses logs usando o Explorador de arquivos do Windows, certifique-se de habilitar a visualização de arquivos ocultos e arquivos do sistema nas Opções de pasta.

No Windows

- `%PROGRAMDATA%\Amazon\SSM\Logs\amazon-ssm-agent.log`
- `%PROGRAMDATA%\Amazon\SSM\Logs\errors.log`

No Linux e no macOS

- `/var/log/amazon/ssm/amazon-ssm-agent.log`
- `/var/log/amazon/ssm/errors.log`

Para nós gerenciados do Linux, você pode encontrar mais informações no arquivo `messages` gravado no seguinte diretório: `/var/log`.

Para obter mais informações sobre a solução de problemas usando logs do agente, consulte How do I use SSM Agent logs to troubleshoot issues with SSM Agent in my managed instance? no Centro de Conhecimento AWS re:Post.

Arquivos de log do agente não alternam (Windows)

Se você especificar a rotação do arquivo de log baseado em data no arquivo seelog.xml (em nós gerenciados do Windows Server) e os logs não forem alternados, especifique o parâmetro `fullname=true`. Veja a seguir um exemplo de um arquivo de configuração seelog.xml com o parâmetro `fullname=true` especificado.

```
<seelog type="adaptive" mininterval="2000000" maxinterval="100000000"
 critmsgcount="500" minlevel="debug">
   <exceptions>
      <exception filepattern="test*" minlevel="error" />
   </exceptions>
   <outputs formatid="fmtinfo">
      <console formatid="fmtinfo" />
      <rollingfile type="date" datepattern="200601021504" maxrolls="4" filename="C:
\ProgramData\Amazon\SSM\Logs\amazon-ssm-agent.log" fullname=true />
      <filter levels="error,critical" formatid="fmterror">
         <rollingfile type="date" datepattern="200601021504" maxrolls="4" filename="C:
\ProgramData\Amazon\SSM\Logs\errors.log" fullname=true />
      </filter>
   </outputs>
   <formats>
      <format id="fmterror" format="%Date %Time %LEVEL [%FuncShort @ %File.%Line] %Msg
%n" />
      <format id="fmtdebug" format="%Date %Time %LEVEL [%FuncShort @ %File.%Line] %Msg
%n" />
      <format id="fmtinfo" format="%Date %Time %LEVEL %Msg%n" />
   </formats>
</seelog>
```

Unable to connect to endpoint

O SSM Agent deve permitir o tráfego de saída HTTPS (porta 443) para os seguintes endpoints:

- `ssm.`*`region`*`.amazonaws.com`
- `ssmmessages.`*`region`*`.amazonaws.com`

A *`região`* representa o identificador da região para uma região da Região da AWS compatível com o AWS Systems Manager, como `us-east-2` para a região Leste dos EUA (Ohio). Para ver uma lista dos valores de *`região`* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

> **Note**
>
> Antes de 2024, `ec2messages.`*`region`*`.amazonaws.com` também era necessário. Para Regiões da AWS lançadas antes de 2024, permitir tráfego para `ssmmessages.`*`region`*`.amazonaws.com` ainda é obrigatório, mas opcional para `ec2messages.`*`region`*`.amazonaws.com`.

Para regiões lançadas em 2024 e posteriormente, permitir tráfego para `ssmmessages.region.amazonaws.com` é obrigatório, mas os endpoints `ec2messages.region.amazonaws.com` não são compatíveis com essas regiões.

O SSM Agent não funcionará se não puder se comunicar com os endpoints anteriores, conforme descrito, mesmo se você usar as Amazon Machine Images (AMIs) fornecidas pela AWS, como Amazon Linux 2 ou Amazon Linux 2023. Sua configuração de rede deve ter acesso aberto à Internet ou você deve ter endpoints personalizados de nuvem virtual privada (VPC) configurados. Se você não planeja criar um endpoint da VPC personalizado, verifique seus gateways de Internet ou gateways NAT. Para obter mais informações sobre endpoints da VPC gerenciados por Redshift, consulte Melhorar a segurança das instâncias do EC2 usando endpoints da VPC para o Systems Manager.

Verificar sua configuração de VPC

Se você estiver usando uma nuvem privada virtual (VPC), para gerenciar instâncias do EC2 com o Systems Manager, os endpoints da VPC devem ser configurados adequadamente para `ssm.region.amazonaws.com`, `ssmmessages.region.amazonaws.com` e, em alguns casos explicados anteriormente neste tópico em Unable to connect to endpoint, para `ec2messages.region.amazonaws.com`.

Note

A alternativa ao uso de um endpoint da VPC é permitir o acesso à Internet de saída em suas instâncias gerenciadas. Nesse caso, as instâncias gerenciadas também devem permitir tráfego de saída HTTPS (porta 443) para os seguintes endpoints:

- `ssm.region.amazonaws.com`
- `ssmmessages.region.amazonaws.com`
- `ec2messages.region.amazonaws.com`

O SSM Agent inicia todas as conexões com o serviço Systems Manager na nuvem. Por essa razão, você não precisa configurar o firewall para permitir o tráfego de entrada nas instâncias para o Systems Manager.

Para obter mais informações sobre chamadas para esses endpoints, consulte Referência: ec2messages, ssmmessages e outras operações da API.

Para solucionar problemas com seus endpoints da VPC, faça o seguinte:

- Garanta que endpoints da VPC sejam incluídos no nível da VPC. Se o endpoint da VPC com um nome de serviço específico não for encontrado na VPC, primeiro verifique se o suporte a DNS está habilitado no nível da VPC. Em seguida, crie um novo endpoint da VPC e associe-o a uma sub-rede em cada zona de disponibilidade.
- Certifique-se de que um nome DNS privado esteja habilitado no nível do endpoint da VPC. Os nomes de DNS privados são habilitados por padrão, mas podem ter sido desabilitados manualmente em algum momento.
- Certifique-se de que os endpoints da VPC existentes estejam associados à sub-rede adequada. Além disso, certifique-se de que o VPCE já esteja associado a uma sub-rede nessa zona de disponibilidade.

Para obter mais informações, consulte os tópicos a seguir.

- Acessar um AWS service (Serviço da AWS) usando um endpoint da VPC de interface no Guia do AWS PrivateLink.
- Associar um nome de DNS privado no Guia do AWS PrivateLink
- Melhorar a segurança das instâncias do EC2 usando endpoints da VPC para o Systems Manager

Verificar seus atributos relacionados a DNS da VPC

Se você estiver usando uma nuvem privada virtual (VPC), como parte da verificação da configuração da VPC, verifique se os atributos `enableDnsSupport` e `enableDnsHostnames` estão habilitados.

Você pode habilitar esses atributos usando a ação da API ModifyVpcAttribute do Amazon EC2 ou o comando modify-vpc-attribute da AWS CLI.

Para obter mais informações sobre como habilitar esses atributos no console da Amazon VPC, consulte Visualizar e atualizar atributos de DNS para sua VPC no Guai do usuário da Amazon VPC.

> **Note**
>
> A alternativa ao uso de um endpoint da VPC é permitir o acesso à Internet de saída em suas instâncias gerenciadas. Nesse caso, as instâncias gerenciadas também devem permitir tráfego de saída HTTPS (porta 443) para os seguintes endpoints:
>
> - `ssm.region.amazonaws.com`
> - `ssmmessages.region.amazonaws.com`
> - `ec2messages.region.amazonaws.com`
>
> O SSM Agent inicia todas as conexões com o serviço Systems Manager na nuvem. Por essa razão, você não precisa configurar o firewall para permitir o tráfego de entrada nas instâncias para o Systems Manager.
> Para obter mais informações sobre chamadas para esses endpoints, consulte Referência: ec2messages, ssmmessages e outras operações da API.

Verificar regras de entrada e saída em grupos de segurança de endpoints

Certifique-se de que todos os endpoints da VPC que você configurou (`ssm`, `ssmmessages` e `ec2messages`) incluam uma regra de entrada em seus grupos de segurança para permitir tráfego na porta 443. Se necessário, você poderá criar um novo grupo de segurança na VPC com uma regra de entrada para permitir o tráfego na porta 443 para o bloco Encaminhamento Entre Domínios Sem Classificação (CIDR) da VPC. Depois de criar o grupo de segurança, anexe-o a cada endpoint da VPC.

Para obter mais informações, consulte os tópicos a seguir.

- Como faço para criar endpoints da VPC para poder usar o Systems Manager para gerenciar instâncias do EC2 privadas sem acesso à Internet? no AWS re:Post
- Blocos CIDR de VPC no Guia do usuário da Amazon VPC

Usa a `ssm-cli` para solucionar problemas de disponibilidade do nó gerenciado

A partir do SSM Agent versão 3.1.501.0, você pode usar a `ssm-cli` para determinar se um nó gerenciado atende aos requisitos principais para ser gerenciado pelo Systems Manager e para ser exibido nas listas de nós gerenciados no Fleet Manager. O `ssm-cli` é uma ferramenta de linha de comando autônoma incluída na instalação do SSM Agent. Comandos pré-configurados estão incluídos para coletar as informações necessárias que ajudam a identificar por que uma instância do Amazon EC2 ou uma máquina que não é do EC2 que você confirmou que está em execução não está incluída nas listas de nós gerenciados no Systems Manager. Esses comandos são executados quando você especifica a opção `get-diagnostics`.

Para obter mais informações, consulte Solucionar problemas de disponibilidade do nó gerenciado usando a `ssm-cli`.

Segurança no AWS Systems Manager

A segurança da nuvem na Amazon Web Services é a nossa maior prioridade. Como cliente da AWS, você se beneficiará de um data center e de uma arquitetura de rede criados para atender aos requisitos das empresas com as maiores exigências de segurança.

A segurança é uma responsabilidade compartilhada entre a AWS e você. O modelo de responsabilidade compartilhada descreve isso como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem: a AWS é responsável pela proteção da infraestrutura que executa os Serviços da AWS na Nuvem AWS. A AWS também fornece serviços que você pode usar com segurança. Auditores de terceiros testam e verificam regularmente a eficácia da nossa segurança como parte dos Programas de conformidade da AWS. Para saber mais sobre os programas de conformidade que se aplicam ao AWS Systems Manager, consulte Serviços da AWS em escopo por programa de conformidade
- Segurança na nuvem: sua responsabilidade é determinada pelo AWS service (Serviço da AWS) que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade dos dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda a entender como aplicar o modelo de responsabilidade compartilhada ao usar o AWS Systems Manager. Os tópicos a seguir mostram como configurar o Systems Manager para atender aos seus objetivos de segurança e conformidade. Saiba também como usar outros Serviços da AWS que ajudam você a monitorar e proteger os recursos do Systems Manager.

Tópicos

- Proteção de dados no AWS Systems Manager
- Gerenciamento de identidade e acesso para o AWS Systems Manager
- Usar perfis vinculados a serviço do Systems Manager
- Registrar em log e monitorar no AWS Systems Manager
- Validação de conformidade do AWS Systems Manager
- Resiliência no AWS Systems Manager
- Segurança da infraestrutura no AWS Systems Manager
- Análise de vulnerabilidade e configuração no AWS Systems Manager
- Melhores práticas de segurança do Systems Manager

Proteção de dados no AWS Systems Manager

A proteção de dados protege os dados em trânsito (à medida que são transferidos no Systems Manager) e em repouso (enquanto estão armazenados em datacenters da AWS).

O AWS modelo de responsabilidade compartilhada se aplica à proteção de dados no AWS Systems Manager. Conforme descrito nesse modelo, AWS é responsável por proteger a infraestrutura global que executa todas as Nuvem AWS. Você é responsável por manter o controle sobre o conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre a privacidade de dados, consulte as Data Privacy FAQ. Para obter mais informações sobre a proteção de dados na Europa, consulte a postagem do blog AWS Shared Responsibility Model and RGPD no Blog de segurança da AWS.

Para fins de proteção de dados, recomendamos que você proteja as credenciais da Conta da AWS e configure as contas de usuário individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com os recursos da AWS. Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure os logs de API e atividade do usuário com AWS CloudTrail. Para obter informações sobre como usar as trilhas do CloudTrail para capturar atividades da AWS, consulte Working with CloudTrail trails no Guia do usuário do AWS CloudTrail.
- Use as soluções de criptografia AWS, juntamente com todos os controles de segurança padrão em Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-3 ao acessar a AWS por meio de uma interface de linha de comandos ou de uma API, use um endpoint do FIPS. Para obter mais informações sobre os endpoints FIPS disponíveis, consulte Federal Information Processing Standard (FIPS) 140-3.

É altamente recomendável que nunca sejam colocadas informações confidenciais ou sigilosas, como endereços de e-mail de clientes, em tags ou campos de formato livre, como um campo Nome. Isso inclui trabalhar com a Systems Manager ou outros Serviços da AWS usando o console, a API, a AWS CLI ou os AWS SDKs. Quaisquer dados inseridos em tags ou em campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, é fortemente recomendável que não sejam incluídas informações de credenciais no URL para validar a solicitação nesse servidor.

Criptografia de dados

Criptografia em repouso

Parâmetros do Parameter Store

Os tipos de parâmetros que é possível criar no Parameter Store, uma ferramenta do AWS Systems Manager, incluem `String`, `StringList` e `SecureString`.

Todos os parâmetros, independentemente do tipo, são criptografados em trânsito e em repouso. Em trânsito, os parâmetros são criptografados usando a transport layer security (TLS) para criar uma conexão HTTPS segura para solicitações de API. Em repouso, eles são criptografados com um Chave pertencente à AWS no AWS Key Management Service (AWS KMS). Para obter mais informações sobre criptografia Chave pertencente à AWS, consulte Chaves pertencentes à AWS no Guia do desenvolvedor AWS Key Management Service .

O tipo de `SecureString` oferece opções adicionais de criptografia e é recomendado para todos os dados confidenciais. Você pode escolher entre os seguintes tipos de chaves AWS KMS para criptografar e descriptografar o valor de um parâmetro de `SecureString`:

- A Chave gerenciada pela AWS da sua conta
- Uma chave gerenciada pelo cliente (CMK) que você criou em sua conta
- Uma CMK em outra Conta da AWS que foi compartilhada com você

Para obter mais informações sobre criptografiaAWS KMS, consulte o Guia do desenvolvedor AWS Key Management Service.

Conteúdo em buckets do S3

Como parte de suas operações do Systems Manager, é possível optar por carregar ou armazenar dados em um ou mais buckets do Amazon Simple Storage Service (Amazon S3).

Para obter mais informações sobre a criptografia do bucket do S3, consulte Proteger dados usando criptografia e Proteção de dados no Amazon S3 no Guia do usuário do Amazon Simple Storage Service.

Veja a seguir os tipos de dados dos quais é possível fazer upload ou ter armazenados em buckets do S3 como parte de suas atividades do Systems Manager:

- A saída dos comandos em Run Command, uma ferramenta do AWS Systems Manager
- Pacotes no Distributor, uma ferramenta do AWS Systems Manager
- Logs de operação de aplicação de patches no Patch Manager, uma ferramenta do AWS Systems Manager
- Listas de substituição de patches do Patch Manager
- Scripts ou manuais do Ansible a serem executados em um fluxo de trabalho do runbook no Automation, uma ferramenta do AWS Systems Manager
- Perfis do Chef InSpec para uso com as verificações no Compliance, uma ferramenta do AWS Systems Manager
- Logs do AWS CloudTrail
- Logs de histórico de sessão no Session Manager, uma ferramenta do AWS Systems Manager
- Relatórios do Explorer, uma ferramenta do AWS Systems Manager
- OpsData do OpsCenter, uma ferramenta do AWS Systems Manager
- Modelos do AWS CloudFormation para uso com fluxos de trabalho de automação
- Dados de conformidade de uma verificação de sincronização de dados de recursos
- Saída de solicitações para criar ou editar associação no State Manager, uma ferramenta do AWS Systems Manager, em nós gerenciados
- Documentos personalizados do Systems Manager (documentos SSM) que você pode executar usando o documento do SSM gerenciado pela AWS do `AWS-RunDocument`

Grupos de logs do CloudWatch Logs

Como parte de suas operações do Systems Manager, é possível optar por transmitir dados para um ou mais grupos de logs do Amazon CloudWatch Logs.

Para obter informações sobre a criptografia de grupos de logs do CloudWatch Logs, consulte Criptografar dados de log no CloudWatch Logs usando o AWS Key Management Service no Guia do usuário do Amazon CloudWatch Logs.

Veja a seguir tipos de dados que é possível ter transmitido para um grupo de logs do CloudWatch Logs como parte de suas atividades do Systems Manager:

- A saída de comandos do Run Command
- A saída de scripts executados usando a ação `aws:executeScript` em um runbook de automação
- Logs de histórico de sessão do Session Manager
- Logs do SSM Agent em nós gerenciados

Criptografia em trânsito

Recomendamos usar um protocolo de criptografia, como o Transport Layer Security (TLS), para criptografar dados sigilosos em trânsito entre os clientes e seus nós.

O Systems Manager fornece o suporte a seguir para criptografia de seus dados em trânsito.

Conexões com endpoints da API do Systems Manager

Os endpoints da API do Systems Manager oferecem suporte a conexões seguras somente em HTTPS. Ao gerenciar recursos do Systems Manager com o AWS Management Console, o AWS SDK ou a API do Systems Manager, toda a comunicação é criptografada com Transport Layer Security (TLS). Para obter uma lista completa de endpoints de API, consulte AWS service (Serviço da AWS) endpoints no Referência geral da Amazon Web Services.

Instâncias gerenciadas

A AWS fornece conectividade segura e privada entre as instâncias do Amazon Elastic Compute Cloud (Amazon EC2). Além disso, nós criptografamos automaticamente os dados em trânsito entre instâncias compatíveis na mesma rede privada virtual (VPC) ou em VPCs emparelhadas, usando algoritmos AEAD com criptografia de 256 bits. Essa funcionalidade de criptografia usa recursos de descarregamento do hardware subjacente, e não há impacto na performance da rede. As instâncias compatíveis são: C5n, G4, I3en, M5dn, M5n, P3dn, R5dn e R5n.

Sessões do Session Manager

Por padrão, o Session Manager usa o TLS 1.3 para criptografar dados de sessão transmitidos entre as máquinas locais de usuários na sua conta e suas instâncias do EC2. Você também pode escolher criptografar ainda mais os dados em trânsito usando uma AWS KMS key criada no AWS KMS. A criptografia do AWS KMS está disponível para os tipos de sessão `Standard_Stream`, `InteractiveCommands` e `NonInteractiveCommands`.

Acesso do Run Command

Por padrão, o acesso remoto aos nós que usam o Run Command é criptografado usando TLS 1.3, e as solicitações para criar uma conexão são assinadas usando SigV4.

Privacidade do tráfego entre redes

Você pode usar o Amazon Virtual Private Cloud (Amazon VPC) para criar limites entre os recursos dos nós gerenciados e controlar o tráfego entre eles, sua rede on-premises e a Internet. Para obter detalhes, consulte Melhorar a segurança das instâncias do EC2 usando endpoints da VPC para o Systems Manager.

Para obter mais informações sobre a segurança da Amazon Virtual Private Cloud, consulte Internetwork traffic privacy in Amazon VPC (Tráfego de trabalho na Internet na Amazon VPC) no Guia do usuário da Amazon VPC.

Gerenciamento de identidade e acesso para o AWS Systems Manager

AWS Identity and Access Management (IAM) é um serviço da AWS service (Serviço da AWS) que ajuda o administrador no controle de segurança de acesso aos recursos da AWS de forma segura. Os administradores do IAM controlam quem pode ser autenticado (fazer login) e autorizado (ter permissões) a usar os recursos do Systems Manager. O IAM é um AWS service (Serviço da AWS) que pode ser usado sem custo adicional.

Tópicos

- Público
- Como autenticar com identidades
- Gerenciar o acesso usando políticas
- Como o AWS Systems Manager funciona com o IAM
- Exemplos de políticas baseadas em identidade do AWS Systems Manager
- Políticas gerenciadas pela AWS do AWS Systems Manager
- Solução de problemas de identidade e acesso do AWS Systems Manager

Público

O uso do AWS Identity and Access Management (IAM) varia dependendo do trabalho que for realizado no Systems Manager.

Usuário do serviço: se você usar o serviço Systems Manager para fazer o trabalho, o administrador fornecerá as credenciais e as permissões necessárias. À medida que usar mais recursos do Systems Manager para fazer seu trabalho, você poderá precisar de permissões adicionais. Entender como o acesso é gerenciado pode ajudá-lo a solicitar as permissões corretas ao seu administrador. Se não for possível acessar um recurso no Systems Manager, consulte Solução de problemas de identidade e acesso do AWS Systems Manager.

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

Administrador do IAM: se você for um administrador do IAM, talvez queira saber detalhes sobre como pode gravar políticas para gerenciar o acesso ao Systems Manager. Para visualizar exemplos Systems Manager de políticas baseadas em identidade do que podem ser usadas no IAM, consulte Exemplos de políticas baseadas em identidade do AWS Systems Manager.

Como autenticar com identidades

A autenticação é a forma como fazer login na AWS usando suas credenciais de identidade. É necessário ser autenticado (fazer login na AWS) como Usuário raiz da conta da AWS, como usuário do IAM, ou assumindo um perfil do IAM.

É possível fazer login na AWS como uma identidade federada usando credenciais fornecidas por uma fonte de identidades. AWS IAM Identity Center (Centro de Identidade do IAM), a autenticação única da empresa e as suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você faz login como identidade federada, o administrador já configurou anteriormente a federação de identidades usando perfis do IAM. Quando você acessa a AWS usando a federação, está indiretamente assumindo um perfil.

A depender do tipo de usuário, você pode fazer login no AWS Management Console ou no portal de acesso AWS. Para obter mais informações sobre como fazer login na AWS, consulte Como fazer login na contaConta da AWS no Início de Sessão da AWS Guia do usuário.

Se você acessar a AWS de forma programática, a AWS fornecerá um kit de desenvolvimento de software (SDK) e uma interface de linha de comandos (CLI) para você assinar de forma criptográfica as solicitações usando as suas credenciais. Se você não utilizar as ferramentas da AWS, deverá assinar as solicitações por conta própria. Para obter mais informações sobre como usar o método recomendado para designar solicitações por conta própria, consulte Versão 4 do AWS Signature para solicitações de API no Guia do usuário do IAM.

Independente do método de autenticação usado, também pode ser necessário fornecer informações adicionais de segurança. Por exemplo, a AWS recomenda o uso da autenticação multifator (MFA) para aumentar a segurança de sua conta. Para saber mais, consulte Autenticação multifator no Guia do usuário do AWS IAM Identity Center e Usar a autenticação multifator da AWS no IAM no Guia do usuário do IAM.

Usuário-raiz Conta da AWS

Ao criar uma Conta da AWS, você começa com uma identidade de login com acesso completo a todos os Serviços da AWS e recursos na conta. Essa identidade, chamada usuário-raiz da Conta da AWS, é acessada por login com o endereço de e-mail e a senha usada para criar a conta. É altamente recomendável não usar o usuário-raiz para tarefas diárias. Proteja as credenciais do usuário-raiz e use-as para executar as tarefas que somente ele puder executar. Para obter a lista completa das tarefas que exigem login como usuário-raiz, consulte Tarefas que exigem credenciais de usuário-raiz no Guia do Usuário do IAM.

Usuários e grupos do IAM

Um usuário do IAM é uma identidade dentro da Conta da AWS que tem permissões específicas para uma única pessoa ou aplicação. Sempre que possível, é recomendável contar com credenciais temporárias em vez de criar usuários do IAM com credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com usuários do IAM, é recomendável alternar as chaves de acesso. Para obter mais informações, consulte Alternar as chaves de acesso regularmente para casos de uso que exijam credenciais de longo prazo no Guia do Usuário do IAM.

Um grupo do IAM é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários

de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdmins e conceder a esse grupo permissões para administrar recursos do IAM.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte Casos de uso para usuários do IAM no Guia do usuário do IAM.

Perfis do IAM

Um perfil do IAM é uma identidade na Conta da AWS que tem permissões específicas. Ele é semelhante a um usuário do IAM, mas não está associado a uma pessoa específica. Para assumir temporariamente um perfil do IAM no AWS Management Console, você pode alternar de um usuário para um perfil do IAM (console). É possível presumir um perfil chamando uma operação de API da AWS CLI ou da AWS, ou usando um URL personalizado. Para obter mais informações sobre métodos para usar perfis, consulte Métodos para assumir um perfil no Guia do usuário do IAM.

Perfis do IAM com credenciais temporárias são úteis nas seguintes situações:

- Acesso de usuário federado: para atribuir permissões a identidades federadas, é possível criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas por ele. Para ter mais informações sobre perfis para federação, consulte Criar um perfil para um provedor de identidade de terceiros (federação) no Guia do usuário do IAM. Se usar o Centro de Identidade do IAM, configure um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o Centro de Identidade do IAM correlaciona o conjunto de permissões a um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte Conjuntos de Permissões no Guia do Usuário do AWS IAM Identity Center.
- Permissões temporárias para usuários do IAM: um usuário ou um perfil do IAM pode presumir um perfil do IAM para obter temporariamente permissões diferentes para uma tarefa específica.
- Acesso entre contas: é possível usar um perfil do IAM para permitir que alguém (uma entidade principal confiável) em outra conta acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, alguns Serviços da AWS permitem anexar uma política diretamente a um recurso (em vez de usar um perfil como proxy). Para conhecer a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

- Acesso entre serviços: alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicações no Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões da entidade principal da chamada, usando um perfil de serviço ou um perfil vinculado ao serviço.
  - Sessões de acesso direto (FAS): qualquer pessoa que utilizar um perfil ou usuário do IAM para executar ações na AWS é considerada uma entidade principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O recurso FAS usa as permissões da entidade principal chamando um AWS service (Serviço da AWS), bem como o AWS service (Serviço da AWS) solicitante, para fazer solicitações para serviços subsequentes. As solicitações de FAS são feitas somente quando um serviço recebe uma solicitação que exige interações com outros Serviços da AWS ou com recursos para serem concluídas. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Sessões de acesso direto.
  - Perfil de serviço: um perfil de serviço é um perfil do IAM que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do Usuário do IAM.
  - Perfil vinculado a serviço: um perfil vinculado a serviço é um tipo de perfil de serviço vinculado a um AWS service (Serviço da AWS). O serviço pode presumir o perfil para executar uma ação em seu nome. Perfis vinculadas ao serviço aparecem em sua Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para perfis vinculados a serviço.
- Aplicações em execução no Amazon EC2: é possível usar um perfil do IAM para gerenciar credenciais temporárias para aplicações em execução em uma instância do EC2 e fazer solicitações da AWS CLI ou da API da AWS. É preferível fazer isso a armazenar chaves de acesso na instância do EC2. Para atribuir um perfil da AWS a uma instância do EC2 e disponibilizá-la para todas as suas aplicações, crie um perfil de instância que esteja anexado a ela. Um perfil de instância contém o perfil e permite que os programas em execução na instância do EC2 obtenham credenciais temporárias. Para mais informações, consulte Utilizar um perfil do IAM para conceder permissões a aplicações em execução nas instâncias do Amazon EC2 no Guia do usuário do IAM.

Gerenciar o acesso usando políticas

Você controla o acesso na AWS criando políticas e anexando-as a identidades ou recursos da AWS. Uma política é um objeto na AWS que, quando associado a uma identidade ou recurso, define suas permissões. A AWS avalia essas políticas quando uma entidade principal (usuário, usuário-raiz ou sessão de perfil) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada na AWS como documentos JSON. Para obter mais informações sobre a estrutura e o conteúdo de documentos de políticas JSON, consulte Visão geral das políticas JSON no Guia do usuário do IAM.

Os administradores podem usar as políticas JSON da AWS para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e perfis não têm permissões. Para conceder permissão aos usuários para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis e os usuários podem assumir os perfis.

As políticas do IAM definem permissões para uma ação independentemente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de perfis do AWS Management Console, da AWS CLI ou da API AWS.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

As políticas baseadas em identidade podem ser categorizadas como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas independentes que podem ser anexadas a vários usuários, grupos e perfis na Conta da AWS. As políticas gerenciadas incluem políticas gerenciadas pela AWS e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM.

Para obter informações sobre as políticas gerenciadas pela AWS para o Systems Manager, consulte Políticas gerenciadas AWS Systems Manager.

Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. As entidades principais podem incluir contas, usuários, funções, usuários federados ou Serviços da AWS.

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Não é possível usar as políticas gerenciadas pela AWS do IAM em uma política baseada em atributos.

Chaves de condição de políticas

As ações que os usuários e perfis podem realizar e os recursos nos quais eles podem realizar essas ações podem ser ainda mais restringidos por condições específicas.

Em documentos de política JSON, o elemento `Condition` (ou bloco `Condition`) permite que você especifique condições nas quais uma declaração está em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usam operadores de condição, como `StringEquals` ou `StringNotLike`, para comparar a condição na política com os valores na solicitação.

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

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

A AWS oferece compatibilidade com chaves de condição globais e chaves de condição específicas do serviço. Para obter mais informações, consulte Chaves de contexto de condição global da AWS no Guia do usuário do IAM.

> **Important**
>
> Se você usa o Systems Manager Automation, recomendamos que você não use a chave de condição aws:SourceIp em suas políticas. O comportamento dessa chave de condição depende de vários fatores, incluindo se um perfil do IAM para a execução de um runbook do Automation é fornecido e as ações de automação usadas no runbook. Como resultado, a chave de condição pode produzir um comportamento inesperado. Por esse motivo, recomendamos que você não a use.

O Systems Manager oferece suporte a várias de suas próprias chaves de condição. Para obter mais informações, consulte Chaves de condição do AWS Systems Manager na Referência de autorização do serviço. As ações e os recursos com os quais você pode usar uma chave de condição específica do Systems Manager estão listados em Tipos de recursos definidos por AWS Systems Manager na Referência de autorização do serviço.

Se sua política precisar depender de um nome de entidade principal de serviço pertencente ao serviço Systems Manager, recomendamos verificar sua existência ou inexistência usando a chave de condição com vários valores `aws:PrincipalServiceNamesList` em vez da chave de condição `aws:PrincipalServiceName`. A chave de condição `aws:PrincipalServiceName` contém somente uma entrada da lista de nomes de entidades principais do serviço e pode nem sempre ser o nome da entidade principal do serviço que você espera. O bloco `Condition` a seguir demonstra a verificação da existência de `ssm.amazonaws.com`.

```
{
    "Condition": {
        "ForAnyValue:StringEquals": {
            "aws:PrincipalServiceNamesList": "ssm.amazonaws.com"
        }
    }
}
```

Para visualizar exemplos de políticas baseadas em identidade do Systems Manager, consulte Exemplos de políticas baseadas em identidade do AWS Systems Manager.

Listas de controle de acesso (ACLs)

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

Amazon S3, AWS WAF e Amazon VPC são exemplos de serviços que oferecem compatibilidade com ACLs. Para saber mais sobre ACLs, consulte Visão geral da lista de controle de acesso (ACL) no Guia do Desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

A AWS oferece compatibilidade com tipos de política menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- Limites de permissões: um limite de permissões é um recurso avançado no qual você define o máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM (usuário ou perfil do IAM). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade com seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs): SCPs são políticas JSON que especificam as permissões máximas para uma organização ou unidade organizacional (UO) no AWS Organizations. O AWS Organizations é um serviço que agrupa e gerencia centralmente várias Contas da AWS pertencentes a sua empresa. Se você habilitar todos os recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as contas. A SCP limita as permissões para entidades em contas de membros, o que inclui cada Usuário raiz da conta da AWS. Para obter mais informações sobre o Organizations e SCPs, consulte Service control policies no Guia do usuário do AWS Organizations.
- Políticas de controle de recursos (RCPs): RCPs são políticas JSON que podem ser usadas para definir o máximo de permissões disponíveis para recursos em suas contas sem atualizar as políticas do IAM anexadas a cada recurso que você possui. A RCP limita as permissões para recursos nas contas-membro e pode afetar as permissões efetivas para identidades, incluindo o Usuário raiz da conta da AWS, independentemente de pertencerem a sua organização. Consulte mais informações sobre o Organizations e as RCPs, incluindo uma lista de Serviços da

AWS compatível com RCPs em Resource control policies (RCPs) no Guia do usuário do AWS Organizations.

- Políticas de sessão: são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em recursos. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações, consulte Políticas de sessão no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como a AWS determina se deve permitir ou não uma solicitação quando há vários tipos de política envolvidos, consulte Lógica da avaliação de políticas no Guia do usuário do IAM.

Como o AWS Systems Manager funciona com o IAM

Antes de usar o AWS Identity and Access Management IAM para gerenciar o acesso ao AWS Systems Manager, você precisa saber quais recursos do IAM estão disponíveis para uso com o Systems Manager. Para ter uma visão geral de como o Systems Manager e outros Serviços da AWS funcionam com o IAM, consulte Serviços da AWS compatíveis com o IAM, no Guia do usuário do IAM.

Tópicos

- Políticas baseadas em identidade Systems Manager
- Políticas baseadas em recursos do Systems Manager
- Autorização baseada em tags do Systems Manager
- Perfis do IAM no Systems Manager

Políticas baseadas em identidade Systems Manager

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados e as condições sob as quais as ações são permitidas ou negadas. O Systems Manager oferece suporte a ações, recursos e chaves de condição específicos. Para saber mais sobre todos

os elementos usados em uma política JSON, consulte Referência de elementos de política JSON do IAM no Guia do usuário do IAM.

Ações

Os administradores podem usar as políticas JSON da AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Action` de uma política JSON descreve as ações que podem ser usadas para permitir ou negar acesso em uma política. As ações de políticas geralmente têm o mesmo nome que a operação de API da AWS associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Algumas operações também exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

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

As ações de políticas no Systems Manager usam o seguinte prefixo antes da ação: `ssm:`. Por exemplo, para conceder a alguém permissão para criar um parâmetro do Systems Manager (parâmetro do SSM) com a operação de API `PutParameter` do Systems Manager, inclua a ação `ssm:PutParameter` na política da pessoa. As declarações de política devem incluir um elemento `Action` ou Systems Manager. O `NotAction` define seu próprio conjunto de ações que descrevem as tarefas que podem ser executadas com esse serviço.

Para especificar várias ações em uma única instrução, separe-as com vírgulas, como segue:

```
"Action": [
      "ssm:action1",
      "ssm:action2"
]
```

> **Note**
>
> As ferramentas do AWS Systems Manager a seguir usam prefixos diferentes antes das ações.
>
> - O AWS AppConfig usa o prefixo `appconfig:` antes das ações.
> - O Incident Manager usa o prefixo `ssm-incidents:` ou `ssm-contacts:` antes das ações.
> - O Systems Manager GUI Connect usa o prefixo `ssm-guiconnect:` antes das ações.

- O Quick Setup usa o prefixo `ssm-quicksetup:` antes das ações.

Você também pode especificar várias ações usando caracteres-curinga (*). Por exemplo, para especificar todas as ações que começam com a palavra `Describe`, inclua a seguinte ação:

```
"Action": "ssm:Describe*"
```

Para ver uma lista das ações do Systems Manager, consulte Ações definidas pelo AWS Systems Manager na Referência de autorização do serviço.

Recursos

Os administradores podem usar as políticas JSON da AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

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

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

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

Por exemplo, o recurso da janela de manutenção do Systems Manager tem o seguinte formato ARN.

```
arn:aws:ssm:region:account-id:maintenancewindow/window-id
```

Para especificar as janelas de manutenção mw-0c50858d01EXAMPLE na instrução na região Leste dos EUA (Ohio), você usaria um ARN semelhante ao que se segue.

```
"Resource": "arn:aws:ssm:us-east-2:123456789012:maintenancewindow/mw-0c50858d01EXAMPLE"
```

Para especificar todas as janelas de manutenção que pertencem a uma conta específica, use o curinga (*).

```
"Resource": "arn:aws:ssm:region:123456789012:maintenancewindow/*"
```

Para ações de API do `Parameter Store`, é possível fornecer ou restringir o acesso a todos parâmetros em um nível da hierarquia usando nomes hierárquicos e políticas do AWS Identity and Access Management (IAM), da seguinte maneira:

```
"Resource": "arn:aws:ssm:region:123456789012:parameter/Dev/ERP/Oracle/*"
```

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

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

Algumas operações de API do Systems Manager aceitam vários recursos. Para especificar vários recursos em uma única instrução, separe seus ARNs com vírgulas, conforme o seguinte.

```
"Resource": [
      "resource1",
      "resource2"
```

> **Note**
>
> A maioria dos Serviços da AWS trata dois pontos (:) e a barra inclinada (/) como o mesmo caractere em ARNs. No entanto, o Systems Manager requer uma correspondência exata nos padrões e regras de recursos. Ao criar padrões de eventos, lembre-se de usar os caracteres corretos do ARN para que correspondam ao ARN do recurso.

A tabela a seguir descreve os formatos de ARN para os tipos de recursos aceitos pelo Systems Manager.

> **Note**
>
> Observe as seguintes exceções aos formatos de ARN.
>
> - As ferramentas do AWS Systems Manager a seguir usam prefixos diferentes antes das ações.
>   - O AWS AppConfig usa o prefixo `appconfig:` antes das ações.

  - O Incident Manager usa o prefixo `ssm-incidents:` ou `ssm-contacts:` antes das ações.
  - O Systems Manager GUI Connect usa o prefixo `ssm-guiconnect` antes das ações.
- Documentos e recursos de definição de automação que pertencem à Amazon, bem como parâmetros públicos fornecidos pela Amazon e por fontes externas. Não incluem IDs de conta em seus formatos de ARN. Por exemplo:
  - O documento do SSM `AWS-RunPatchBaseline`:

    `arn:aws:ssm:us-east-2::document/AWS-RunPatchBaseline`
  - O runbooks de automação `AWS-ConfigureMaintenanceWindows`:

    `arn:aws:ssm:us-east-2::automation-definition/AWS-ConfigureMaintenanceWindows`
  - O parâmetro público `/aws/service/bottlerocket/aws-ecs-1-nvidia/x86_64/1.13.4/image_version`:

    `arn:aws:ssm:us-east-2::parameter/aws/service/bottlerocket/aws-ecs-1-nvidia/x86_64/1.13.4/image_version`

  Para obter mais informações sobre esses três tipos de recursos consulte os seguintes tópicos:
  - Trabalhar com documentos
  - Executar uma operação automatizada com tecnologia do Systems Manager Automation
  - Trabalhar com parâmetros públicos no Parameter Store
- O Quick Setup usa o prefixo `ssm-quicksetup:` antes das ações.

Tipo de recurso	Formato ARN
Aplicação (AWS AppConfig)	arn:aws:appconfig:*region*:*account-id* :application/*application-id*
Associação	arn:aws:ssm:*region*:*account-id* :association/*association-id*
Execução de automação	arn:aws:ssm:*region*:*account-id* :automation-execution/*automation-execution-id*

Tipo de recurso	Formato ARN
Definição de automação (com sub-recurso da versão)	arn:aws:ssm:*region*:*account-id* :automation-definition/*automation-definition-id* :*version-id* [1]
Perfil de configuração (AWS AppConfig)	arn:aws:appconfig:*region*:*account-id* :application/*application-id* /configurationprofile/*configurationprofile-id*
Contatos (Incident Manager)	arn:aws:ssm-contacts:*region*:*account-id* :contact/*contact-alias*
Estratégia de implantação (AWS AppConfig)	arn:aws:appconfig:*region*:*account-id* :deploymentstrategy/*deploymentstrategy-id*
Documento	arn:aws:ssm:*region*:*account-id* :document/*document-name*
Ambiente (AWS AppConfig)	arn:aws:appconfig:*region*:*account-id* :application/*application-id* /environment/*environment-id*
O incidente	arn:aws:ssm-incidents:*region*:*account-id* :incident-record/*response-plan-name* /*incident-id*
Janela de manutenção	arn:aws:ssm:*region*:*account-id* :maintenancewindow/*window-id*
Nó gerenciado	arn:aws:ssm:*region*:*account-id* :managed-instance/*managed-node-id*
Inventário de nós gerenciados	arn:aws:ssm:*region*:*account-id* :managed-instance-inventory /*managed-node-id*
OpsItem	arn:aws:ssm:*region*:*account-id* :opsitem/*OpsItem-id*

Tipo de recurso	Formato ARN
Parameter	Parâmetro de um único nível: • arn:aws:ssm:*region*:*account-id* :parameter/*parameter-name*/ Um parâmetro com nome de uma estrutura hierárquica: • arn:aws:ssm:*region*:*account-id* :parameter/*parameter-name-root* /*level-2*/*level-3*/*level-4*/*level-5* [2]
Lista de referência de patches	arn:aws:ssm:*region*:*account-id* :patchbaseline/*patch-baseline-id*
Plano de resposta	arn:aws:ssm-incidents:*region*:*account-id* :response-plan/*response-plan-name*
Sessão	arn:aws:ssm:*region*:*account-id* :session/*session-id* [3]
Todos os recursos do Systems Manager	arn:aws:ssm:*
Todos os recursos do Systems Manager pertencentes à Conta da AWS especificada na Região da AWS especificada	arn:aws:ssm:*region*:*account-id* :*

[1]

Para definições de automação, o Systems Manager oferece suporte a um recurso de segundo nível, o ID de versão. Na AWS, esses recursos de segundo nível são conhecidos como sub-recursos. A especificação de um sub-recurso de uma versão para um recurso de definição de automação permite que você forneça acesso a determinadas versões de uma definição de automação. Por

exemplo, é provável que você queira garantir que apenas a versão mais recente de uma definição de automação seja usada no gerenciamento do nó.

2

Para organizar e gerenciar parâmetros, é possível criar nomes para parâmetros com uma estrutura hierárquica. Com construção hierárquica, um nome de parâmetro pode incluir um caminho definido usando-se barras. É possível atribuir um nome a um recurso de parâmetro com um máximo de quinze níveis. Sugerimos que você crie hierarquias que reflitam uma estrutura hierárquica existente no seu ambiente. Para ter mais informações, consulte Criar parâmetros do Parameter Store no Systems Manager.

3

Na maioria dos casos, o ID da sessão é criado usando o ID do usuário da conta que iniciou a sessão, além de um sufixo alfanumérico. Por exemplo:

```
arn:aws:us-east-2:111122223333:session/JohnDoe-1a2b3c4sEXAMPLE
```

No entanto, se o ID de usuário não estiver disponível, o ARN será construído dessa forma:

```
arn:aws:us-east-2:111122223333:session/session-1a2b3c4sEXAMPLE
```

Para obter mais informações sobre o formato de ARNs, consulte Nomes de recurso da Amazon (ARNs) na Referência geral da Amazon Web Services.

Para obter uma lista dos tipos de recursos do Systems Manager e seus ARNs, consulte Recursos definidos pelo AWS Systems Manager na Referência de autorização do serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte Ações definidas pelo AWS Systems Manager.

Chaves de condição do Systems Manager

Os administradores podem usar as políticas JSON da AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Condition` (ou bloco `Condition`) permite que você especifique condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usem agentes de condição, como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

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

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

A AWS oferece compatibilidade com chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as chaves de condição globais da AWS, consulte Chaves de Contexto de Condição Globais da AWS no Guia do Usuário do IAM.

Para ver uma lista de chaves de condição do Systems Manager, consulte Chaves de Condição do AWS Systems Manager no campo Referência de Autorização do Serviço. Para saber com quais ações e recursos você pode usar uma chave de condição, consulte Ações definidas pelo AWS Systems Manager.

Para obter informações sobre como usar a chave de condição `ssm:resourceTag/*`, consulte os tópicos a seguir:

- Restringir o acesso aos comandos em nível raiz por meio do SSM Agent
- Restringir o acesso ao Run Command com base em etiquetas
- Restringir o acesso à sessão com base em tags de instância

Para obter informações sobre como usar as chaves de condição `ssm:Recursive`, `ssm:Policies` e `ssm:Overwrite`, consulte Impedir o acesso às operações da API Parameter Store.

Exemplos

Para ver exemplos de políticas baseadas em identidade do Systems Manager, consulte Exemplos de políticas baseadas em identidade do AWS Systems Manager.

Políticas baseadas em recursos do Systems Manager

Outros Serviços da AWS, como o Amazon Simple Storage Service (Amazon S3), são compatíveis com políticas de permissões baseadas em recursos. Por exemplo: você pode anexar uma política de permissões a um bucket do S3 para gerenciar permissões de acesso a esse bucket.

O Systems Manager não é compatível com as políticas baseadas em recursos.

Autorização baseada em tags do Systems Manager

É possível anexar tags a recursos do Systems Manager ou passar tags em uma solicitação ao Systems Manager. Para controlar o acesso com base em tags, forneça informações sobre as tags no elemento de condição de uma política usando as chaves de condição `ssm:resourceTag/key-name`, `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`. É possível adicionar tags aos seguintes tipos de recursos ao criá-los ou atualizá-los:

- Documento
- Nó gerenciado
- Janela de manutenção
- Parameter
- Lista de referência de patches
- OpsItem

Para visualizar um exemplo de política baseada em identidade para limitar o acesso a um recurso baseado em tags desse recurso, consulte Visualizar documentos do Systems Manager com base em tags.

Perfis do IAM no Systems Manager

Um perfil do IAM é uma entidade dentro da sua Conta da AWS que tem permissões específicas.

Usar credenciais temporárias com o Systems Manager

É possível usar credenciais temporárias para fazer login com federação, assumir um perfil do IAM ou assumir um perfil entre contas. As credenciais de segurança temporárias são obtidas chamando operações da API do AWS Security Token Service (AWS STS), como AssumeRole ou GetFederationToken.

O Systems Manager oferece compatibilidade com o uso de credenciais temporárias.

Perfis vinculados ao serviço

Funções vinculadas ao serviço permitem que os Serviços da AWS acessem recursos em outros serviços para concluir uma ação em seu nome. As funções vinculadas ao serviço são listadas em sua conta do IAM e são de propriedade do serviço. Um administrador do pode visualizar, mas não pode editar as permissões para funções vinculadas ao serviço.

Systems Manager é compatível com funções vinculadas ao serviço. Para obter detalhes sobre como criar ou gerenciar perfis vinculados a serviço do Systems Manager, consulte Usar perfis vinculados a serviço do Systems Manager.

Perfis de serviço

Esse atributo permite que um serviço assuma um perfil de serviço em seu nome. O perfil permite que o serviço acesse recursos em outros serviços para concluir uma ação em seu nome. As funções de serviço são exibidas em sua conta do IAM e são de propriedade da conta. Isso significa que um administrador do pode alterar as permissões para essa função. Porém, fazer isso pode alterar a funcionalidade do serviço.

O Systems Manager oferece suporte às funções de serviço.

Selecionar uma função do IAM no Systems Manager

Para o Systems Manager interagir com os nós gerenciados, é necessário escolher uma função que permita que o Systems Manager acesse os nós em seu nome. Caso já tenha criado uma função de serviço ou função vinculada ao serviço, o Systems Manager fornecerá uma lista das funções para sua escolha. É importante escolher uma função que permita o acesso para iniciar e interromper nós gerenciados.

Para acessar as instâncias do EC2, é necessário configurar as permissões da instância. Para obter informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager.

Para acessar nós que não são do EC2 em um ambiente híbrido e multinuvem, o perfil necessário para sua Conta da AWS é um perfil de serviço do IAM. Para obter informações, consulte Criar o perfil de serviço do IAM obrigatório para o Systems Manager em ambientes híbridos e multinuvem.

Um fluxo de trabalho de automação pode ser iniciado no contexto de uma função de serviço (ou função de admissão). Isso permite que o serviço execute ações em seu nome. Se você não especificar uma função de admissão, a Automação usará o contexto do usuário que invocou a execução. No entanto, determinadas situações exigem especificar uma função de serviço para

automação. Para ter mais informações, consulte Configurar o acesso a uma função de serviço (função assumida) para automações.

Políticas gerenciadas AWS Systems Manager

A AWS aborda muitos casos de uso comuns fornecendo políticas autônomas do IAM que são criadas e administradas pela AWS. Essas políticas gerenciadas da AWS concedem as permissões necessárias para casos de uso comuns. Assim, você não precisa investigar quais permissões são necessárias. (Você também pode criar as próprias políticas do IAM personalizadas a fim de conceder permissões para ações e recursos do Systems Manager.)

Para obter mais informações sobre políticas gerenciadas para o Systems Manager, consulte Políticas gerenciadas pela AWS do AWS Systems Manager

Para obter informações gerais sobre políticas gerenciadas, consulte Políticas gerenciadas pela AWS no Guia do usuário do IAM.

Exemplos de políticas baseadas em identidade do AWS Systems Manager

Por padrão, as entidades do AWS Identity and Access Management (IAM) (usuários e perfis) não têm permissões para criar ou modificar os recursos do AWS Systems Manager. Eles também não podem executar tarefas usando o console do Systems Manager, a AWS Command Line Interface (AWS CLI) ou a API da AWS. Um administrador deve criar as políticas do IAM que concedam aos usuários e aos perfis permissões para executar operações de API específicas nos recursos especificados que precisam. O administrador deve anexar essas políticas aos usuários ou grupos que exigem essas permissões.

Veja a seguir um exemplo de uma política de permissões que permite que um usuário exclua documentos com nomes que começam com **`MyDocument-`** na região Leste dos EUA (Ohio) (us-east-2) Região da AWS.

JSON

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "ssm:DeleteDocument"
```

```
      ],
      "Resource" : [
        "arn:aws:ssm:us-east-2:111122223333:document/MyDocument-*"
      ]
    }
  ]
}
```

Para saber como criar uma política baseada em identidade do IAM usando esses exemplos de documentos de política JSON, consulte Criar políticas do IAM no Guia do usuário do IAM.

Tópicos

- Práticas recomendadas de política
- Exemplo: permissão para usar o console do Systems Manager
- Exemplo: permissão para possibilitar que os usuários visualizem suas próprias permissões
- Exemplo: permissão para ler e descrever parâmetros individuais
- Prevenção contra o ataque do “substituto confuso” em todos os serviços
- Exemplos de política gerenciada pelo cliente
- Visualizar documentos do Systems Manager com base em tags

Práticas recomendadas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos do Systems Manager em sua conta. Essas ações podem incorrer em custos para seus Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas gerenciadas pela AWS e avance para as permissões de privilégio mínimo: para começar a conceder permissões a seus usuários e workloads, use as políticas gerenciadas pela AWS, que concedem permissões para muitos casos de uso comuns. Elas estão disponíveis em seus Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo cliente da AWS que são específicas para seus casos de uso. Para obter mais informações, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para funções de trabalho no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também

conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre como usar o IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do usuário do IAM.

- Use condições nas políticas do IAM para restringir ainda mais o acesso: você pode adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Também pode usar condições para conceder acesso a ações de serviço, se elas forem usadas por meio de um AWS service (Serviço da AWS) específico, como o AWS CloudFormation. Para obter mais informações, consulte Elementos da política JSON do IAM: condição no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para obter mais informações, consulte Validação de políticas do IAM Access Analyzer no Guia do Usuário do IAM.
- Exigir autenticação multifator (MFA): se houver um cenário que exija usuários do IAM ou um usuário raiz em sua Conta da AWS, ative a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para obter mais informações, consulte Configuração de acesso à API protegido por MFA no Guia do Usuário do IAM.

Para obter mais informações sobre as práticas recomendadas do IAM, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

Exemplo: permissão para usar o console do Systems Manager

Para acessar o console do Systems Manager, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir listar e visualizar detalhes dos recursos do Systems Manager e outros recursos em sua Conta da AWS.

Se você criar uma política baseada em identidade mais restritiva do que as permissões mínimas requeridas, o console não funcionará conforme planejado para as entidades do IAM (usuários ou perfis) com essa política.

Não é necessário conceder permissões mínimas do console para usuários que fizerem chamadas somente a AWS CLI ou a API do AWS. Em vez disso, permita o acesso somente às ações que corresponderem a operação da API que você estiver tentando executar.

Para garantir que os usuários e os perfis ainda possam usar o console do Systems Manager, anexe também a política gerenciada pela AWS AmazonSSMFullAccess ou AmazonSSMReadOnlyAccess às entidades. Para obter informações, consulte Adicionar permissões a um usuário no Guia do usuário do IAM.

Exemplo: permissão para possibilitar que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou de forma programática usando a AWS CLI ou a API da AWS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
```

```
        }
    ]
}
```

Exemplo: permissão para ler e descrever parâmetros individuais

Example Ler e descrever um parâmetro

É possível conceder acesso a um parâmetro anexando a política a seguir a uma identidade.

JSON

```
{
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ssm:GetParameter",
      "ssm:DescribeParameters"
      ],
    "Resource": "arn:aws:ssm:us-east-1:111122223333:parameter/parameter-name"
  }
]
}
```

Prevenção contra o ataque do “substituto confuso” em todos os serviços

O problema de "confused deputy" é uma questão de segurança em que uma entidade que não tem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executá-la. Na AWS, a personificação entre serviços pode resultar no problema do ‘confused deputy’. A personificação entre serviços pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço chamado). O serviço de chamada pode ser manipulado de modo a usar suas permissões para atuar nos recursos de outro cliente de uma forma na qual ele não deveria ter permissão para acessar. Para evitar isso, a AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com entidades principais de serviço que receberam acesso aos recursos em sua conta.

Recomendamos o uso das chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` em políticas de recursos para limitar as permissões do AWS Systems

Manager que o concede a outro serviço no recurso para o recurso. Se o valor `aws:SourceArn` não contiver o ID da conta, como um nome do recurso da Amazon (ARN) para um bucket do S3, você deverá usar ambas as chaves de contexto de condição global para limitar as permissões. Se você utilizar ambas as chaves de contexto de condição global e o valor de `aws:SourceArn` contiver o ID da conta, o valor de `aws:SourceAccount` e a conta no valor de `aws:SourceArn` deverão utilizar o mesmo ID de conta quando utilizados na mesma declaração da política. Use `aws:SourceArn` se quiser apenas um recurso associado a acessibilidade de serviço. Use `aws:SourceAccount` se quiser permitir que qualquer recurso nessa conta seja associado ao uso entre serviços.

As seções a seguir fornecem políticas de exemplo para ferramentas do AWS Systems Manager.

Exemplo de política de ativação híbrida

Para perfis de serviço usados em uma ativação híbrida, o valor do `aws:SourceArn` deve ser o ARN da Conta da AWS. Especifique a Região da AWS no ARN onde você criou sua ativação híbrida. Se você não souber o ARN completo do recurso ou se estiver especificando vários recursos, use a chave da condição de contexto global `aws:SourceArn` com curingas (*) para as partes desconhecidas do ARN. Por exemplo, `arn:aws:ssm:*:`*`region`*`:`*`123456789012`*`:*`.

O exemplo a seguir demonstra o uso das chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` para automação para evitar o problema do "confused deputy" na região Leste dos EUA (Ohio) (us-east-2).

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"",
         "Effect":"Allow",
         "Principal":{
            "Service":"ssm.amazonaws.com"
         },
         "Action":"sts:AssumeRole",
         "Condition":{
            "StringEquals":{
               "aws:SourceAccount":"123456789012"
            },
            "ArnEquals":{
               "aws:SourceArn":"arn:aws:ssm:us-east-2:123456789012:*"
```

```
            }
        }
    }
  ]
}
```

Exemplo de política de sincronização de dados

O Inventory, o Explorer e o Compliance do Systems Manager permitem que você crie uma sincronização de dados de recursos para centralizar o armazenamento de seus dados operacionais (OpsData) em um bucket central do Amazon Simple Storage Service. Se você quiser criptografar a sincronização dos dados do recurso, usando o AWS Key Management Service (AWS KMS), crie uma nova chave que inclua a política a seguir, ou atualize uma chave existente e adicione essa política a ela. O `aws:SourceArn` e chaves de condição `aws:SourceAccount` nesta política impedem o problema confused deputy. Veja a seguir um exemplo de política:

JSON

```
{
    "Version": "2012-10-17",
    "Id": "ssm-access-policy",
    "Statement": [
        {
            "Sid": "ssm-access-policy-statement",
            "Action": [
                "kms:GenerateDataKey"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Resource": "arn:aws:kms:us-east-2:123456789012:key/KMS_key_id",
            "Condition": {
                "StringLike": {
                    "aws:SourceAccount": "123456789012"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:ssm:*:123456789012:role/aws-
service-role/ssm.amazonaws.com/AWSServiceRoleForAmazonSSM"
                }
            }
```

```
        }
    ]
}
```

> **Note**
>
> O ARN no exemplo de política permite que o sistema criptografe os dados operacionais de todas as fontes, exceto do AWS Security Hub. Se você precisar criptografar dados do Security Hub, por exemplo, se você usar o Explorer para coletar dados do Security Hub, você deve anexar uma política adicional que especifique o seguinte ARN: `"aws:SourceArn": "arn:aws:ssm:*:account-id:role/aws-service-role/opsdatasync.ssm.amazonaws.com/AWSServiceRoleForSystemsManagerOpsDataSync"`

Exemplos de política gerenciada pelo cliente

Você pode criar políticas independentes que você administra em sua própria Conta da AWS. Nós as chamamos de políticas gerenciadas pelo cliente. Você pode anexar essas políticas a várias entidades principais em sua Conta da AWS. Ao anexar uma política a uma entidade principal, você atribui à entidade as permissões que estão definidas na política. Para obter mais informações, consulte Criar exemplos de política gerenciada pelo cliente no Guia do usuário do IAM.

Os exemplos a seguir de políticas de usuário concedem permissões para várias ações do Systems Manager. Use-os para limitar o acesso do Systems Manager para as entidades do IAM (usuários e perfis). Essas políticas funcionam na execução de ações na API do Systems Manager, em AWS SDKs ou na AWS CLI. Para usuários que usam o console, você precisa conceder permissões adicionais específicas ao console. Para obter mais informações, consulte Exemplo: permissão para usar o console do Systems Manager.

> **Note**
>
> Todos os exemplos usam a região do Oeste dos EUA (Oregon) (us-west-2) e contêm IDs de contas fictícias. O ID da conta não deve ser especificado no nome do recurso da Amazon (ARN) para documentos públicos da AWS (documentos que começam com `AWS-*`).

Exemplos

- Exemplo 1: permitir que um usuário execute operações do Systems Manager em uma única região
- Exemplo 2: permitir que um usuário liste documentos de uma única região

Exemplo 1: permitir que um usuário execute operações do Systems Manager em uma única região

O exemplo a seguir concede permissões para executar operações do Systems Managersomente na região Leste dos EUA (Ohio) (us-east-2).

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:*"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-2:111122223333:*"
            ]
        }
    ]
}
```

Exemplo 2: permitir que um usuário liste documentos de uma única região

O exemplo a seguir concede permissões para listar todos os nomes de documentos que começam com **`Update`** na região Leste dos EUA (Ohio) (us-east-2).

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:ListDocuments"
```

```
            ],
            "Resource": [
                "arn:aws:ssm:us-east-2:111122223333:document/Update*"
            ]
        }
    ]
}
```

Exemplo 3: Permitir que um usuário utilize um documento específico do SSM para executar comandos em nós específicos

O exemplo de política do IAM permite a seguir que um usuário faça o seguinte na região Leste dos EUA (Ohio) (us-east-2):

- ListaSystems ManagerDocumentos (documentos SSM) e versões de documentos.
- Visualize detalhes sobre documentos.
- Envie um comando usando o documento especificado na política. O nome do documento é determinado pela entrada a seguir.

  ```
  arn:aws:ssm:us-east-2:aws-account-ID:document/Systems-Manager-document-name
  ```

- Envie um comando para três nós. Os nós são determinados pelas entradas a seguir na segunda seção `Resource`.

  ```
  "arn:aws:ec2:us-east-2:aws-account-ID:instance/i-02573cafcfEXAMPLE",
  "arn:aws:ec2:us-east-2:aws-account-ID:instance/i-0471e04240EXAMPLE",
  "arn:aws:ec2:us-east-2:aws-account-ID:instance/i-07782c72faEXAMPLE"
  ```

- Visualize detalhes sobre um comando depois que ele tiver sido enviado.
- Inicie e interrompa fluxos de trabalho no Automation, uma ferramenta do AWS Systems Manager.
- Obtenha informações sobre fluxos de trabalho de automação.

Se você deseja conceder permissões a um usuário para o uso deste documento para envio de comandos em qualquer nó ao qual o usuário tenha acesso, especifique uma entrada semelhante à seguinte na seção `Resource` e remova as outras entradas do nó. O exemplo a seguir usa a região Leste dos EUA (Ohio) (us-east-2).

```
"arn:aws:ec2:us-east-2:*:instance/*"
```

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ssm:ListDocuments",
                "ssm:ListDocumentVersions",
                "ssm:DescribeDocument",
                "ssm:GetDocument",
                "ssm:DescribeInstanceInformation",
                "ssm:DescribeDocumentParameters",
                "ssm:DescribeInstanceProperties"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "ssm:SendCommand",
            "Effect": "Allow",
            "Resource": [
                "arn:aws:ec2:us-east-2:111122223333:instance/
i-02573cafcfEXAMPLE",
                "arn:aws:ec2:us-east-2:111122223333:instance/
i-0471e04240EXAMPLE",
                "arn:aws:ec2:us-east-2:111122223333:instance/
i-07782c72faEXAMPLE",

                "arn:aws:ssm:us-east-2:111122223333:document/Systems-Manager-
document-name"
            ]
        },
        {
            "Action": [
                "ssm:CancelCommand",
                "ssm:ListCommands",
                "ssm:ListCommandInvocations"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
```

```
            "Action": "ec2:DescribeInstanceStatus",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "ssm:StartAutomationExecution",
            "Effect": "Allow",
            "Resource": [
                "arn:aws:ssm:us-east-2:111122223333:automation-definition/*"
            ]
        },
        {
            "Action": "ssm:DescribeAutomationExecutions",
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Action": [
                "ssm:StopAutomationExecution",
                "ssm:GetAutomationExecution"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Visualizar documentos do Systems Manager com base em tags

É possível utilizar condições na política baseada em identidade para controlar o acesso aos recursos do Systems Manager com base em tags. Este exemplo mostra como é possível criar uma política que permite visualizar um documento do SSM. No entanto, a permissão será concedida somente se a tag do documento `Owner` tiver o valor do nome desse usuário. Essa política também concede as permissões necessárias concluir essa ação no console.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListDocumentsInConsole",
            "Effect": "Allow",
            "Action": "ssm:ListDocuments",
            "Resource": "*"
        },
        {
            "Sid": "ViewDocumentIfOwner",
            "Effect": "Allow",
            "Action": "ssm:GetDocument",
            "Resource": "arn:aws:ssm:*:*:document/*",
            "Condition": {
                "StringEquals": {"ssm:ResourceTag/Owner": "${aws:username}"}
            }
        }
    ]
}
```

Você pode anexar essa política aos usuários na sua conta. Se um usuário chamado `richard-roe` tentar visualizar um documento do Systems Manager, o documento deverá ser marcado `0wner=richard-roe` ou `owner=richard-roe`. Caso contrário, o acesso será negado. A chave da tag de condição `0wner` corresponde a `0wner` e a `owner` porque os nomes de chaves de condição não fazem distinção entre maiúsculas e minúsculas. Para obter mais informações, consulte Elementos da política JSON do IAM: condição no Guia do usuário do IAM.

Políticas gerenciadas pela AWS do AWS Systems Manager

Uma política gerenciada pela AWS é uma política independente criada e administrada pela AWS. As políticas gerenciadas pela AWS são criadas para fornecer permissões a vários casos de uso comuns a fim de que você possa começar a atribuir permissões a usuários, grupos e perfis.

Lembre-se de que as políticas gerenciadas pela AWS podem não conceder permissões de privilégio mínimo para casos de uso específicos, por estarem disponíveis para uso por todos os clientes

da AWS. Recomendamos que você reduza ainda mais as permissões definindo as políticas gerenciadas pelo cliente que são específicas para seus casos de uso.

Você não pode alterar as permissões definidas em políticas gerenciadas pela AWS. Se a AWS atualiza as permissões definidas em um política gerenciada por AWS, a atualização afeta todas as identidades de entidades principais (usuários, grupos e perfis) às quais a política estiver vinculada. É provável que a AWS atualize uma política gerenciada por AWS quando um novo AWS service (Serviço da AWS) for lançado, ou novas operações de API forem disponibilizadas para os serviços existentes.

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

Tópicos

- Política gerenciada da AWS: AmazonSSMServiceRolePolicy
- Política gerenciada da AWS: AmazonSSMAutomationRole
- Política gerenciada pela AWS: AmazonSSMReadOnlyAccess
- Política gerenciada pela AWS: AWSSystemsManagerOpsDataSyncServiceRolePolicy
- Política gerenciada pela AWS: AmazonSSMManagedEC2InstanceDefaultPolicy
- Política gerenciada da AWS: SSMQuickSetupRolePolicy
- Política gerenciada da AWS: AWSQuickSetupDeploymentRolePolicy
- Política gerenciada da AWS: AWSQuickSetupPatchPolicyDeploymentRolePolicy
- Política gerenciada da AWS: AWSQuickSetupPatchPolicyBaselineAccess
- Política gerenciada da AWS: AWSSystemsManagerEnableExplorerExecutionPolicy
- Política gerenciada da AWS: AWSSystemsManagerEnableConfigRecordingExecutionPolicy
- Política gerenciada da AWS: AWSQuickSetupDevOpsGuruPermissionsBoundary
- Política gerenciada da AWS: AWSQuickSetupDistributorPermissionsBoundary
- Política gerenciada da AWS: AWSQuickSetupSSMHostMgmtPermissionsBoundary
- Política gerenciada da AWS: AWSQuickSetupPatchPolicyPermissionsBoundary
- Política gerenciada da AWS: AWSQuickSetupSchedulerPermissionsBoundary
- Política gerenciada da AWS: AWSQuickSetupCFGCPacksPermissionsBoundary
- Política gerenciada pela AWS: AWS-SSM-DiagnosisAutomation-AdministrationRolePolicy

- Política gerenciada pela AWS: AWS-SSM-DiagnosisAutomation-ExecutionRolePolicy
- Política gerenciada pela AWS: AWS-SSM-RemediationAutomation-AdministrationRolePolicy
- Política gerenciada pela AWS: AWS-SSM-RemediationAutomation-ExecutionRolePolicy
- Política gerenciada pela AWS: AWSQuickSetupSSMManageResourcesExecutionPolicy
- Política gerenciada pela AWS: AWSQuickSetupSSMLifecycleManagementExecutionPolicy
- Política gerenciada pela AWS: AWSQuickSetupSSMDeploymentRolePolicy
- Política gerenciada pela AWS: AWSQuickSetupSSMDeploymentS3BucketRolePolicy
- Política gerenciada da AWS: AWSQuickSetupEnableDHMCExecutionPolicy
- Política gerenciada da AWS: AWSQuickSetupEnableAREXExecutionPolicy
- Política gerenciada pela AWS: AWSQuickSetupManagedInstanceProfileExecutionPolicy
- Política gerenciada pela AWS: AWSQuickSetupFullAccess
- Política gerenciada pela AWS: AWSQuickSetupReadOnlyAccess
- Política gerenciada pela AWS: AWSQuickSetupManageJITNAResourcesExecutionPolicy
- Política gerenciada pela AWS: AWSQuickSetupJITNADeploymentRolePolicy
- Política gerenciada pela AWS: AWSSystemsManagerJustInTimeAccessServicePolicy
- Política gerenciada pela AWS: AWSSystemsManagerJustInTimeAccessTokenPolicy
- Política gerenciada pela AWS: AWSSystemsManagerJustInTimeAccessTokenSessionPolicy
- Política gerenciada pela AWS: AWSSystemsManagerJustInTimeNodeAccessRolePropagationPolicy
- Política gerenciada pela AWS: AWSSystemsManagerNotificationsServicePolicy
- Política gerenciada pela AWS: AWS-SSM-Automation-DiagnosisBucketPolicy
- Política gerenciada pela AWS: AWS-SSM-RemediationAutomation-OperationalAccountAdministrationRolePolicy
- Política gerenciada pela AWS: AWS-SSM-DiagnosisAutomation-OperationalAccountAdministrationRolePolicy
- Atualizações do Systems Manager para políticas gerenciadas pela AWS
- Políticas gerenciadas adicionais para o Systems Manager

Política gerenciada da AWS: AmazonSSMServiceRolePolicy

Esta política fornece acesso a vários recursos da AWS que são gerenciados pelo AWS Systems Manager ou usados nas operações do Systems Manager.

Não é possível anexar `AmazonSSMServiceRolePolicy` às entidades do AWS Identity and Access Management (IAM). Essa política é anexada a uma função vinculada ao serviço que permite que o AWS Systems Manager realize ações em seu nome. Para obter mais informações, consulte Usar perfis para coletar inventário e visualizar OpsData.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `ssm`: permite que as entidades principais iniciem e avancem as execuções tanto do Run Command quanto do Automation, recuperem informações sobre as operações de Run Command e do Automation, recuperem informações sobre calendários Change Calendar de parâmetros do Parameter Store, atualizem e recuperem informações sobre as configurações do serviço Systems Manager para recursos OpsCenter e leiam informações sobre tags que foram aplicadas aos recursos.
- `cloudformation`: permite que as entidades principais recuperem informações sobre operações e instâncias de conjuntos de pilhas e excluam conjuntos de pilhas no recurso `arn:aws:cloudformation:*:*:stackset/AWS-QuickSetup-SSM*:*`. Permite que as entidades principais excluam instâncias de pilha associadas aos seguintes recursos:

  ```
  arn:aws:cloudformation:*:*:stackset/AWS-QuickSetup-SSM*:*
  arn:aws:cloudformation:*:*:stackset-target/AWS-QuickSetup-SSM*:*
  arn:aws:cloudformation:*:*:type/resource/*
  ```

- `cloudwatch`: permite que as entidades principais recuperem informações sobre alarmes do Amazon CloudWatch.
- `compute-optimizer`: permite que as entidades principais recuperem o status de inscrição (opt-in) de uma conta no serviço AWS Compute Optimizer e recuperem recomendações para instâncias do Amazon EC2 que atendam a um conjunto específico de requisitos declarados.
- `config`: permite que as entidades principais recuperem as configurações de correção de informações e os gravadores de configuração no AWS Config e determinem se as regras do AWS Config e os recursos da AWS especificados estão em conformidade.
- `events`: que as entidades principais recuperem informações sobre as regras do EventBridge, criem regras e destinos do EventBridge exclusivamente para o serviço Systems Manager (`ssm.amazonaws.com`) e excluam regras e destinos para o recurso `arn:aws:events:*:*:rule/SSMExplorerManagedRule`.

- `ec2`: permite que as entidades principais recuperem informações sobre instâncias do Amazon EC2.
- `iam`: permite que as entidades principais transmitam permissões de perfis para o serviço Systems Manager (`ssm.amazonaws.com`).
- `lambda`: permite que as entidades principais invoquem funções do Lambda que são configuradas especificamente para uso pelo Systems Manager.
- `resource-explorer-2`: permite que as entidades principais recuperem dados sobre instâncias do EC2 para determinar se cada instância é gerenciada ou não pelo Systems Manager no momento.

  A ação `resource-explorer-2:CreateManagedView` é permitida para o recurso `arn:aws:resource-explorer-2:*:*:managed-view/AWSManagedViewForSSM*`.
- `resource-groups`: permite que as entidades principais recuperem grupos de recursos de lista e seus membros do AWS Resource Groups para recursos que pertencem a um grupo de recursos.
- `securityhub`: permite que as entidades principais recuperem informações sobre os recursos do hub do AWS Security Hub na conta atual.
- `states`: permite que as entidades principais iniciem e recuperem informações para o AWS Step Functions que estão configuradas especificamente para uso pelo Systems Manager.
- `support`: permite que as entidades principais recuperem informações sobre verificações e casos no AWS Trusted Advisor.
- `tag`: permite que as entidades principais recuperem informações sobre todos os recursos marcados ou anteriormente marcados com tags que estão localizados em uma Região da AWS especificada para uma conta.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AmazonSSMServiceRolePolicy no AWS Managed Policy Reference Guide.

Política gerenciada da AWS: AmazonSSMAutomationRole

É possível anexar a política `AmazonSSMAutomationRole` às identidades do IAM. Esta política fornece permissões para o serviço Automation AWS Systems Manager para executar atividades definidas nos runbooks de automação.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `lambda`: permite que as entidades principais invoquem funções do Lambda com nomes que começam com “Automation”. Isso é necessário para que os runbooks de automação executem funções do Lambda como parte de seu fluxo de trabalho.
- `ec2`: permite que as entidades principais realizem várias operações do Amazon EC2, incluindo criar, copiar e cancelar o registro de imagens; gerenciar snapshots; iniciar, executar, interromper e encerrar instâncias; gerenciar o status da instância; e criar, excluir e descrever tags. Essas permissões permitem que os runbooks de automação gerenciem os recursos do Amazon EC2 durante a execução.
- `cloudformation`: permite que as entidades principais criem, descrevam, atualizem e excluam pilhas do AWS CloudFormation. Isso permite que os runbooks de automação gerenciem a infraestrutura como código por meio do CloudFormation.
- `ssm`: permite que as entidades principais usem todas as ações do Systems Manager. Esse acesso abrangente é necessário para que os runbooks de automação interajam com todos os recursos do Systems Manager.
- `sns`: permite que as entidades principais publiquem mensagens nos tópicos do Amazon SNS com nomes que começam com “Automation”. Isso permite que os runbooks de automação enviem notificações durante a execução.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AmazonSSMAutomationRole no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AmazonSSMReadOnlyAccess

É possível anexar a política `AmazonSSMReadOnlyAccess` às identidades do IAM. Essa política concede acesso somente para leitura às operações de API do AWS Systems Manager, como `Describe*`, `Get*` e `List*`.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AmazonSSMReadOnlyAccess no AWS Managed Policy Reference Guide.

Política gerenciada pela AWS: AWSSystemsManagerOpsDataSyncServiceRolePolicy

Não é possível anexar a `AWSSystemsManagerOpsDataSyncServiceRolePolicy` às entidades do IAM. Essa política é anexada a uma função vinculada ao serviço que permite que o Systems Manager realize ações em seu nome. Para obter mais informações, consulte Usar perfis para criar OpsData e OpsItems para o Explorer.

O AWSSystemsManagerOpsDataSyncServiceRolePolicy permite que a função vinculada ao AWSServiceRoleForSystemsManagerOpsDataSync crie e atualize o OpsItems e OpsData nas descobertas do AWS Security Hub.

A política permite que o Systems Manager conclua as seguintes ações em todos os recursos relacionados ("Resource": "*"), exceto quando indicado:

- ssm:GetOpsItem [1]
- ssm:UpdateOpsItem [1]
- ssm:CreateOpsItem
- ssm:AddTagsToResource [2]
- ssm:UpdateServiceSetting [3]
- ssm:GetServiceSetting [3]
- securityhub:GetFindings
- securityhub:GetFindings
- securityhub:BatchUpdateFindings [4]

[1] As ações ssm:GetOpsItem e ssm:UpdateOpsItem têm permissões pela seguinte condição apenas para o serviço do Systems Manager:

```
"Condition": {
    "StringEquals": {
        "aws:ResourceTag/ExplorerSecurityHubOpsItem": "true"
    }
}
```

[2] A ação ssm:AddTagsToResource tem permissões apenas para o seguinte recurso:

```
arn:aws:ssm:*:*:opsitem/*
```

[3] As ações ssm:UpdateServiceSetting e ssm:GetServiceSetting têm permissões apenas nos seguintes recursos:

```
arn:aws:ssm:*:*:servicesetting/ssm/opsitem/*
arn:aws:ssm:*:*:servicesetting/ssm/opsdata/*
```

[4] A ação `securityhub:BatchUpdateFindings` tem permissões pela seguinte condição apenas para o serviço do Systems Manager:

```
{
   "Effect": "Deny",
   "Action": "securityhub:BatchUpdateFindings",
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "securityhub:ASFFSyntaxPath/Workflow.Status": "SUPPRESSED"
    }
   }
  },
  {
   "Effect": "Deny",
   "Action": "securityhub:BatchUpdateFindings",
   "Resource": "*",
   "Condition": {
    "Null": {
     "securityhub:ASFFSyntaxPath/Confidence": false
    }
   }
  },
  {
   "Effect": "Deny",
   "Action": "securityhub:BatchUpdateFindings",
   "Resource": "*",
   "Condition": {
    "Null": {
     "securityhub:ASFFSyntaxPath/Criticality": false
    }
   }
  },
  {
   "Effect": "Deny",
   "Action": "securityhub:BatchUpdateFindings",
   "Resource": "*",
   "Condition": {
    "Null": {
     "securityhub:ASFFSyntaxPath/Note.Text": false
    }
   }
  },
```

```
  {
   "Effect": "Deny",
   "Action": "securityhub:BatchUpdateFindings",
   "Resource": "*",
   "Condition": {
    "Null": {
     "securityhub:ASFFSyntaxPath/Note.UpdatedBy": false
    }
   }
  },
  {
   "Effect": "Deny",
   "Action": "securityhub:BatchUpdateFindings",
   "Resource": "*",
   "Condition": {
    "Null": {
     "securityhub:ASFFSyntaxPath/RelatedFindings": false
    }
   }
  },
  {
   "Effect": "Deny",
   "Action": "securityhub:BatchUpdateFindings",
   "Resource": "*",
   "Condition": {
    "Null": {
     "securityhub:ASFFSyntaxPath/Types": false
    }
   }
  },
  {
   "Effect": "Deny",
   "Action": "securityhub:BatchUpdateFindings",
   "Resource": "*",
   "Condition": {
    "Null": {
     "securityhub:ASFFSyntaxPath/UserDefinedFields.key": false
    }
   }
  },
  {
   "Effect": "Deny",
   "Action": "securityhub:BatchUpdateFindings",
   "Resource": "*",
```

```
   "Condition": {
    "Null": {
     "securityhub:ASFFSyntaxPath/UserDefinedFields.value": false
    }
   }
  },
  {
   "Effect": "Deny",
   "Action": "securityhub:BatchUpdateFindings",
   "Resource": "*",
   "Condition": {
    "Null": {
     "securityhub:ASFFSyntaxPath/VerificationState": false
    }
   }
```

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWSSystemsManagerOpsDataSyncServiceRolePolicy no AWS Managed Policy Reference Guide.

Política gerenciada pela AWS: AmazonSSMManagedEC2InstanceDefaultPolicy

Somente associe `AmazonSSMManagedEC2InstanceDefaultPolicy` a perfis do IAM para instâncias do Amazon EC2 que você deseja que tenham permissão para usar a funcionalidade Systems Manager. Você não deve vincular esse perfil a outras entidades do IAM, como usuários e grupos do IAM, ou a perfis do IAM que servem outros propósitos. Para obter mais informações, consulte Gerenciar instâncias do EC2 automaticamente com a Configuração de gerenciamento de hosts padrão.

Esta política concede permissões que permitem que o SSM Agent em sua instância do Amazon EC2 se comunique com o serviço Systems Manager na nuvem para realizar uma variedade de tarefas. Ela também concede permissões para os dois serviços que fornecem tokens de autorização para garantir que as operações sejam executadas na instância correta.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `ssm` — Permite que entidades principais recuperem documentos, executem comandos usando Run Command, estabeleçam sessões usando o Session Manager, coletem um inventário da instância e verificar se há patches e conformidade de patches usando o Patch Manager.

- `ssmmessages` — Permite que entidades principais acessem, para cada instância, um token de autorização personalizado criado pelo Amazon Message Gateway Service. O Systems Manager valida o token personalizado de autorização em relação ao nome do recurso da Amazon (ARN)da instância que foi fornecido na operação de API. Esse acesso é necessário para garantir que o SSM Agent execute as operações de API na instância correta.
- `ec2messages` — Permite que entidades principais acessem, para cada instância, um token de autorização personalizado criado pelo Amazon Message Delivery Service. O Systems Manager valida o token personalizado de autorização em relação ao nome do recurso da Amazon (ARN)da instância que foi fornecido na operação de API. Esse acesso é necessário para garantir que o SSM Agent execute as operações de API na instância correta.

Para obter informações relacionadas sobre os endpoints `ssmmessages` e `ec2messages`, incluindo as diferenças entre os dois, consulte Operações de API relacionadas a agentes (endpoints `ssmmessages` e `ec2messages`).

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AmazonSSMManagedEC2InstanceDefaultPolicy no AWS Managed Policy Reference Guide.

Política gerenciada da AWS: SSMQuickSetupRolePolicy

Não é possível anexar SSMQuickSetupRolePolicy às entidades do IAM. Essa política é anexada a uma função vinculada ao serviço que permite que o Systems Manager realize ações em seu nome. Para obter mais informações, consulte Usar funções para manter a integridade e a uniformidade dos recursos provisionados da Quick Setup.

Essa política concede permissões somente de leitura que permitem que o Systems Manager verifique a integridade da configuração, garanta o uso consistente dos parâmetros e dos recursos provisionados e corrija os recursos quando um desvio é detectado. Também concede permissões administrativas para criar um perfil vinculado ao serviço.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `ssm`: permite que as entidades principais leiam informações de sincronizações de dados de recursos e documentos do SSM no Systems Manager, incluindo contas de administrador delegado. Isso é necessário para que a Quick Setup possa determinar o estado em que os recursos configurados devem estar.

- `organizations`: permite que as entidades principais leiam informações sobre as contas-membros que pertencem a uma organização, conforme configurado em AWS Organizations. Isso é necessário para que a Quick Setup possa identificar todas as contas em uma organização em que as verificações de integridade dos recursos devem ser realizadas.
- `cloudformation`: permite que as entidades principais leiam as informações de AWS CloudFormation. Isso é necessário para que a Quick Setup possa coletar dados sobre as pilhas do AWS CloudFormation usadas para gerenciar o estado dos recursos e as operações do conjunto de pilhas do CloudFormation.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte SSMQuickSetupRolePolicy no Guia de referência de políticas gerenciadas da AWS.

Política gerenciada da AWS: AWSQuickSetupDeploymentRolePolicy

A política gerenciada `AWSQuickSetupDeploymentRolePolicy` oferece suporte a vários tipos de configuração Quick Setup. Esses tipos de configuração criam perfis e automações do IAM que configuram os serviços e os atributos da Amazon Web Services frequentemente usados com práticas recomendadas.

Você pode anexar `AWSQuickSetupDeploymentRolePolicy` às entidades do IAM.

Essa política concede as permissões administrativas necessárias para criar recursos associados às seguintes configurações da Quick Setup:

- Configurar o gerenciamento de host do Amazon EC2 usando a Quick Setup
- Criar um gravador de configuração do AWS Config usando o Quick Setup
- Implantar um pacote de conformidade do AWS Config usando o Quick Setup
- Configurar o DevOps Guru usando a Quick Setup
- Implantar pacotes do Distributor usando o Quick Setup
- Parar e iniciar instâncias do EC2 automaticamente de acordo com uma programação usando a Quick Setup

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `iam`: permite que as entidades principais gerenciem e excluam os perfis do IAM necessários para tarefas de configuração de automação e gerenciem as políticas de perfis de automação.
- `cloudformation`: permite que as entidades principais criem e gerenciem conjuntos de pilhas.
- `config`: permite que as entidades principais criem, gerenciem e excluam pacotes de conformidade.
- `events`: permite que as entidades principais criem, atualizem e excluam regras de eventos para ações agendadas.
- `resource-groups`: permite que as entidade principais recuperem consultas de recursos associadas a grupos de recursos segmentados pelas configurações da Quick Setup.
- `ssm`: permite que as entidades principais criem runbooks e associações de automação que apliquem configurações da Quick Setup.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWSQuickSetupDeploymentRolePolicy no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada da AWS: AWSQuickSetupPatchPolicyDeploymentRolePolicy

A política gerenciada `AWSQuickSetupPatchPolicyDeploymentRolePolicy` oferece suporte ao tipo Configurar a aplicação de patches para instâncias em uma organização usando uma política de patch do Quick Setup da Quick Setup. Esse tipo de configuração ajuda a automatizar a aplicação de patches em aplicações e nós em uma única conta ou em toda a organização.

Você pode anexar `AWSQuickSetupPatchPolicyDeploymentRolePolicy` às suas entidades do IAM. O Systems Manager também anexa essa política a um perfil de serviço que permite que o Systems Manager realize ações em seu nome.

Essa política concede permissões administrativas que permitem que o Quick Setup crie recursos associados a uma configuração de política de patch.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `iam`: permite que as entidades principais gerenciem e excluam os perfis do IAM necessários para tarefas de configuração de automação e gerenciem as políticas de perfis de automação.

- `cloudformation`: permite que as entidades principais leiam as informações da pilha do AWS CloudFormation e controlem as pilhas do AWS CloudFormation que foram criadas pela Quick Setup usando os conjuntos de pilhas do AWS CloudFormation.
- `ssm`: permite que as entidades principais criem, atualizem, leiam e excluam os runbooks de automação necessários para tarefas de configuração e para criar, atualizar e excluir associações do State Manager.
- `resource-groups`: permite que as entidade principais recuperem consultas de recursos associadas a grupos de recursos segmentados pelas configurações da Quick Setup.
- `s3`: permite que as entidades principais listem os buckets do Amazon S3 e gerenciem os buckets para armazenar os logs de acesso à política de patches.
- `lambda`: permite que as entidades principais gerenciem as funções de remediação do AWS Lambda que mantêm as configurações no estado correto.
- `logs`: permite que as entidades principais descrevam e gerenciem grupos de log para recursos de configuração do Lambda.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWSQuickSetupPatchPolicyDeploymentRolePolicy no Guia de referência de políticas gerenciadas da AWS.

Política gerenciada da AWS: AWSQuickSetupPatchPolicyBaselineAccess

A política gerenciada `AWSQuickSetupPatchPolicyBaselineAccess` oferece suporte ao tipo Configurar a aplicação de patches para instâncias em uma organização usando uma política de patch do Quick Setup da Quick Setup. Esse tipo de configuração ajuda a automatizar a aplicação de patches em aplicações e nós em uma única conta ou em toda a organização.

Você pode anexar `AWSQuickSetupPatchPolicyBaselineAccess` às suas entidades do IAM. O Systems Manager também anexa essa política a um perfil de serviço que permite que o Systems Manager realize ações em seu nome.

Esta política fornece permissões somente de leitura para acessar listas de referência de patches que foram configuradas por um administrador na Conta da AWS atual ou na organização usando a Quick Setup. As listas de referência de patches são armazenadas em um bucket do Amazon S3 e podem ser usadas para aplicar patches em instâncias em uma única conta ou em toda a organização.

Detalhes das permissões

Esta política inclui a seguinte permissão.

- `s3`: permite que as entidades principais leiam substituições da lista de referência de patches armazenadas nos buckets do Amazon S3.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWSQuickSetupPatchPolicyBaselineAccess no Guia de referência de políticas gerenciadas da AWS.

Política gerenciada da AWS: `AWSSystemsManagerEnableExplorerExecutionPolicy`

A política gerenciada `AWSSystemsManagerEnableExplorerExecutionPolicy` oferece suporte à habilitação do Explorer, uma ferramenta do AWS Systems Manager.

Você pode anexar `AWSSystemsManagerEnableExplorerExecutionPolicy` às suas entidades do IAM. O Systems Manager também anexa essa política a um perfil de serviço que permite que o Systems Manager realize ações em seu nome.

Esta política concede permissões administrativas para habilitar o Explorer. Isso inclui permissões para atualizar as configurações de serviços relacionados do Systems Manager e criar uma função vinculada a serviços para o Systems Manager.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `config`: permite que as entidades principais ajudem na habilitação do Explorer, fornecendo acesso somente de leitura aos detalhes do gravador de configuração.
- `iam`: permite que as entidades principais ajudem a habilitar o Explorer.
- `ssm`: permite que as entidades principais iniciem um fluxo de trabalho de automação que habilite o Explorer.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWSSystemsManagerEnableExplorerExecutionPolicy no Guia de referência de políticas gerenciadas da AWS.

Política gerenciada da AWS: **AWSSystemsManagerEnableConfigRecordingExecutionPolicy**

A política gerenciada `AWSSystemsManagerEnableConfigRecordingExecutionPolicy` oferece suporte ao tipo de configuração Criar um gravador de configuração do AWS Config usando o Quick Setup do Quick Setup. Esse tipo de configuração habilita s Quick Setup para rastrear e registrar alterações nos tipos de recursos da AWS que você escolher para o AWS Config. Ele também permite que a Quick Setup configure as opções de entrega e notificações para os dados gravados.

Você pode anexar `AWSSystemsManagerEnableConfigRecordingExecutionPolicy` às suas entidades do IAM. O Systems Manager também anexa essa política a um perfil de serviço que permite que o Systems Manager realize ações em seu nome.

Esta política concede permissões administrativas que permitem que a Quick Setup habilite e configure o registro de configuração do AWS Config.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `s3`: permite que as entidades principais criem e configurem buckets do Amazon S3 para entrega de gravações de configuração.
- `sns`: permite que as entidades principais listem e criem tópicos do Amazon SNS.
- `config`: permite que as entidades principais configurem e iniciem o gravador de configuração; e ajudem a habilitar o Explorer.
- `iam`: permite que as entidades principais criem, obtenham e transmitam uma função vinculada a serviços para o AWS Config e criem uma função vinculada a serviços para o Systems Manager; e ajudem a habilitar o Explorer.
- `ssm`: permite que as entidades principais iniciem um fluxo de trabalho de automação que habilite o Explorer.
- `compute-optimizer`: permite que as entidades principais ajudem a habilitar o Explorer, fornecendo acesso somente leitura para determinar se um recurso está inscrito no AWS Compute Optimizer.
- `support`: permite que as entidades principais ajudem a habilitar o Explorer, fornecendo acesso somente leitura para determinar se um recurso está inscrito no AWS Compute Optimizer.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWSSystemsManagerEnableConfigRecordingExecutionPolicy no Guia de referência de políticas gerenciadas da AWS.

Política gerenciada da AWS: AWSQuickSetupDevOpsGuruPermissionsBoundary

> **Note**
>
> Esta política é um limite de permissões. Um limite de permissões define o número máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM. Você não deve usar e anexar políticas de limite de permissões da Quick Setup por conta própria. As políticas de limite de permissões da Quick Setup só devem ser anexadas a perfis gerenciados da Quick Setup. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.

A política gerenciada `AWSQuickSetupDevOpsGuruPermissionsBoundary` oferece suporte ao tipo Configurar o DevOps Guru usando a Quick Setup. O tipo de configuração habilita o Amazon DevOps Guru baseado em machine learning. O serviço DevOps Guru pode ajudar a melhorar o desempenho operacional e a disponibilidade de uma aplicação.

Quando você cria uma configuração `AWSQuickSetupDevOpsGuruPermissionsBoundary` usando a Quick Setup, o sistema aplica esse limite de permissões aos perfis do IAM que são criados quando a configuração é implantada. O limite de permissões limita o escopo dos perfis que a Quick Setup cria.

Esta política concede permissões administrativas que permitem que a Quick Setup habilite e configure o Amazon DevOps Guru.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `iam`: permite que as entidades principais criem funções vinculadas a serviços para o DevOps Guru e o Systems Manager; e listem funções que ajudam a habilitar o Explorer.
- `cloudformation`: permite que as entidades principais listem e descrevam as pilhas do AWS CloudFormation.
- `sns`: permite que as entidades principais listem e criem tópicos do Amazon SNS.

- `devops-guru`: permite que as entidades principais configurem o DevOps Guru e adicionem um canal de notificação.
- `config`: permite que as entidades principais ajudem na habilitação do Explorer, fornecendo acesso somente de leitura aos detalhes do gravador de configuração.
- `ssm`: permite que as entidades principais iniciem um fluxo de trabalho de automação que habilita o Explorer; e leiam e atualizem as configurações de serviço do Explorer.
- `compute-optimizer`: permite que as entidades principais ajudem a habilitar o Explorer, fornecendo acesso somente leitura para determinar se um recurso está inscrito no AWS Compute Optimizer.
- `support`: permite que as entidades principais ajudem a habilitar o Explorer, fornecendo acesso somente leitura para determinar se um recurso está inscrito no AWS Compute Optimizer.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWSQuickSetupDevOpsGuruPermissionsBoundary no Guia de referência de políticas gerenciadas da AWS.

Política gerenciada da AWS: AWSQuickSetupDistributorPermissionsBoundary

> **Note**
>
> Esta política é um limite de permissões. Um limite de permissões define o número máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM. Você não deve usar e anexar políticas de limite de permissões da Quick Setup por conta própria. As políticas de limite de permissões da Quick Setup só devem ser anexadas a perfis gerenciados da Quick Setup. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.

A política gerenciada `AWSQuickSetupDistributorPermissionsBoundary` oferece suporte ao tipo de configuração Implantar pacotes do Distributor usando o Quick Setup do Quick Setup. O tipo de configuração ajuda a permitir a distribuição de pacotes de software, como agentes, para instâncias do Amazon Elastic Compute Cloud (Amazon EC2), usando o Distributor, uma ferramenta do AWS Systems Manager.

Quando você cria uma configuração `AWSQuickSetupDistributorPermissionsBoundary` usando a Quick Setup, o sistema aplica esse limite de permissões aos perfis do IAM que são criados

quando a configuração é implantada. O limite de permissões limita o escopo dos perfis que a Quick Setup cria.

Esta política concede permissões administrativas para que a Quick Setup possa distribuir pacotes de software, como agentes, para suas instâncias do Amazon EC2 usando o Distributor.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `iam`: permite que as entidades principais obtenham e transmitam a função de automação do Distributor; criem, leiam, atualizem e excluam a função de instância padrão; passem a função de instância padrão para o Amazon EC2 e Systems Manager; anexem políticas de gerenciamento de instâncias às funções de instância; criem uma função vinculada a serviços para o Systems Manager; adicionem a função de instância padrão aos perfis de instância; leiam informações sobre perfis do IAM e perfis de instância; e criem o perfil de instância padrão.
- `ec2`: permite que as entidades principais associem o perfil de instância padrão às instâncias do EC2 e ajudem a habilitar o Explorer.
- `ssm`: permite que as entidades principais iniciem fluxos de trabalho de automação que configurem instâncias e instalem pacotes; e ajudem a iniciar o fluxo de trabalho de automação que habilita o Explorer; e leiam e atualizem as configurações de serviço do Explorer.
- `config`: permite que as entidades principais ajudem na habilitação do Explorer, fornecendo acesso somente de leitura aos detalhes do gravador de configuração.
- `compute-optimizer`: permite que as entidades principais ajudem a habilitar o Explorer, fornecendo acesso somente leitura para determinar se um recurso está inscrito no AWS Compute Optimizer.
- `support`: permite que as entidades principais ajudem a habilitar o Explorer, fornecendo acesso somente leitura para determinar se um recurso está inscrito no AWS Compute Optimizer.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWSQuickSetupDistributorPermissionsBoundary no Guia de referência de políticas gerenciadas da AWS.

Política gerenciada da AWS: AWSQuickSetupSSMHostMgmtPermissionsBoundary

> **Note**
>
> Esta política é um limite de permissões. Um limite de permissões define o número máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM. Você não deve usar e anexar políticas de limite de permissões da Quick Setup por conta própria. As políticas de limite de permissões da Quick Setup só devem ser anexadas a perfis gerenciados da Quick Setup. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.

A política gerenciada `AWSQuickSetupSSMHostMgmtPermissionsBoundary` oferece suporte ao tipo de configuração Configurar o gerenciamento de host do Amazon EC2 usando a Quick Setup do Quick Setup. Esse tipo de configuração configura os perfis do IAM e habilita ferramentas comumente usadas do Systems Manager para gerenciar com segurança suas instâncias do Amazon EC2.

Quando você cria uma configuração `AWSQuickSetupSSMHostMgmtPermissionsBoundary` usando a Quick Setup, o sistema aplica esse limite de permissões aos perfis do IAM que são criados quando a configuração é implantada. O limite de permissões limita o escopo dos perfis que a Quick Setup cria.

Esta política concede permissões administrativas para que a Quick Setup possa configurar as ferramentas do Systems Manager necessárias para gerenciar com segurança as instâncias do EC2.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `iam`: permite que as entidades principais obtenham e passem o perfil de serviço para a automação. Permite que as entidades principais criem, leiam, atualizem e excluam a função de instância padrão; passem a função de instância padrão para o Amazon EC2 e Systems Manager; anexem políticas de gerenciamento de instâncias às funções de instância; criem uma função vinculada a serviços para o Systems Manager; adicionem a função de instância padrão aos perfis de instância; leiam informações sobre perfis do IAM e perfis de instância; e criem o perfil de instância padrão.
- `ec2`: permite que as entidades principais associem e desassociem o perfil de instância padrão das instâncias do EC2.

- `ssm`: permite que as entidades principais iniciem fluxos de trabalho de automação que habilitam o Explorer; leiam e atualizem as configurações de serviço do Explorer; configurem instâncias; e habilitem as ferramentas do Systems Manager nas instâncias.
- `compute-optimizer`: permite que as entidades principais ajudem a habilitar o Explorer, fornecendo acesso somente leitura para determinar se um recurso está inscrito no AWS Compute Optimizer.
- `support`: permite que as entidades principais ajudem a habilitar o Explorer, fornecendo acesso somente leitura para determinar se um recurso está inscrito no AWS Compute Optimizer.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWSQuickSetupSSMHostMgmtPermissionsBoundary no Guia de referência de políticas gerenciadas da AWS.

Política gerenciada da AWS: AWSQuickSetupPatchPolicyPermissionsBoundary

> **Note**
>
> Esta política é um limite de permissões. Um limite de permissões define o número máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM. Você não deve usar e anexar políticas de limite de permissões da Quick Setup por conta própria. As políticas de limite de permissões da Quick Setup só devem ser anexadas a perfis gerenciados da Quick Setup. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.

A política gerenciada `AWSQuickSetupPatchPolicyPermissionsBoundary` oferece suporte ao tipo Configurar a aplicação de patches para instâncias em uma organização usando uma política de patch do Quick Setup da Quick Setup. Esse tipo de configuração ajuda a automatizar a aplicação de patches em aplicações e nós em uma única conta ou em toda a organização.

Quando você cria uma configuração `AWSQuickSetupPatchPolicyPermissionsBoundary` usando a Quick Setup, o sistema aplica esse limite de permissões aos perfis do IAM que são criados quando a configuração é implantada. O limite de permissões limita o escopo dos perfis que a Quick Setup cria.

Esta política concede permissões administrativas para que a Quick Setup possa habilitar e configurar políticas de patch no Patch Manager, uma ferramenta do AWS Systems Manager.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `iam`: permite que as entidades principais obtenham a função de automação do Patch Manager; passem funções de automação para operações de correção do Patch Manager; criem o perfil de instância padrão`AmazonSSMRoleForInstancesQuickSetup`; passem a função de instância padrão para o Amazon EC2 e o Systems Manager; anexem políticas gerenciadas da AWS selecionadas à função de instância; criem uma função vinculada a serviços para o Systems Manager; adicionem a função de instância padrão aos perfis e funções de instância; criem um perfil de instância padrão; e marquem funções que tenham permissões de leitura substituições da lista de referência de patches.
- `ssm`: permite que as entidades principais atualizem a função da instância que é gerenciada pelo Systems Manager; gerenciem associações criadas pelas políticas de patch do Patch Manager criadas no Quick Setup; marquem instâncias segmentadas por uma configuração de política de patch; leiam informações sobre instâncias e status de patches; iniciem fluxos de trabalho de automação que configuram, habilitam e corrijam a correção de instâncias; iniciem fluxos de trabalho de automação que habilitam o Explorer; ajudem a habilitar o Explorer; e leiam e atualizem as configurações do serviço do Explorer.
- `ec2`: permite que as entidades principais associem e desassociem o perfil de instância padrão das instâncias do EC2; marquem instâncias segmentadas por uma configuração de política de patch; marquem instâncias segmentadas por uma configuração de política de patch; e ajudem a habilitar o Explorer.
- `s3`: permite que as entidades principais criem e configurem buckets do S3 para armazenar substituições da lista de referência de patches.
- `lambda`: permite que as entidades principais invoquem funções do AWS Lambda que configuram a aplicação de patches e realizem operações de limpeza após a exclusão de uma configuração de política de patch da Quick Setup.
- `logs`: permite que as entidades principais configurem o registro em log para funções Patch Manager Quick Setup AWS Lambda.
- `config`: permite que as entidades principais ajudem na habilitação do Explorer, fornecendo acesso somente de leitura aos detalhes do gravador de configuração.
- `compute-optimizer`: permite que as entidades principais ajudem a habilitar o Explorer, fornecendo acesso somente leitura para determinar se um recurso está inscrito no AWS Compute Optimizer.

- `support`: permite que as entidades principais ajudem a habilitar o Explorer, fornecendo acesso somente leitura para determinar se um recurso está inscrito no AWS Compute Optimizer.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWSQuickSetupPatchPolicyPermissionsBoundary no Guia de referência de políticas gerenciadas da AWS.

Política gerenciada da AWS: AWSQuickSetupSchedulerPermissionsBoundary

> **Note**
>
> Esta política é um limite de permissões. Um limite de permissões define o número máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM. Você não deve usar e anexar políticas de limite de permissões da Quick Setup por conta própria. As políticas de limite de permissões da Quick Setup só devem ser anexadas a perfis gerenciados da Quick Setup. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.

A política gerenciada `AWSQuickSetupSchedulerPermissionsBoundary` oferece suporte ao tipo de configuração Parar e iniciar instâncias do EC2 automaticamente de acordo com uma programação usando a Quick Setup do Quick Setup. Este tipo de configuração permite que você interrompa e inicie suas instâncias do EC2 e outros recursos nos horários que você especificar.

Quando você cria uma configuração `AWSQuickSetupSchedulerPermissionsBoundary` usando a Quick Setup, o sistema aplica esse limite de permissões aos perfis do IAM que são criados quando a configuração é implantada. O limite de permissões limita o escopo dos perfis que a Quick Setup cria.

Esta política concede permissões administrativas que permitem que a Quick Setup habilite e configurar operações agendadas nas instâncias do EC2 e em outros recursos.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `iam`: permite que as entidades principais recuperem e passem perfis para ações de automação de gerenciamento de instâncias; gerenciem, passem e anexem perfis de instância padrão para

o gerenciamento de instâncias do EC2; criem perfis de instância padrão; adicionem perfis de instância padrão para perfis de instância; criem um perfil vinculado ao serviço para o Systems Manager; leiam informações sobre perfis do IAM e perfis de instância; associem um perfil de instância padrão às instâncias do EC2; e iniciem fluxos de trabalho de automação para configurar instâncias e habilitar ferramentas do Systems Manager nelas.

- `ssm`: permite que as entidades principais iniciem fluxos de trabalho de automação que habilitam o Explorer; e leiam e atualizem as configurações de serviço do Explorer.
- ec2 : permite que as entidades principais localizem instâncias segmentadas e as iniciem e interrompam de acordo com um cronograma.
- `config`: permite que as entidades principais ajudem na habilitação do Explorer, fornecendo acesso somente de leitura aos detalhes do gravador de configuração.
- `compute-optimizer`: permite que as entidades principais ajudem a habilitar o Explorer, fornecendo acesso somente leitura para determinar se um recurso está inscrito no AWS Compute Optimizer.
- `support`: permite que as entidades principais ajudem a habilitar o Explorer, fornecendo acesso somente de leitura às verificações do AWS Trusted Advisor de uma conta.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWSQuickSetupSchedulerPermissionsBoundary no Guia de referência de políticas gerenciadas da AWS.

Política gerenciada da AWS: AWSQuickSetupCFGCPacksPermissionsBoundary

> **Note**
>
> Esta política é um limite de permissões. Um limite de permissões define o número máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM. Você não deve usar e anexar políticas de limite de permissões da Quick Setup por conta própria. As políticas de limite de permissões da Quick Setup só devem ser anexadas a perfis gerenciados da Quick Setup. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.

A política gerenciada `AWSQuickSetupCFGCPacksPermissionsBoundary` oferece suporte ao tipo de configuração Implantar um pacote de conformidade do AWS Config usando o Quick Setup do Quick Setup. Este tipo de configuração implanta pacotes de conformidade do AWS Config. Os

pacotes de conformidade são uma coleção de regras e ações de remediação do AWS Config que possam ser implantadas como uma única entidade.

Quando você cria uma configuração `AWSQuickSetupCFGCPacksPermissionsBoundary` usando a Quick Setup, o sistema aplica esse limite de permissões aos perfis do IAM que são criados quando a configuração é implantada. O limite de permissões limita o escopo dos perfis que a Quick Setup cria.

Esta política concede permissões administrativas que permitem ao Quick Setup implantar pacotes de conformidade do AWS Config.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `iam`: permite que as entidades principais criem, obtenham e passem uma função vinculada a serviço para o AWS Config.
- `sns`: permite que as entidades principais listem aplicações da plataforma no Amazon SNS.
- `config`: permite que as entidades principais implantem pacotes de conformidade do AWS Config; obtenham o status dos pacotes de conformidade; e obtenham informações sobre gravadores de configuração.
- `ssm`: permite que as entidades principais obtenham informações sobre documentos SSM e fluxos de trabalho de automação; obtenham informações sobre tags de recursos; e obtenham informações e atualizem as configurações do serviço.
- `compute-optimizer`: permite que as entidades principais obtenham o status de aceitação de uma conta.
- `support`: permite que as entidades principais obtenham informações sobre verificações do AWS Trusted Advisor.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWSQuickSetupCFGCPacksPermissionsBoundary no Guia de referência de políticas gerenciadas da AWS.

Política gerenciada pela AWS: AWS-SSM-DiagnosisAutomation-AdministrationRolePolicy

A política `AWS-SSM-DiagnosisAutomation-AdministrationRolePolicy` concede permissões para diagnosticar problemas com nós que interagem com os serviços do Systems

Manager iniciando fluxos de trabalho do Automation em contas e regiões nas quais os nós são gerenciados.

Você pode anexar `AWS-SSM-DiagnosisAutomation-AdministrationRolePolicy` às suas entidades do IAM. O Systems Manager também anexa essa política a um perfil de serviço que permite que o Systems Manager realize ações de diagnóstico em seu nome.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `ssm`: permite que as entidades principais executem runbooks de automação que diagnosticam problemas nos nós e acessem o status de execução de fluxos de trabalho e recuperem os detalhes de execução de automação. A política concede permissões para descrever execuções de automação, descrever execuções de etapas de automação, obter detalhes da execução de automação e iniciar execuções de automação para documentos relacionados ao diagnóstico.
- `kms`: permite que as entidades principais usem chaves do AWS Key Management Service especificadas pelo cliente para decodificação e geração de chaves de dados ao acessar objetos criptografados em buckets do Amazon S3 usados para operações de diagnóstico. Essas permissões são restritas às chaves marcadas com `SystemsManagerManaged` e são usadas por meio do serviço do Amazon S3 com requisitos específicos de contexto de criptografia.
- `sts`: permite que as entidades principais assumam perfis de execução de diagnóstico para executar runbooks do Automation na mesma conta. Essa permissão é restrita aos perfis com o padrão de nomenclatura `AWS-SSM-DiagnosisExecutionRole` e inclui uma condição para garantir que a conta do recurso corresponda à conta principal.
- `iam`: permite que as entidades principais passem o perfil de administração de diagnóstico para o Systems Manager a fim de executar runbooks de automação. Essa permissão é restrita a perfis com o padrão de nomenclatura `AWS-SSM-DiagnosisAdminRole` e só pode ser passada para o serviço do Systems Manager.
- `s3`: permite que as entidades principais acessem, leiam, gravem e excluam objetos nos buckets do Amazon S3 usados para operações de diagnóstico. Essas permissões são restritas a buckets com o padrão de nomenclatura `do-not-delete-ssm-diagnosis-` e incluem condições para garantir que as operações sejam realizadas na mesma conta.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWS-SSM-DiagnosisAutomation-AdministrationRolePolicy no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AWS-SSM-DiagnosisAutomation-ExecutionRolePolicy

A política gerenciada `AWS-SSM-DiagnosisAutomation-ExecutionRolePolicy` fornece permissão administrativa para executar runbooks do Automation em uma região e Conta da AWS de destino a fim de diagnosticar problemas com nós gerenciados que interagem com os serviços do Systems Manager.

Você pode anexar `AWS-SSM-DiagnosisAutomation-ExecutionRolePolicy` às suas entidades do IAM. O Systems Manager também anexa essa política a um perfil de serviço que permite que o Systems Manager realize ações em seu nome.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `ec2`: permite que as entidades principais descrevam os recursos do Amazon EC2 e da Amazon VPC e suas configurações para diagnosticar problemas com os serviços do Systems Manager. Isso inclui permissões para descrever VPCs, atributos e endpoints de VPCs, sub-redes, grupos de segurança, instâncias e gateways da internet.
- `ssm`: permite que as entidades principais executem runbooks do Automation específicos para diagnósticos e acessem o status do fluxo de trabalho de automação e os metadados de execução. Isso inclui permissões para descrever execuções de etapas de automação, informações de instâncias e execuções de automação, e também obter detalhes da execução de automação e iniciar execuções de automação para documentos específicos de diagnóstico do EC2 não gerenciados pela AWS.
- `kms`: permite que as entidades principais usem chaves do AWS Key Management Service especificadas pelo cliente para decodificação e geração de chaves de dados ao acessar objetos criptografados em buckets do Amazon S3 usados para operações de diagnóstico. Essas permissões são restritas às chaves marcadas com `SystemsManagerManaged` e usadas por meio do serviço do Amazon S3 com requisitos específicos de contexto de criptografia para buckets de diagnóstico.
- `iam`: permite que as entidades principais passem o perfil de execução de diagnóstico para o Systems Manager a fim de executar documentos de automação. Essa permissão é restrita a perfis com o padrão de nomenclatura `AWS-SSM-DiagnosisExecutionRole` e só pode ser passada para o serviço do Systems Manager.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWS-SSM-DiagnosisAutomation-ExecutionRolePolicy no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AWS-SSM-RemediationAutomation-AdministrationRolePolicy

A política `AWS-SSM-RemediationAutomation-AdministrationRolePolicy` fornece permissões para corrigir problemas com os serviços do Systems Manager executando atividades definidas nos documentos de automação, usados principalmente para executar os documentos de automação. Essa política permite iniciar fluxos de trabalho de automação em contas e regiões em que os nós são gerenciados para resolver problemas de conectividade e configuração.

Você pode anexar `AWS-SSM-RemediationAutomation-AdministrationRolePolicy` às suas entidades do IAM. O Systems Manager também anexa essa política a um perfil de serviço que permite que o Systems Manager realize ações de correção em seu nome.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `ssm`: permite que as entidades principais executem runbooks de automação que remediam problemas nos nós e acessem o status de execução de fluxos de trabalho e recuperem os detalhes de execução de automação. A política concede permissões para descrever execuções de automação e execuções de etapas de automação, obter detalhes da execução de automação e iniciar execuções de automação para documentos relacionados ao diagnóstico.
- `kms`: permite que as entidades principais usem chaves do AWS Key Management Service especificadas pelo cliente para descriptografia e geração de chaves de dados ao acessar objetos criptografados em buckets do Amazon S3 usados para operações de correção. Essas permissões são restritas às chaves marcadas com `SystemsManagerManaged` e são usadas por meio do serviço do Amazon S3 com requisitos específicos de contexto de criptografia.
- `sts`: permite que as entidades principais assumam perfis de execução de correção para executar runbooks de automação na mesma conta. Essa permissão é restrita aos perfis com o padrão de nomenclatura `AWS-SSM-RemediationExecutionRole` e inclui uma condição para garantir que a conta do recurso corresponda à conta principal.
- `iam`: permite que as entidades principais passem o perfil de administração de correção para o Systems Manager a fim de executar runbooks de automação. Essa permissão é restrita a perfis

com o padrão de nomenclatura `AWS-SSM-RemediationAdminRole` e só pode ser passada para o serviço do Systems Manager.

- `s3`: permite que as entidades principais acessem, leiam, gravem e excluam objetos nos buckets do Amazon S3 usados para operações de correção. Essas permissões são restritas a buckets com o padrão de nomenclatura `do-not-delete-ssm-diagnosis-` e incluem condições para garantir que as operações sejam realizadas na mesma conta.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWS-SSM-RemediationAutomation-AdministrationRolePolicy no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AWS-SSM-RemediationAutomation-ExecutionRolePolicy

A política gerenciada `AWS-SSM-RemediationAutomation-ExecutionRolePolicy` fornece permissões administrativas para executar runbooks de automação em uma região e conta de destino a fim de corrigir problemas de conectividade e rede com nós gerenciados que interagem com os serviços do Systems Manager. Essa política permite atividades de correção definidas nos documentos de automação, usadas principalmente para executar os documentos de automação para resolver problemas de conectividade e configuração.

Também é possível anexar a política às suas entidades do IAM. O Systems Manager também anexa essa política a um perfil de serviço que permite que o Systems Manager execute ações de correção em seu nome.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `ssm`: permite que as entidades principais recuperem informações sobre as execuções de automação e suas etapas de execução e iniciem runbooks específicos de automação de correção, incluindo documentos de `AWS-OrchestrateUnmanagedEC2Actions` e`AWS-RemediateSSMAgent`. A política concede permissões para descrever execuções de automação e execuções de etapas de automação, obter detalhes da execução de automação e iniciar execuções de automação para documentos relacionados ao diagnóstico.
- `ec2`: permite que as entidades principais descrevam e modifiquem recursos de rede da Amazon VPC para corrigir problemas de conectividade. Isso inclui:
  - Descrição de atributos, sub-redes e endpoints da Amazon VPC e grupos de segurança.

  - Criação de endpoints da Amazon VPC para serviços do Systems Manager (`ssm`, `ssmmessages` e `ec2messages`) com as tags necessárias.
  - Modificação dos atributos da Amazon VPC para habilitar o suporte ao DNS e nomes de host.
  - Criação e gerenciamento de grupos de segurança com tags específicas para acesso ao endpoint da Amazon VPC.
  - Autorização e revogação de regras de grupos de segurança para acesso HTTPS com as tags apropriadas.
  - Criação de tags em grupos de segurança, regras de grupos de segurança e endpoints da Amazon VPC durante a criação de recursos.
- `kms`: permite que as entidades principais usem chaves do AWS Key Management Service especificadas pelo cliente para descriptografia e geração de chaves de dados ao acessar objetos criptografados em buckets do Amazon S3 usados para operações de correção. Essas permissões são restritas às chaves marcadas com `SystemsManagerManaged` e são usadas por meio do serviço do Amazon S3 com requisitos específicos de contexto de criptografia.
- `iam`: permite que as entidades principais passem o perfil de execução de correção para o Systems Manager a fim de executar runbooks de automação. Essa permissão é restrita a perfis com o padrão de nomenclatura `AWS-SSM-RemediationExecutionRole` e só pode ser passada para o serviço do Systems Manager.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWS-SSM-RemediationAutomation-ExecutionRolePolicy no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AWSQuickSetupSSMManageResourcesExecutionPolicy

Essa política concede permissões que permitem à Quick Setup executar o runbook do Automation `AWSQuickSetupType-SSM-SetupResources`. Esse runbook cria perfis do IAM para associações da Quick Setup que, por sua vez, são criadas por uma implantação do `AWSQuickSetupType-SSM`. Ele também concede permissões para limpar um bucket do Amazon S3 associado durante uma operação de exclusão da Quick Setup.

Você pode anexar a política Systems Manager às suas entidades do IAM. O também anexa essa política a um perfil de serviço que permite que o Systems Manager execute ações em seu nome.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `iam`: permite que as entidades principais listem e gerenciem perfis do IAM para uso com as operações do Quick Setup Systems Manager Explorer e visualizem, anexem e desanexem políticas do IAM para uso com a Quick Setup e o Systems Manager Explorer. Essas permissões são necessárias para que a Quick Setup possa criar os perfis necessários para algumas de suas operações de configuração.
- `s3`: permite que as entidades principais recuperem informações sobre objetos e excluam objetos dos buckets do Amazon S3, na conta da entidade principal, que são usados especificamente em operações de configuração da Quick Setup. Isso é necessário para que os objetos do S3 que não são mais exigidos após a configuração possam ser removidos.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWSQuickSetupSSMManageResourcesExecutionPolicy no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AWSQuickSetupSSMLifecycleManagementExecutionPolicy

A política `AWSQuickSetupSSMLifecycleManagementExecutionPolicy` concede permissões administrativas que permitem à Quick Setup executar um recurso personalizado do AWS CloudFormation em eventos de ciclo de vida durante a implantação da Quick Setup no Systems Manager.

Essa política pode ser anexada a suas identidades do IAM. O Systems Manager também anexa essa política a um perfil de serviço que permite que o Systems Manager realize ações em seu nome.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `ssm`: permite que as entidades principais obtenham informações sobre execuções do Automation e iniciem execuções de automação para configurar determinadas operações da Quick Setup.
- `iam`: permite que as entidades principais passem perfis do IAM para configurar determinados recursos da Quick Setup.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWSQuickSetupSSMLifecycleManagementExecutionPolicy no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AWSQuickSetupSSMDeploymentRolePolicy

A política gerenciada `AWSQuickSetupSSMDeploymentRolePolicy` concede permissões administrativas que permitem à Quick Setup criar recursos que são usados durante o processo de integração do Systems Manager.

Embora seja possível anexar essa política a suas entidades do IAM, fazer isso não é recomendado. A Quick Setup cria entidades que anexam essa política a um perfil de serviço que permite ao Systems Manager realizar ações em seu nome.

Essa política não está relacionada à política `SSMQuickSetupRolePolicy` que é usada para fornecer permissões para o perfil vinculado ao serviço `AWSServiceRoleForSSMQuickSetup`.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `ssm`: permite que as entidades principais gerenciem associações para determinados recursos que são criados usando modelos do AWS CloudFormation e um conjunto específico de documentos do SSM, gerenciem perfis e políticas de perfis usados para diagnosticar e corrigir nós gerenciados por meio de modelos do AWS CloudFormation e anexem e excluam políticas para eventos de ciclo de vida da Quick Setup
- `iam`: permite que as entidades principais marquem perfis e passem permissões de perfis para o serviço do Systems Manager e do Lambda, e passem permissões de perfis para as operações de diagnóstico.
- `lambda`: permite que as entidades principais marquem e gerenciem funções do ciclo de vida do Quick Setup na conta da entidade principal usando modelos do AWS CloudFormation.
- `cloudformation`: permite que as entidades principais leiam as informações de AWS CloudFormation. Isso é necessário para que a Quick Setup possa coletar dados sobre as pilhas do AWS CloudFormation usadas para gerenciar o estado dos recursos e as operações do conjunto de pilhas do CloudFormation.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWSQuickSetupSSMDeploymentRolePolicy no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AWSQuickSetupSSMDeploymentS3BucketRolePolicy

A política `AWSQuickSetupSSMDeploymentS3BucketRolePolicy` concede permissões para listar todos os buckets do S3 em uma conta e para gerenciar e recuperar informações sobre buckets específicos na conta da entidade principal que são gerenciados via modelos do AWS CloudFormation.

Você pode anexar `AWSQuickSetupSSMDeploymentS3BucketRolePolicy` às suas entidades do IAM. O Systems Manager também anexa essa política a um perfil de serviço que permite que o Systems Manager realize ações em seu nome.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `s3`: permite que as entidades principais listem todos os buckets do S3 em uma conta e gerenciem e recuperem informações sobre buckets específicos na conta da entidade principal que são gerenciados via modelos do AWS CloudFormation.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWSQuickSetupSSMDeploymentS3BucketRolePolicy no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada da AWS: AWSQuickSetupEnableDHMCExecutionPolicy

Essa política concede permissões administrativas que permitem que as entidades principais executem o runbook do Automation `AWSQuickSetupType-EnableDHMC` que habilita a configuração de gerenciamento de host padrão. A opção Configuração de gerenciamento de hosts padrão permite que o Systems Manager gerencie automaticamente as instâncias do Amazon EC2 na forma de instâncias gerenciadas. Uma instância gerenciada é uma instância do EC2 que foi configurada para uso com o Systems Manager. Essa política também concede permissões para criar perfis do IAM que são especificados nas configurações do serviço Systems Manager como os perfis padrão para SSM Agent.

Você pode anexar `AWSQuickSetupEnableDHMCExecutionPolicy` às suas entidades do IAM. O Systems Manager também anexa essa política a um perfil de serviço que permite que o Systems Manager realize ações em seu nome.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `ssm`: permite que as entidades principais atualizem e obtenham informações sobre as configurações do serviço Systems Manager.
- `iam`: permite que as entidades principais criem e recuperem informações sobre os perfis do IAM para operações da Quick Setup.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWSQuickSetupEnableDHMCExecutionPolicy no AWS Managed Policy Reference Guide.

Política gerenciada da AWS: AWSQuickSetupEnableAREXExecutionPolicy

Essa política concede permissões administrativas que permitem que o Systems Manager execute o runbook do Automation `AWSQuickSetupType-EnableAREX`, o qual permite o uso do Explorador de recursos da AWS com o Systems Manager. O Explorador de Recursos possibilita a visualização de recursos em sua conta com uma experiência de pesquisa semelhante a um mecanismo de pesquisa da Internet. A política também concede permissões para gerenciar índices e visualizações do Explorador de Recursos.

Você pode anexar `AWSQuickSetupEnableAREXExecutionPolicy` às suas entidades do IAM. O Systems Manager também anexa essa política a um perfil de serviço que permite que o Systems Manager realize ações em seu nome.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `iam`: permite que as entidades principais criem um perfil vinculado ao serviço no serviço AWS Identity and Access Management (IAM).
- `resource-explorer-2`: permite que as entidades principais recuperem informações sobre visualizações e índices do Explorador de Recursos, criem visualizações e índices do Explorador de recursos e alterem o tipo de índice dos índices exibidos na Quick Setup.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWSQuickSetupEnableAREXExecutionPolicy no AWS Managed Policy Reference Guide.

Política gerenciada pela AWS: AWSQuickSetupManagedInstanceProfileExecutionPolicy

Essa política concede permissões administrativas que permitem ao Systems Manager criar um perfil de instância do IAM padrão para a ferramenta Quick Setup e anexá-lo a instâncias do Amazon EC2 que ainda não têm um perfil de instância anexado. A política também concede ao Systems Manager a capacidade de anexar permissões aos perfis de instância existentes. Isso é feito para garantir que as permissões necessárias para o Systems Manager se comunicar com o SSM Agent em instâncias do EC2 estejam em vigor.

Você pode anexar `AWSQuickSetupManagedInstanceProfileExecutionPolicy` às suas entidades do IAM. O Systems Manager também anexa essa política a um perfil de serviço que permite que o Systems Manager realize ações em seu nome.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `ssm`: permite que as entidades principais iniciem fluxos de trabalho de automação associados aos processos da Quick Setup.
- `ec2`: permite que as entidades principais anexem perfis de instância do IAM às instâncias do EC2 que são gerenciadas pela Quick Setup.
- `iam`: permite que as entidades principais criem, atualizem e recuperem informações sobre perfis do IAM que são usadas em processos da Quick Setup, criem perfis de instância do IA e anexem a política gerenciada `AmazonSSMManagedInstanceCore` aos perfis de instância do IAM.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWSQuickSetupManagedInstanceProfileExecutionPolicy no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AWSQuickSetupFullAccess

Essa política concede permissões administrativas que permitem acesso total às ações e dados da API AWS Systems Manager Quick Setup no AWS Management Console e nos AWS SDKs, bem

como acesso limitado a outros recursos do AWS service (Serviço da AWS) necessários para as operações da Quick Setup.

É possível anexar a política `AWSQuickSetupFullAccess` às identidades do IAM.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `ssm`: permite que as entidades principais habilitem o Explorer, realizem operações de sincronização de dados de recursos no State Manager e executem operações usando documentos de comandos do SSM e runbooks do Automation.

  Explorer, State Manager, Documentos e Automation são todos ferramentas do Systems Manager.
- `cloudformation`: permite que as entidades principais realizem as operações do AWS CloudFormation necessárias para provisionar recursos em Regiões da AWS e Contas da AWS.
- `ec2`: permite que as entidades principais selecionem os parâmetros necessários para uma determinada configuração e forneçam validação no AWS Management Console.
- `iam`: permite que as entidades principais criem os perfis de serviço necessários e os perfis vinculadas ao serviço para as operações da Quick Setup.
- `organizations`: permite que as entidades principais leiam o status das contas em uma organização do AWS Organizations, recuperem a estrutura de uma organização, habilitem acesso confiável e registrem uma conta de administrador delegado na conta de gerenciamento.
- `resource-groups`: permite que as entidades principais selecionem os parâmetros necessários para uma determinada configuração e forneçam validação no AWS Management Console.
- `s3`: permite que as entidades principais selecionem os parâmetros necessários para uma determinada configuração e forneçam validação no AWS Management Console.
- `ssm-quicksetup`: permite que as entidades principais realizem ações somente leitura na Quick Setup.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWSQuickSetupFullAccess no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AWSQuickSetupReadOnlyAccess

Essa política concede permissões somente leitura que permitem que as entidades principais visualizem dados e relatórios do AWS Systems Manager Quick Setup, incluindo informações de outros recursos do AWS service (Serviço da AWS) necessários para as operações da Quick Setup.

É possível anexar a política `AWSQuickSetupReadOnlyAccess` às identidades do IAM.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `ssm`: permite que as entidades principais leiam documentos de comandos do SSM e runbooks do Automation e recuperem o status de execuções da associação do State Manager.
- `cloudformation`: permite que as entidades principais iniciem as operações necessárias para recuperar o status das implantações do AWS CloudFormation.
- `organizations`: permite que as entidades principais leiam o status das contas em uma organização do AWS Organizations.
- `ssm-quicksetup`: permite que as entidades principais realizem ações somente leitura na Quick Setup.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWSQuickSetupReadOnlyAccess no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AWSQuickSetupManageJITNAResourcesExecutionPolicy

A política gerenciada `AWSQuickSetupManageJITNAResourcesExecutionPolicy` permite que o Quick Setup, uma ferramenta no Systems Manager, configure o acesso a nós just-in-time.

Você pode anexar `AWSQuickSetupManageJITNAResourcesExecutionPolicy` às suas entidades do IAM. O Systems Manager também anexa essa política a um perfil de serviço que permite que o Systems Manager realize ações em seu nome.

Esta política concede permissões administrativas que possibilitam que o Systems Manager crie recursos associados ao acesso a nós just-in-time.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `ssm`: permite que as entidades principais obtenham e atualizem a configuração do serviço que especifica o provedor de identidade para acesso a nós just-in-time.
- `iam`: permite que as entidades principais criem, marquem e obtenham perfis, anexem políticas de perfil para políticas gerenciadas de acesso a nós just-in-time e criem funções vinculadas ao serviço para notificações e acesso a nós just-in-time.

Para visualizar mais detalhes sobre a política, incluindo a versão mais recente do documento de política JSON, consulte AWSQuickSetupManageJITNAResourcesExecutionPolicy no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AWSQuickSetupJITNADeploymentRolePolicy

A política gerenciada `AWSQuickSetupJITNADeploymentRolePolicy` permite que o Quick Setup implante o tipo de configuração necessário para configurar o acesso a nós just-in-time.

Você pode anexar `AWSQuickSetupJITNADeploymentRolePolicy` às suas entidades do IAM. O Systems Manager também anexa essa política a um perfil de serviço que permite que o Systems Manager realize ações em seu nome.

Esta política concede permissões administrativas que possibilitam que o Systems Manager crie recursos associados ao acesso a nós just-in-time.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `cloudformation`: permite que as entidades principais criem, atualizem, excluam e leiam pilhas do AWS CloudFormation.
- `ssm`: permite que as entidades principais criem, excluam, atualizem e leiam associações do State Manager que são chamadas pelo AWS CloudFormation.
- `iam`: permite que as entidades principais criem, excluam, leiam e marquem perfis do IAM que são chamados pelo AWS CloudFormation.

Para visualizar mais detalhes sobre a política, incluindo a versão mais recente do documento de política JSON, consulte AWSQuickSetupJITNADeploymentRolePolicy no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AWSSystemsManagerJustInTimeAccessServicePolicy

A política gerenciada `AWSSystemsManagerJustInTimeAccessServicePolicy` fornece permissões aos recursos gerenciados pela AWS ou usados pelo recurso de acesso a nós just-in-time do Systems Manager.

Não é possível anexar a `AWSSystemsManagerJustInTimeAccessServicePolicy` às entidades do IAM. Essa política é anexada a uma função vinculada ao serviço que permite que o Systems Manager realize ações em seu nome. Para obter mais informações, consulte Uso de perfis para habilitar o acesso a nós just-in-time.

Esta política concede permissões administrativas que permitem o acesso aos recursos associados ao acesso a nós just-in-time.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `ssm`: permite que as entidades principais criem, descrevam e atualizem o OpsItems, descrevam sessões e listem tags para nós gerenciados.
- `ssm-guiconnect`: permite que as entidades principais listem conexões.
- `identitystore`: permite obter IDs de usuários e grupos, descrever usuários e listar associações de grupos.
- `sso-directory`: permite que as entidades principais descrevam usuários e determinem se um usuário é membro de um grupo.
- `sso`: permite que as entidades principais descrevam as regiões registradas e listem instâncias e associações de diretórios.
- `cloudwatch`: permite que as entidades principais coloquem dados de métricas do namespace `AWS/SSM/JustInTimeAccess`.
- `ec2`: permite que as entidades principais descrevam as tags.

Para visualizar mais detalhes sobre a política, incluindo a versão mais recente do documento de política JSON, consulte AWSSystemsManagerJustInTimeAccessServicePolicy no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AWSSystemsManagerJustInTimeAccessTokenPolicy

A política gerenciada `AWSSystemsManagerJustInTimeAccessTokenPolicy` permite que o Systems Manager gere tokens de acesso usados para acesso a nós just-in-time.

Você pode anexar `AWSSystemsManagerJustInTimeAccessTokenPolicy` às entidades do IAM.

Esta política concede permissões administrativas que possibilitam que o Systems Manager inicie sessões e gere tokens de acesso para acesso a nós just-in-time.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `ssm`: permite que as entidades principais enviem comandos, listem invocações de comandos e iniciem, retomem e encerrem sessões do Session Manager.
- `ssm-guiconnect`: permite que as entidades principais iniciem, descrevam e encerrem conexões RDP do Systems Manager GUI Connect.
- `kms`: permite que as entidades principais criem concessões e gerem dados de chave.
- `sso`: permite que as entidades principais listem associações de diretórios.
- `identitystore`: permite que as entidades principais descrevam um usuário.

Para visualizar mais detalhes sobre a política, incluindo a versão mais recente do documento de política JSON, consulte AWSSystemsManagerJustInTimeAccessTokenPolicy no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AWSSystemsManagerJustInTimeAccessTokenSessionPolicy

A política gerenciada `AWSSystemsManagerJustInTimeAccessTokenSessionPolicy` possibilita que o Systems Manager aplique permissões com escopo reduzido a um token de acesso a nós just-in-time.

Você pode anexar `AWSSystemsManagerJustInTimeAccessTokenSessionPolicy` às entidades do IAM.

Esta política concede permissões administrativas que possibilitam que o Systems Manager reduza o escopo das permissões para tokens de acesso a nós just-in-time.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `ssm`: permite que as entidades principais iniciem sessões do Session Manager usando o documento `SSM-SessionManagerRunShell`. Além disso, quando chamadas primeiro via `ssm-guiconnect`, iniciam sessões usando o documento `AWS-StartPortForwardingSession`, listam invocações de comandos e enviam comandos usando o documento `AWSSSO-CreateSSOUser`.
- `ssm-guiconnect`: permite que as entidades principais cancelem, obtenham e iniciem conexões em todos os recursos.
- `kms`: permite que as entidades principais criem concessões e gerem chaves de dados para chaves marcadas com `SystemsManagerJustInTimeNodeAccessManaged` quando chamadas via `ssm-guiconnect` por meio de um serviço da AWS.
- `sso`: permite que as entidades principais listem associações de diretórios quando chamadas via `ssm-guiconnect`.
- `identitystore`: permite que as entidades principais descrevam um usuário quando chamadas via `ssm-guiconnect`.

Para visualizar mais detalhes sobre a política, incluindo a versão mais recente do documento de política JSON, consulte AWSSystemsManagerJustInTimeAccessTokenSessionPolicy no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AWSSystemsManagerJustInTimeNodeAccessRolePropagationPolicy

A política gerenciada `AWSSystemsManagerJustInTimeNodeAccessRolePropagationPolicy` permite que o Systems Manager compartilhe políticas de negação de acesso da conta de administrador delegado para contas de membros e replique-as em várias Regiões da AWS.

Você pode anexar `AWSSystemsManagerJustInTimeNodeAccessRolePropagationPolicy` às entidades do IAM.

Essa política fornece as permissões administrativas necessárias para o Systems Manager compartilhar e criar políticas de negação de acesso. Isso garante que as políticas de negação de acesso sejam aplicadas a todas as contas em uma organização do AWS Organizations e regiões configuradas para acesso a nós just-in-time.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `ssm`: permite que as entidades principais gerenciem documentos do SSM e políticas de recursos.
- `ssm-quicksetup`: permite que as entidades principais leiam os gerenciadores de configurações do Quick Setup.
- `organizations`: permite que as entidades principais listem os detalhes sobre uma organização do AWS Organizations e administradores delegados.
- `ram`: permite que as entidades principais criem, marquem e descrevam compartilhamentos de recursos.
- `iam`: permite que as entidades principais descrevam um perfil de serviço.

Para visualizar mais detalhes sobre a política, incluindo a versão mais recente do documento de política JSON, consulte AWSSystemsManagerJustInTimeNodeAccessRolePropagationPolicy no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AWSSystemsManagerNotificationsServicePolicy

A política gerenciada `AWSSystemsManagerNotificationsServicePolicy` permite que o Systems Manager envie notificações por e-mail de solicitações de acesso a nós just-in-time para aprovadores de solicitações de acesso.

Não é possível anexar a `AWSSystemsManagerJustInTimeAccessServicePolicy` às entidades do IAM. Essa política é anexada a uma função vinculada ao serviço que permite que o Systems Manager realize ações em seu nome. Para obter mais informações, consulte Uso de perfis para enviar notificações de solicitação de acesso a nós just-in-time.

Essa política concede permissões administrativas que possibilitam que o Systems Manager envie notificações por e-mail de solicitações de acesso a nós just-in-time para aprovadores de solicitações de acesso.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `identitystore`: permite que as entidades principais listem e descrevam usuários e associações de grupos.
- `sso`: permite que as entidades principais listem instâncias, diretórios e descrevam regiões registradas.

- `sso-directory`: permite que as entidades principais descrevam usuários e listem membros em um grupo.
- `iam`: permite que as entidades principais obtenham informações sobre perfis.

Para visualizar mais detalhes sobre a política, incluindo a versão mais recente do documento de política JSON, consulte AWSSystemsManagerNotificationsServicePolicy no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AWS-SSM-Automation-DiagnosisBucketPolicy

A política gerenciada `AWS-SSM-Automation-DiagnosisBucketPolicy` fornece permissões para diagnosticar problemas com nós que interagem com os serviços do AWS Systems Manager, permitindo o acesso aos buckets do S3 que são usados para diagnóstico e correção de problemas.

É possível anexar a política `AWS-SSM-Automation-DiagnosisBucketPolicy` às identidades do IAM. O Systems Manager também anexa essa política a um perfil do IAM que permite que o Systems Manager execute ações de diagnóstico em seu nome.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `s3`: concede às entidades principais acesso de leitura e gravação a objetos em um bucket do Amazon S3

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWS-SSM-Automation-DiagnosisBucketPolicy no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AWS-SSM-RemediationAutomation-OperationalAccountAdministrationRolePolicy

A política gerenciada `AWS-SSM-RemediationAutomation-OperationalAccountAdministrationRolePolicy` fornece permissões para que uma conta operacional diagnostique problemas com os nós fornecendo para isso permissões específicas da organização.

É possível anexar a `AWS-SSM-RemediationAutomation-OperationalAccountAdministrationRolePolicy` a suas identidades do IAM. O Systems

Manager também anexa essa política a um perfil do IAM que permite que o Systems Manager execute ações de diagnóstico em seu nome.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `organizations`: permite que as entidades principais listem uma raiz da organização e obtenham contas-membro para determinar as contas de destino.
- `sts`: permite que as entidades principais assumam perfis de execução de correção para executar documentos do SSM Automation em contas e regiões diversas na mesma organização.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWS-SSM-RemediationAutomation-OperationalAccountAdministrationRolePolicy no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AWS-SSM-DiagnosisAutomation-OperationalAccountAdministrationRolePolicy

A política gerenciada `AWS-SSM-DiagnosisAutomation-OperationalAccountAdministrationRolePolicy` fornece permissões para que uma conta operacional diagnostique problemas com os nós fornecendo para isso permissões específicas da organização.

É possível anexar a política `AWS-SSM-DiagnosisAutomation-OperationalAccountAdministrationRolePolicy` às identidades do IAM. O Systems Manager também anexa essa política a um perfil do IAM que permite que o Systems Manager execute ações de diagnóstico em seu nome.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `organizations`: permite que as entidades principais listem uma raiz da organização e obtenham contas-membro para determinar as contas de destino.
- `sts`: permite que as entidades principais assumam perfis de execução de diagnóstico para executar documentos do SSM Automation em contas e regiões diversas na mesma organização.

Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AWS-SSM-DiagnosisAutomation-OperationalAccountAdministrationRolePolicy no Guia de referência de políticas gerenciadas pela AWS.

Atualizações do Systems Manager para políticas gerenciadas pela AWS

Na tabela a seguir, veja detalhes sobre atualizações em políticas gerenciadas pela AWS para o Systems Manager desde que esse serviço começou a rastrear essas alterações em 12 de março de 2021. Para obter informações sobre outras políticas gerenciadas para o serviço Systems Manager, consulte Políticas gerenciadas adicionais para o Systems Manager mais adiante neste tópico. Para obter alertas automáticos sobre alterações feitas nesta página, inscreva-se no feed RSS na página Histórico do documento do Systems Manager.

Alteração	Descrição	Data
AmazonSSMAutomationRole: atualização da documentação	O Systems Manager adicionou uma documentação abrangente para a política `AmazonSSMAutomationRole` existente, que fornece permissões para o serviço do Systems Manager Automation executar atividades definidas nos runbooks de automação.	15 de julho de 2025
AWS-SSM-RemediationAutomation-ExecutionRolePolicy: atualização de uma política existente	O Systems Manager atualizou a política gerenciada para melhorar a postura de segurança da API ssm:StartAutomationExecution exigindo permissões para os tipos de recursos de “document” e “automation-execution”. A política atualizada fornece	16 de julho de 2025

Alteração	Descrição	Data
	permissões mais abrangentes e detalhadas para a execução da automação de correção incluindo descrições aprimoradas dos recursos de correção de rede, permissões mais específicas de criação de endpoints da Amazon VPC, permissões detalhadas de gerenciamento de grupos de segurança e controles aprimorados de marcação de recursos para operações de correção.	
AWS-SSM-RemediationAutomation-AdministrationRolePolicy: atualização de uma política existente	O Systems Manager atualizou a política gerenciada para oferecer suporte a melhorias na autorização da API para operações de automação de correção. A política atualizada aprimora as permissões para a execução de atividades definidas nos documentos de automação, com controles de segurança aprimorados e padrões de acesso a recursos para fluxos de trabalho de correção.	16 de julho de 2025

Alteração	Descrição	Data
AWS-SSM-DiagnosisAutomation-ExecutionRolePolicy: atualização de uma política existente	O Systems Manager atualizou a política gerenciada para fornecer permissões mais detalhadas e precisas para a execução da automação de diagnósticos. A política atualizada inclui descrições aprimoradas para acesso aos recursos do Amazon EC2 e da Amazon VPC, permissões de automação do SSM mais específicas e descrições aprimoradas das permissões do AWS KMS e do IAM com restrições de recursos adequadas.	16 de julho de 2025
AWS-SSM-DiagnosisAutomation-AdministrationRolePolicy: atualização de uma política existente	O Systems Manager atualizou a política gerenciada para fornecer permissões e condições de segurança mais específicas para operações de automação de diagnóstico. A política atualizada fornece controles de segurança aprimorados para uso de chaves do AWS KMS, acesso ao bucket do Amazon S3 e suposições de perfis, com condições mais rígidas baseadas em recursos e restrições em nível de conta.	16 de julho de 2025

Alteração	Descrição	Data
AWSQuickSetupDeploymentRolePolicy: atualização em uma política	O Systems Manager adicionou permissões à política gerenciada `AWSQuickSetupDeploymentRolePolicy` para acesso ao runbook AWSQuickSetupType-ManageInstanceProfile de propriedade da Amazon. Essa permissão permite que a Quick Setup crie associações utilizando a política gerenciada em vez de políticas em linha.	14 de julho de 2025
AWSSystemsManagerJustInTimeNodeAccessRolePropagationPolicy: atualização de uma política	O Systems Manager adicionou permissões para permitir que o Systems Manager marque um recurso compartilhado pelo AWS Resource Access Manager para acesso a nós just-in-time.	30 de abril de 2025
AWSQuickSetupManageJITNAResourcesExecutionPolicy: atualização em uma política	O Systems Manager adicionou permissões que possibilitam que o Systems Manager marque perfis do IAM criados para acesso a nós just-in-time.	30 de abril de 2025
AWSSystemsManagerJustInTimeAccessTokenSessionPolicy: nova política	O Systems Manager adicionou uma nova política para possibilitar que o Systems Manager aplique permissões com escopo reduzido a um token de acesso a nós just-in-time.	30 de abril de 2025

Alteração	Descrição	Data
AWSSystemsManagerNotificationsServicePolicy: nova política	O Systems Manager adicionou uma nova política que possibilita que o Systems Manager envie notificações por e-mail de solicitações de acesso a nós just-in-time para aprovadores de solicitações de acesso.	30 de abril de 2025
AWSSystemsManagerJustInTimeNodeAccessRolePropagationPolicy: nova política	O Systems Manager adicionou uma nova política que possibilita que o Systems Manager replique políticas de aprovação em diferentes regiões.	30 de abril de 2025
AWSSystemsManagerJustInTimeAccessTokenPolicy: nova política	O Systems Manager adicionou uma nova política que possibilita que o Systems Manager gere tokens de acesso usados para acesso a nós just-in-time.	30 de abril de 2025
AWSSystemsManagerJustInTimeAccessServicePolicy: nova política	O Systems Manager adicionou uma nova política para fornecer permissões aos recursos gerenciados pela AWS ou usados pelo recurso de acesso a nós just-in-time do Systems Manager.	30 de abril de 2025

Alteração	Descrição	Data
AWSQuickSetupManageJITNAResourcesExecutionPolicy: nova política	O Systems Manager adicionou uma nova política que possibilita que o Quick Setup, uma ferramenta no Systems Manager, crie os perfis do IAM necessários para o acesso a nós just-in-time.	30 de abril de 2025
AWSQuickSetupJITNADeploymentRolePolicy: nova política	O Systems Manager adicionou uma nova política que fornece permissões que possibilitam que o Quick Setup implante o tipo de configuração necessário para configurar o acesso a nós just-in-time.	30 de abril de 2025
AWSSystemsManagerJustInTimeNodeAccessRolePropagationPolicy: atualização de uma política	O Systems Manager adicionou permissões para permitir que o Systems Manager marque um recurso compartilhado pelo AWS Resource Access Manager para acesso a nós just-in-time.	30 de abril de 2025
AWSQuickSetupManageJITNAResourcesExecutionPolicy: atualização de uma política	O Systems Manager adicionou permissões que possibilitam que o Systems Manager marque perfis do IAM criados para acesso a nós just-in-time.	30 de abril de 2025

Alteração	Descrição	Data
AWSSystemsManagerJustInTimeAccessTokenSessionPolicy: nova política	O Systems Manager adicionou uma nova política para possibilitar que o Systems Manager aplique permissões com escopo reduzido a um token de acesso a nós just-in-time.	30 de abril de 2025
AWSSystemsManagerNotificationsServicePolicy: nova política	O Systems Manager adicionou uma nova política que possibilita que o Systems Manager envie notificações por e-mail de solicitações de acesso a nós just-in-time para aprovadores de solicitações de acesso.	30 de abril de 2025
AWSSystemsManagerJustInTimeNodeAccessRolePropagationPolicy: nova política	O Systems Manager adicionou uma nova política que possibilita que o Systems Manager replique políticas de aprovação em diferentes regiões.	30 de abril de 2025
AWSSystemsManagerJustInTimeAccessTokenPolicy: nova política	O Systems Manager adicionou uma nova política que possibilita que o Systems Manager gere tokens de acesso usados para acesso a nós just-in-time.	30 de abril de 2025

Alteração	Descrição	Data
AWSSystemsManagerJustInTimeAccessServicePolicy: nova política	O Systems Manager adicionou uma nova política para fornecer permissões aos recursos gerenciados pela AWS ou usados pelo recurso de acesso a nós just-in-time do Systems Manager.	30 de abril de 2025
AWSQuickSetupManageJITNAResourcesExecutionPolicy: nova política	O Systems Manager adicionou uma nova política que possibilita que o Quick Setup, uma ferramenta no Systems Manager, crie os perfis do IAM necessários para o acesso a nós just-in-time.	30 de abril de 2025
AWSQuickSetupJITNADeploymentRolePolicy: nova política	O Systems Manager adicionou uma nova política que fornece permissões que possibilitam que o Quick Setup implante o tipo de configuração necessário para configurar o acesso a nós just-in-time.	30 de abril de 2025
`AWS-SSM-DiagnosisAutomation-OperationalAccountAdministrationRolePolicy` – Nova política	O Systems Manager adicionou uma nova política que fornece permissões para que uma conta operacional diagnostique problemas com os nós fornecendo para isso permissões específicas da organização.	21 de novembro de 2024

Alteração	Descrição	Data
AWS-SSM-RemediationAutomation-OperationalAccountAdministrationRolePolicy – Nova política	O Systems Manager adicionou uma nova política que fornece permissões para que uma conta operacional diagnosti que problemas com os nós fornecendo para isso permissões específicas da organização.	21 de novembro de 2024
AWS-SSM-Automation-DiagnosisBucketPolicy – Nova política	O Systems Manager adicionou uma nova política para oferecer suporte ao início de fluxos de trabalho do Automation que diagnosti cam problemas com nós gerenciados em contas e regiões específicas.	21 de novembro de 2024
AmazonSSMServiceRolePolicy : atualização para uma política existente	O Systems Manager adicionou novas permissões para permitir ao Explorador de recursos da AWS coletar detalhes sobre as instâncias do Amazon EC2 e a exibição dos resultados em widgets no novo painel do Systems Manager.	21 de novembro de 2024

Alteração	Descrição	Data
SSMQuickSetupRolePolicy : atualização para uma política existente	O Systems Manager atualizou a política gerenciada SSMQuickSetupRolePolicy . Essas atualizações permitem que o perfil vinculado ao serviço associado gerencie as AWSServiceRoleForSSMQuickSetup sincronizações de dados de recursos.	21 de novembro de 2024
AWS-SSM-DiagnosisAutomation-AdministrationRolePolicy – Nova política	O Systems Manager adicionou uma nova política para oferecer suporte ao início de fluxos de trabalho do Automation que diagnosticam problemas com nós gerenciados em uma conta específica e regiões.	21 de novembro de 2024
AWS-SSM-DiagnosisAutomation-ExecutionRolePolicy – Nova política	O Systems Manager adicionou uma nova política para oferecer suporte ao início de fluxos de trabalho do Automation que diagnosticam problemas com nós gerenciados em uma conta e uma região específica.	21 de novembro de 2024

Alteração	Descrição	Data
AWS-SSM-RemediationAutomation-AdministrationRolePolicy – Nova política	O Systems Manager adicionou uma nova política para oferecer suporte ao início de fluxos de trabalho do Automation que corrigem problemas com nós gerenciados em contas e regiões específicas.	21 de novembro de 2024
AWS-SSM-RemediationAutomation-ExecutionRolePolicy – Nova política	O Systems Manager adicionou uma nova política para oferecer suporte ao início de fluxos de trabalho do Automation que corrigem problemas com nós gerenciados em uma conta e uma região específicas.	21 de novembro de 2024
AWSQuickSetupSSMDeploymentRolePolicy: atualização de uma política	O Systems Manager adicionou permissões para possibilitar que o Systems Manager marque perfis do IAM e do Lambda criados para o console unificado.	7 de maio de 2025
AWSQuickSetupSSMManageResourcesExecutionPolicy – Nova política	O Systems Manager adicionou uma nova política para oferecer suporte à execução de uma operação na Quick Setup que cria perfis do IAM para associações da Quick Setup, que por sua vez são criadas por uma implantação da `AWSQuickSetupType-SSM` .	21 de novembro de 2024

Alteração	Descrição	Data
AWSQuickSetupSSMLifecycleManagementExecutionPolicy – Nova política	O Systems Manager adicionou uma nova política para oferecer suporte à execução pela Quick Setup de um recurso do AWS CloudFormation personalizado em eventos de ciclo de vida durante uma implantação da Quick Setup.	21 de novembro de 2024
AWSQuickSetupSSMDeploymentRolePolicy – Nova política	O Systems Manager adicionou uma nova política para oferecer suporte à concessão de permissões administrativas que permitem à Quick Setup criar recursos que são usados durante o processo de integração do Systems Manager.	21 de novembro de 2024
AWSQuickSetupSSMDeploymentS3BucketRolePolicy – Nova política	O Systems Manager adicionou uma nova política para oferecer suporte ao gerenciamento e à recuperação de informações sobre buckets específicos na conta da entidade principal que são gerenciados via modelos do AWS CloudFormation	21 de novembro de 2024

Alteração	Descrição	Data
AWSQuickSetupEnableDHMCExecutionPolicy – Nova política	O Systems Manager está introduzindo uma nova política para permitir que a Quick Setup crie um perfil do IAM que, por sua vez, usa a AmazonSSMManagedEC2InstanceDefaultPolicy existente. Essa política contém todas as permissões necessárias para o SSM Agent se comunicar com o serviço Systems Manager. A nova política também permite modificações nas configurações do serviço Systems Manager.	21 de novembro de 2024
AWSQuickSetupEnableAREXExecutionPolicy – Nova política	O Systems Manager adicionou uma nova política Quick Setup para permitir a criação de um perfil vinculado ao serviço para o Explorador de recursos da AWS a fim de permitir o acesso a visualizações do Explorador de Recursos e a índices agregadores.	21 de novembro de 2024

Alteração	Descrição	Data
AWSQuickSetupManagedInstanceProfileExecutionPolicy – Nova política	O Systems Manager adicionou uma nova política para permitir que a Quick Setup crie um perfil de instância da Quick Setup padrão e anexe-o a qualquer instância do Amazon EC2 que não tenha um perfil de instância associado. Essa nova política também permite à Quick Setup anexar permissões aos perfis existentes para garantir que todas as permissões necessárias do Systems Manager tenham sido concedidas.	21 de novembro de 2024
AWSQuickSetupFullAccess – Nova política	O Systems Manager adicionou uma nova política para permitir que as entidades tenham acesso total às ações e dados da API da AWS Systems Manager Quick Setup no AWS Management Console e nos AWS SDKs, bem como acesso limitado a outros recursos do AWS service (Serviço da AWS) necessários para as operações da Quick Setup.	21 de novembro de 2024

Alteração	Descrição	Data
AWSQuickSetupReadOnlyAccess – Nova política	O Systems Manager adicionou uma nova política para conceder permissões somente leitura que permitem que as entidades principais visualizem dados e relatórios do AWS Systems Manager Quick Setup, incluindo informações de outros recursos do AWS service (Serviço da AWS) necessários para as operações da Quick Setup.	21 de novembro de 2024
SSMQuickSetupRolePolicy : atualização para uma política existente	O Systems Manager adicionou novas permissões para permitir que Quick Setup verifique a integridade de conjuntos de pilhas adicionais do AWS CloudFormation que ele criou.	13 de agosto de 2024
AmazonSSMManagedEC2InstanceDefaultPolicy : atualização para uma política existente	Systems Manager adicionou IDs de declaração (Sids) à política JSON para `AmazonSSMManagedEC2InstanceDefaultPolicy` . Esses Sids fornecem descrições em linha do propósito de cada declaração de política.	18 de julho de 2024

Alteração	Descrição	Data
SSMQuickSetupRolePolicy – Nova política	O Systems Manager adicionou uma nova política para que o Quick Setup possa verificar a integridade dos recursos implantados e corrigir instâncias que se afastaram da configuração original.	3 de julho de 2024
AWSQuickSetupDeploymentRolePolicy – Nova política	O Systems Manager adicionou uma nova política para oferecer suporte a vários tipos de configuração da Configuração Rápida que criam perfis e automações do IAM, que, por sua vez, configuram serviços e recursos da Amazon Web Services usados com frequência com as melhores práticas recomendadas.	3 de julho de 2024
AWSQuickSetupPatchPolicyDeploymentRolePolicy : nova política	O Systems Manager adicionou uma nova política para que a Quick Setup possa criar recursos associados às configurações da Patch Manager da política de patch do Quick Setup.	3 de julho de 2024
AWSQuickSetupPatchPolicyBaselineAccess: nova política	O Systems Manager adicionou uma nova política para permitir que a Quick Setup acesse as listas de referência de patches no Patch Manager com permissões somente de leitura.	3 de julho de 2024

Alteração	Descrição	Data
AWSSystemsManagerEnableExplorerExecutionPolicy: nova política	O Systems Manager adicionou uma nova política para permitir que a Quick Setup conceda permissões administrativas para habilitar o Explorer.	3 de julho de 2024
AWSSystemsManagerEnableConfigRecordingExecutionPolicy: nova política	O Systems Manager adicionou uma nova política para permitir que a Quick Setup habilite e configure a gravação de configuração do AWS Config.	3 de julho de 2024
AWSQuickSetupDevOpsGuruPermissionsBoundary: nova política	O Systems Manager adicionou uma nova política para permitir que a Quick Setup habilite e configure o Amazon DevOps Guru.	3 de julho de 2024
AWSQuickSetupDistributorPermissionsBoundary: nova política	O Systems Manager adicionou uma nova política para permitir que a Quick Setup habilite e configure o Distributor, uma ferramenta do AWS Systems Manager.	3 de julho de 2024
AWSQuickSetupSSMHostMgmtPermissionsBoundary: nova política	O Systems Manager adicionou uma nova política para permitir que a Quick Setup habilite e configure as ferramentas do Systems Manager para gerenciar com segurança as instâncias do Amazon EC2.	3 de julho de 2024

Alteração	Descrição	Data
AWSQuickSetupPatchPolicyPermissionsBoundary: nova política	O Systems Manager adicionou uma nova política para permitir que a Quick Setup habilite e configure políticas de patch no Patch Manager, uma ferramenta do AWS Systems Manager.	3 de julho de 2024
AWSQuickSetupSchedulerPermissionsBoundary: nova política	O Systems Manager adicionou uma nova política para permitir que a Quick Setup habilite e configure operações programadas em instâncias do Amazon EC2 e outros recursos.	3 de julho de 2024
AWSQuickSetupCFGCPacksPermissionsBoundary: nova política	O Systems Manager adicionou uma nova política para permitir que a Quick Setup implemente pacotes de conformidade do AWS Config.	3 de julho de 2024
AWSSystemsManagerOpsDataSyncServiceRolePolicy: atualização para uma política existente	O OpsCenter atualizou a política para melhorar a segurança do código de serviço dentro do perfil vinculado a serviço para o Explorer gerenciar operações relacionadas a OpsData.	3 de julho de 2023

Alteração	Descrição	Data
AmazonSSMManagedEC2InstanceDefaultPolicy – Nova política	O Systems Manager adicionou uma nova política para permitir a funcionalidade do Systems Manager em instâncias do Amazon EC2 sem o uso de um perfil de instância do IAM.	18 de agosto de 2022
AmazonSSMServiceRolePolicy: atualização para uma política existente	O Systems Manager adicionou novas permissões para permitir que o Explorer crie uma regra gerenciada quando você ativar o Security Hub no Explorer ou no OpsCenter. Novas permissões foram adicionadas para verificar se a configuração e o Compute Optimizer atendem aos requisitos necessários antes de permitir o OpsData.	27 de abril de 2021
AWSSystemsManagerOpsDataSyncServiceRolePolicy – Nova política	O Systems Manager adicionou uma nova política para criar e atualizar o OpsItems e as descobertas do OpsData do Security Hub no Explorer e no OpsCenter.	27 de abril de 2021
AmazonSSMServiceRolePolicy: atualização para uma política existente	O Systems Manager adicionou novas permissões para permitir a visualização agregada de detalhes do OpsData e OpsItems em várias contas e em Regiões da AWS e no Explorer.	24 de março de 2021

Alteração	Descrição	Data
O Systems Manager iniciou o rastreamento das alterações	O Systems Manager começou a monitorar as alterações para as políticas gerenciadas da AWS.	12 de março de 2021

Políticas gerenciadas adicionais para o Systems Manager

Além das políticas gerenciadas descritas anteriormente neste tópico, o Systems Manager também oferece suporte às políticas gerenciadas a seguir.

- `AmazonSSMAutomationApproverAccess`: política gerenciada pela AWS que permite visualizar execuções de automação e enviar decisões de aprovação para automação que está aguardando aprovação.
- `AmazonSSMDirectoryServiceAccess`: política gerenciada pela AWS que permite ao SSM Agent acessar o AWS Directory Service em nome do usuário para solicitações de ingresso no domínio pelo nó gerenciado.
- `AmazonSSMFullAccess`: política gerenciada pela AWS que concede acesso total à API e a documentos do Systems Manager.
- `AmazonSSMMaintenanceWindowRole`: política gerenciada pela AWS que fornece janelas de manutenção com permissões para a API do Systems Manager.
- `AmazonSSMManagedInstanceCore` – Política gerenciada do AWS que permite que uma instância use a funcionalidade básica do serviço do Systems Manager.
- `AmazonSSMPatchAssociation`: política gerenciada pela AWS que fornece acesso a instâncias filhas para operações de associação de patches.
- `AmazonSSMReadOnlyAccess`: política gerenciada pela AWS que concede acesso a operações de API somente leitura do Systems Manager, como `Get*` e `List*`.
- `AWSSSMOpsInsightsServiceRolePolicy`: política gerenciada pela AWS que fornece permissões para criar e atualizar OPSitems de insights operacionais no Systems Manager. Usada para fornecer permissões por meio do perfil vinculado ao serviço `AWSServiceRoleForAmazonSSM_OpsInsights`.
- `AWSSystemsManagerAccountDiscoveryServicePolicy`: política gerenciada pela AWS que concede ao Systems Manager permissões para descobrir informações da Conta da AWS.

- AWSSystemsManagerChangeManagementServicePolicy: política gerenciada pela AWS fornece acesso aos recursos da AWS gerenciados ou usados pelo framework de gerenciamento de alterações do Systems Manager e usados pelo perfil vinculado ao serviço AWSServiceRoleForSystemsManagerChangeManagement.
- AmazonEC2RoleforSSM: política obsoleta, não deve mais ser usada. Em seu lugar, use a política AmazonSSMManagedInstanceCore para permitir a funcionalidade principal do serviço Systems Manager em instâncias do EC2. Para obter informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager.

Solução de problemas de identidade e acesso do AWS Systems Manager

Use as seguintes informações para ajudar a diagnosticar e corrigir problemas comuns que podem ser encontrados ao trabalhar com o AWS Systems Manager e o AWS Identity and Access Management (IAM).

Tópicos

- Não tenho autorização para executar uma ação no Systems Manager
- Não estou autorizado a executar iam:PassRole
- Quero permitir que as pessoas fora da minha Conta da AWS acessem meus recursos do Systems Manager

Não tenho autorização para executar uma ação no Systems Manager

Se o AWS Management Console informar que você não está autorizado a executar uma ação, você deverá entrar em contato com o administrador para obter assistência. Seu administrador é a pessoa que forneceu a você suas credenciais de início de sessão.

O erro exemplificado a seguir ocorre quando o usuário mateojackson tenta usar o console para visualizar detalhes sobre um documento, mas não tem permissões ssm:GetDocument.

```
User: arn:aws:ssm::123456789012:user/mateojackson isn't authorized to perform:
 ssm:GetDocument on resource: MyExampleDocument
```

Neste caso, Mateo pede ao administrador para atualizar suas políticas para permitir a ele o acesso ao recurso MyExampleDocument usando a ação ssm:GetDocument.

Não estou autorizado a executar iam:PassRole

Se você receber uma mensagem de erro informando que não está autorizado a executar a ação `iam:PassRole`, as suas políticas devem ser atualizadas para permitir que você passe uma função para o Systems Manager.

Alguns Serviços da AWS permitem que você passe uma função existente para o serviço, em vez de criar uma nova função de serviço ou função vinculada ao serviço. Para fazê-lo, você deve ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando uma usuária do IAM chamada `marymajor` tenta utilizar o console para executar uma ação no Systems Manager. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se você precisar de ajuda, entre em contato com seu administrador AWS. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que as pessoas fora da minha Conta da AWS acessem meus recursos do Systems Manager

Você pode criar uma função que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. Você pode especificar quem é confiável para assumir o perfil. Para serviços que oferecem compatibilidade com políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Saiba mais consultando o seguinte:

- Para saber se o Systems Manager oferece compatibilidade com esses atributos, consulte Como o AWS Systems Manager funciona com o IAM.
- Saiba como conceder acesso a seus recursos em todos os Contas da AWS pertencentes a você, consulte Fornecendo Acesso a um Usuário do IAM em Outro Conta da AWS Pertencente a Você no Guia de Usuário do IAM.

- Para saber como conceder acesso a seus recursos para terceiros Contas da AWS, consulte Fornecimento de acesso a Contas da AWS pertencentes a terceiros no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para saber a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Usar perfis vinculados a serviço do Systems Manager

O AWS Systems Manager utiliza perfis vinculados a serviço do AWS Identity and Access Management (IAM). O perfil vinculado a serviço é um tipo exclusivo de perfil do IAM vinculado diretamente ao Systems Manager. As funções vinculadas a serviços são predefinidas pelo Systems Manager e incluem todas as permissões que o serviço requer para chamar outros Serviços da AWS em seu nome.

> **Note**
>
> Uma função de perfil de serviço é diferente de uma função vinculada a serviço. Umm perfil de serviço é um tipo de perfil do AWS Identity and Access Management (IAM) que concede permissões para um AWS service (Serviço da AWS), para que o serviço possa acessar recursos da AWS. Apenas alguns cenários do Systems Manager exigem uma função de serviço. Ao criar uma função de serviço para o Systems Manager, você pode escolher as permissões a serem concedidas, a fim de que elas possam acessar ou interagir com outros recursos da AWS.

Um perfil vinculado a serviço facilita a configuração do Systems Manager porque você não precisa adicionar as permissões necessárias manualmente. O Systems Manager define as permissões de seus perfis vinculados a serviço e, a menos que definido em contrário, somente o Systems Manager pode assumir seus perfis. As permissões definidas incluem a política de confiança e a política de permissões, e essa política de permissões não pode ser anexada a nenhuma outra entidade do IAM.

Um perfil vinculado a serviço poderá ser excluído somente após a exclusão dos recursos relacionados. Isso protege seus recursos do Systems Manager, pois você não pode remover por engano as permissões de acesso aos recursos.

> **Note**
>
> Para nós que não são do EC2 em um ambiente híbrido e multinuvem, você precisa de uma perfil do IAM adicional que permita que as máquinas se comuniquem com o serviço do Systems Manager. Trata-se da função de serviço do IAM para o Systems Manager. Essa função concede a confiança AssumeRole do AWS Security Token Service (AWS STS) ao serviço Systems Manager. A ação `AssumeRole` retorna um conjunto de credenciais de segurança temporárias (que consistem em um ID de chave de segurança, uma chave de acesso secreta e um token de segurança). Você pode usar essas credenciais temporárias para acessar recursos da AWS aos quais talvez não tenha acesso normalmente. Para obter mais informações, consulte Criar o perfil de serviço do IAM obrigatório para o Systems Manager em ambientes híbridos e multinuvem e AssumeRole na Referência de API do AWS Security Token Service.

Para obter informações sobre outros serviços compatíveis com funções vinculadas a serviços, consulte Serviços da AWS que funcionam com o IAM e procure os serviços que apresentam Sim na coluna Funções vinculadas a serviços. Escolha um Sim com um link para visualizar a documentação do perfil vinculado para esse serviço.

Tópicos

- Usar perfis para coletar inventário e visualizar OpsData
- Usar perfis para coletar informações da Conta da AWS para o OpsCenter e o Explorer
- Usar perfis para criar OpsData e OpsItems para o Explorer
- Usar perfis para criar insights operacionais de OpsItems no OpsCenter do Systems Manager
- Usar funções para manter a integridade e a uniformidade dos recursos provisionados da Quick Setup
- Usar perfis para exportar Explorer OpsData
- Uso de perfis para habilitar o acesso a nós just-in-time
- Uso de perfis para enviar notificações de solicitação de acesso a nós just-in-time

Usar perfis para coletar inventário e visualizar OpsData

O Systems Manager usa o perfil vinculado ao serviço chamado **`AWSServiceRoleForAmazonSSM`**. O AWS Systems Manager usa esse perfil de serviço do IAM para gerenciar os recursos AWS em seu nome.

Permissões do perfil vinculado ao serviço para inventário, OpsData e OpsItems

A função vinculada a serviço `AWSServiceRoleForAmazonSSM` confia somente em `ssm.amazonaws.com` para assumir essa função.

O perfil vinculado ao serviço `AWSServiceRoleForAmazonSSM` do Systems Manager pode ser usado para o seguinte:

- A ferramenta Inventory do Systems Manager usa o perfil vinculado ao serviço `AWSServiceRoleForAmazonSSM` para coletar metadados de inventário de tags e grupos de recursos.
- A ferramenta Explorer usa o perfil vinculado ao serviço `AWSServiceRoleForAmazonSSM` para habilitar a visualização de OpsData e OpsItems de várias contas. Essa função vinculada ao serviço do também permite que oExplorerpara criar uma regra gerenciada quando você habilita o Security Hub como uma fonte de dados doExplorerouOpsCenter.

> **Important**
>
> Anteriormente, o console do Systems Manager permitia a você escolher o perfil `AWSServiceRoleForAmazonSSM` vinculado ao serviço do IAM gerenciado pela AWS para usar como perfil de manutenção para suas tarefas. O uso desse perfil e sua política associada, `AmazonSSMServiceRolePolicy`, para tarefas de janela de manutenção não é mais recomendado. Se estiver usando esse perfil para tarefas de janela de manutenção agora, recomendamos parar de usá-lo. Em vez disso, crie seu próprio perfil do IAM para permitir a comunicação entre o Systems Manager e outros Serviços da AWS quando as tarefas da janela de manutenção são executadas.
> Para obter mais informações, consulte Configurar o Maintenance Windows.

A política gerida que é utilizada para fornecer permissões para oAWSServiceRoleForAmazonSSMfunção éAmazonSSMServiceRolePolicy. Para

obter detalhes sobre as permissões necessárias, consulte Política gerenciada da AWS: AmazonSSMServiceRolePolicy.

Criar uma função vinculada ao serviço **AWSServiceRoleForAmazonSSM** para o Systems Manager

Você pode usar o console do IAM para criar uma função vinculada ao serviço com o caso de uso do EC2. Usando comandos para o IAM noAWS Command Line Interface(AWS CLI) ou usando a API do IAM, crie uma função vinculada ao serviço com `ossm.amazonaws.com`O nome do serviço. Para obter mais informações, consulte Criar um perfil vinculado a serviço no Guia do usuário do IAM.

Se excluir esse perfil vinculado ao serviço e precisar criá-lo novamente, será possível usar esse mesmo processo para recriar o perfil em sua conta.

Editar uma função vinculada ao serviço **AWSServiceRoleForAmazonSSM** para o Systems Manager

O Systems Manager não permite que você edite a função vinculada a serviço `AWSServiceRoleForAmazonSSM`. Depois de criar uma função vinculada a serviço, você não poderá alterar o nome da função, já que várias entidades poderão fazer referência à função. No entanto, você poderá editar a descrição do perfil usando o IAM. Para obter mais informações, consulte Editar um perfil vinculado ao serviço no Guia do usuário do IAM.

Excluir uma função vinculada ao serviço **AWSServiceRoleForAmazonSSM** para o Systems Manager

Se você não precisar mais usar um recurso ou serviço que exija uma função vinculada ao serviço, é recomendável exclui-la. Dessa forma, você não terá uma entidade não utilizada que não seja monitorada ativamente ou mantida. Também é possível usar o console do IAM, a AWS CLI ou a API do IAM para excluir manualmente a função vinculada ao serviço. Para isso, primeiro você deve limpar manualmente os recursos de sua função vinculada ao serviço e, em seguida, excluí-la manualmente.

Como o perfil vinculado ao serviço `AWSServiceRoleForAmazonSSM` pode ser usado por várias ferramentas, verifique se nenhum deles está usando o perfil antes de tentar excluí-lo.

- Inventory: se você excluir o perfil vinculado ao serviço usado pela ferramenta Inventory, os dados do Inventory referentes a tags e grupos de recursos não serão mais sincronizados. Você deve limpar os recursos de sua função vinculada ao serviço antes de exclui-la manualmente.

- Explorer: se você excluir o perfil vinculado ao serviço usado pela ferramenta Explorer, o OpsData entre contas e entre regiões e os OpsItems não serão mais visíveis.

> **Note**
>
> Se o serviço Systems Manager estiver usando o perfil quando você tentar excluir etiquetas ou grupos de recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Para excluir recursos do Systems Manager usados por **`AWSServiceRoleForAmazonSSM`**

1. Para excluir etiquetas, consulte Adicionar e excluir etiquetas em um recurso individual.
2. Para excluir grupos de recursos, consulte Excluir grupos do AWS Resource Groups.

Para excluir manualmente a função vinculada ao serviço **`AWSServiceRoleForAmazonSSM`** usando o IAM

Use o console do IAM, a AWS CLI ou a API do IAM para excluir a função vinculada ao serviço `AWSServiceRoleForAmazonSSM`. Para obter mais informações, consulte Excluir um perfil vinculado ao serviço no Guia do usuário do IAM.

Regiões compatíveis com o perfil vinculado ao serviço **`AWSServiceRoleForAmazonSSM`** do Systems Manager

O Systems Manager é compatível com a função vinculadas ao serviço `AWSServiceRoleForAmazonSSM` em todas as Regiões da AWS em que o serviço está disponível. Para obter mais informações, consulte Endpoints e cotas do AWS Systems Manager.

Usar perfis para coletar informações da Conta da AWS para o OpsCenter e o Explorer

O Systems Manager usa a função vinculada ao serviço chamada **`AWSServiceRoleForAmazonSSM_AccountDiscovery`**. O AWS Systems Manager usa esse perfil de serviço do IAM para chamar outros Serviços da AWS para descobrir informações sobre a Conta da AWS.

Permissões de função vinculada ao serviço para detecção de conta do Systems Manager

O perfil vinculado ao serviço `AWSServiceRoleForAmazonSSM_AccountDiscovery` confia nos seguintes serviços para aceitar o perfil:

- `accountdiscovery.ssm.amazonaws.com`

A política de permissões da função permite que o Systems Manager conclua as seguintes ações nos recursos especificados:

- `organizations:DescribeAccount`
- `organizations:DescribeOrganizationalUnit`
- `organizations:DescribeOrganization`
- `organizations:ListAccounts`
- `organizations:ListAWSServiceAccessForOrganization`
- `organizations:ListChildren`
- `organizations:ListParents`
- `organizations:ListDelegatedServicesForAccount`
- `organizations:ListDelegatedAdministrators`
- `organizations:ListRoots`

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua um perfil vinculado a serviço. Para obter mais informações, consulte Permissões de perfil vinculado ao serviço no Guia do usuário do IAM.

Criar uma função vinculada ao serviço **`AWSServiceRoleForAmazonSSM_AccountDiscovery`** para o Systems Manager

Você deverá criar um perfil vinculado ao serviço se quiser usar o Explorer e o OpsCenter, ambos ferramentas do Systems Manager, entre várias Contas da AWS. Para o OpsCenter, você deve criar a função vinculada ao serviço. Para ter mais informações, consulte (Opcional) Configuração do OpsCenter para gerenciar OpsItems de forma centralizada entre contas.

Para o Explorer, se você criar uma sincronização de dados de recurso usando o Systems Manager no AWS Management Console, é possível criar a função vinculada ao serviço escolhendo o

botão Create role (Criar função). Se você deseja criar uma sincronização de dados de recursos programaticamente, você deve criar a função antes de criar a sincronização de dados de recurso. Você pode criar a função usando a operação da API CreateServiceLinkedRole.

Editar uma função vinculada ao serviço **AWSServiceRoleForAmazonSSM_AccountDiscovery** para o Systems Manager

O Systems Manager não permite que você edite a função vinculada a serviço `AWSServiceRoleForAmazonSSM_AccountDiscovery`. Depois de criar uma função vinculada a serviço, você não poderá alterar o nome da função, já que várias entidades poderão fazer referência à função. No entanto, você poderá editar a descrição do perfil usando o IAM. Para obter mais informações, consulte Editar um perfil vinculado ao serviço no Guia do usuário do IAM.

Excluir uma função vinculada ao serviço **AWSServiceRoleForAmazonSSM_AccountDiscovery** para o Systems Manager

Se você não precisar mais usar um atributo ou serviço que requer um perfil vinculado ao serviço, é recomendável excluí-la. Dessa forma, você não terá uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar suo perfil vinculado ao serviço para excluí-la manualmente.

Limpar a função vinculada ao serviço **AWSServiceRoleForAmazonSSM_AccountDiscovery**

Antes de usar o IAM para excluir a função vinculada ao serviço `AWSServiceRoleForAmazonSSM_AccountDiscovery`, você primeiro deve excluir todas as sincronizações de dados de recursos do Explorer.

> **Note**
>
> Se o serviço Systems Manager estiver usando a função quando você tenta excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Excluir manualmente a função vinculada ao serviço **AWSServiceRoleForAmazonSSM_AccountDiscovery**

Use o console do IAM, a AWS CLI ou a API da AWS para excluir o perfil vinculado ao serviço `AWSServiceRoleForAmazonSSM_AccountDiscovery`. Para obter mais informações, consulte Excluir uma função vinculada ao serviço no Guia do usuário do IAM.

Regiões compatíveis com o perfil vinculado ao serviço **`AWSServiceRoleForAmazonSSM_AccountDiscovery`** do Systems Manager

O Systems Manager oferece suporte a funções vinculadas a serviços em todas as regiões em que o serviço está disponível. Para obter mais informações, consulte Endpoints e cotas do AWS Systems Manager.

Atualiza para a perfil vinculado ao serviço AWSServiceRoleForAmazonSSM_AccountDiscovery

Visualize detalhes sobre as atualizações do perfil vinculado ao serviço AWSServiceRoleForAmazonSSM_AccountDiscovery service-linked desde que esse serviço começou a monitorar essas alterações. Para obter alertas automáticos sobre alterações feitas nesta página, inscreva-se no feed RSS na página Histórico do documento do Systems Manager.

Alteração	Descrição	Data
Novas permissões adicionadas	Esta função vinculada ao serviço agora inclui as permissões `organizations:DescribeOrganizationalUnit` e `organizations:ListRoots`. Essas permissões permitem que uma conta de gerenciamento do AWS Organizations ou uma conta de administrador delegado do Systems Manager trabalhem com OpsItems entre contas. Para ter mais informações, consulte (Opcional) Configuração do OpsCenter para gerenciar OpsItems de forma centralizada entre contas.	17 de outubro de 2022

Usar perfis para criar OpsData e OpsItems para o Explorer

O Systems Manager usa o perfil vinculado ao serviço chamado **`AWSServiceRoleForSystemsManagerOpsDataSync`**. O AWS Systems Manager usa esse perfil de serviço do IAM para o Explorer, visando criar o OpsData e OpsItems.

Permissões de função vinculada ao serviço para sincronização de OpsData do Systems Manager

A função vinculada ao serviço `AWSServiceRoleForSystemsManagerOpsDataSync` confia nos seguintes serviços para aceitar a função:

- `opsdatasync.ssm.amazonaws.com`

A política de permissões da função permite que o Systems Manager conclua as seguintes ações nos recursos especificados:

- O Systems Manager Explorer exige que uma função vinculada ao serviço conceda permissão para atualizar uma descoberta de segurança quando um OpsItem é atualizado, criar e atualizar um OpsItem, e desativar a origem dos dados do Security Hub quando uma regra gerenciada do SSM é excluída pelos clientes.

A política gerida que é utilizada para fornecer permissões para o`AWSServiceRoleForSystemsManagerOpsDataSync`função é`AWSSystemsManagerOpsDataSyncServiceRolePolicy`. Para obter detalhes sobre as permissões necessárias, consulte Política gerenciada pela AWS: AWSSystemsManagerOpsDataSyncServiceRolePolicy.

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua uma função vinculada a serviço. Para obter mais informações, consulte Permissões de função vinculada a serviços no Guia do Usuário do IAM.

Criar uma função vinculada ao serviço **`AWSServiceRoleForSystemsManagerOpsDataSync`** para o Systems Manager

Não é necessário criar manualmente uma função vinculada ao serviço. Quando você habilita o Explorer no AWS Management Console, o Systems Manager cria a função vinculada ao serviço para você.

> **Important**
>
> Essa função vinculada ao serviço pode ser exibida em sua conta se você concluiu uma ação em outro serviço que usa os recursos compatíveis com essa função. Além disso, se você estava usando o serviço Systems Manager antes de 1º de janeiro de 2017, quando começou a oferecer suporte às funções vinculadas a serviços, o Systems Manager criou a função `AWSServiceRoleForSystemsManagerOpsDataSync` em sua conta. Para saber mais, consulte Uma nova função apareceu na minha conta do IAM.

Se excluir essa função vinculada ao serviço e precisar criá-la novamente, você pode usar esse mesmo processo para recriar a função na sua conta. Quando você habilita o Explorer no AWS Management Console, o Systems Manager cria a função vinculada ao serviço novamente.

Você também pode usar o console do IAM para criar um perfil vinculado ao serviço com o caso de uso do perfil de serviço da AWS que permite que o Explorer crie OpsData e OpsItems. Na AWS CLI ou na API do AWS, crie uma função vinculada ao serviço com o nome de serviço `opsdatasync.ssm.amazonaws.com`. Para obter mais informações, consulte Criar uma função vinculada ao serviço no Manual do usuário do IAM. Se você excluir essa função vinculada ao serviço, será possível usar esse mesmo processo para criar a função novamente.

Editar uma função vinculada ao serviço **`AWSServiceRoleForSystemsManagerOpsDataSync`** para o Systems Manager

O Systems Manager não permite que você edite a função vinculada a serviço `AWSServiceRoleForSystemsManagerOpsDataSync`. Depois de criar uma função vinculada a serviço, você não poderá alterar o nome da função, já que várias entidades poderão fazer referência à função. No entanto, você poderá editar a descrição do perfil usando o IAM. Para ter mais informações, consulte Editar uma função vinculada a serviço no Guia do usuário do IAM.

Excluir uma função vinculada ao serviço **`AWSServiceRoleForSystemsManagerOpsDataSync`** para o Systems Manager

Se você não precisar mais usar um atributo ou serviço que requer um perfil vinculado ao serviço, é recomendável excluí-la. Dessa forma, você não terá uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar os atributos de seu perfil vinculado ao serviço antes de excluí-lo manualmente.

> **Note**
>
> Se o serviço Systems Manager estiver usando a função quando você tenta excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

O procedimento para excluir recursos do Systems Manager usados pelo perfil `AWSServiceRoleForSystemsManagerOpsDataSync` depende se você configurou o Explorerou o OpsCenter para se integrar com o Security Hub.

Para excluir recursos do Systems Manager usados pela função **`AWSServiceRoleForSystemsManagerOpsDataSync`**

- Para impedir que o Explorer crie novos OpsItems para descobertas do Security Hub, consulte Como parar de receber descobertas.
- Para impedir que o OpsCenter crie novos OpsItems para descobertas do Security Hub, consulte

Para excluir manualmente a função vinculada ao serviço **`AWSServiceRoleForSystemsManagerOpsDataSync`** usando o IAM

Use o console do IAM, a AWS CLI ou a API da AWS para excluir a função vinculada ao serviço `AWSServiceRoleForSystemsManagerOpsDataSync`. Para obter mais informações, consulte Excluir uma função vinculada ao serviço no Guia do usuário do IAM.

Regiões compatíveis com o perfil vinculado ao serviço **`AWSServiceRoleForSystemsManagerOpsDataSync`** do Systems Manager

O Systems Manager oferece suporte a funções vinculadas a serviços em todas as regiões nas quais o serviço estiver disponível. Para obter mais informações, consulte Endpoints e cotas do AWS Systems Manager.

O Systems Manager não oferece suporte ao uso de funções vinculadas a serviços em todas as regiões em que o serviço está disponível. Você pode usar a função `AWSServiceRoleForSystemsManagerOpsDataSync` nas seguintes regiões.

Nome do Região da AWS	Identidade da região	Suporte no Systems Manager
Leste dos EUA (Norte da Virgínia)	us-east-1	Sim
Leste dos EUA (Ohio)	us-east-2	Sim
Oeste dos EUA (N. da Califórnia)	us-west-1	Sim
Oeste dos EUA (Oregon)	us-west-2	Sim
Ásia-Pacífico (Mumbai)	ap-south-1	Sim
Ásia-Pacífico (Osaka)	ap-northeast-3	Sim
Ásia-Pacífico (Seul)	ap-northeast-2	Sim
Ásia-Pacífico (Singapura)	ap-southeast-1	Sim
Ásia-Pacífico (Sydney)	ap-southeast-2	Sim
Ásia-Pacífico (Tóquio)	ap-northeast-1	Sim
Canadá (Central)	ca-central-1	Sim
Europa (Frankfurt)	eu-central-1	Sim
Europa (Irlanda)	eu-west-1	Sim
Europa (Londres)	eu-west-2	Sim
Europa (Paris)	eu-west-3	Sim
Europa (Estocolmo)	eu-north-1	Sim
América do Sul (São Paulo)	sa-east-1	Sim
AWS GovCloud (US)	us-gov-west-1	Não

Usar perfis para criar insights operacionais de OpsItems no OpsCenter do Systems Manager

O Systems Manager usa a função vinculada ao serviço chamada **`AWSServiceRoleForAmazonSSM_OpsInsights`**. O AWS Systems Manager usa essa função de serviço do IAM para criar e atualizar insights operacionais de OpsItems no Systems Manager OpsCenter.

Permissões de perfil vinculado ao serviço **`AWSServiceRoleForAmazonSSM_OpsInsights`** para OpsItems de insight operacional do Systems Manager

O perfil vinculado ao serviço `AWSServiceRoleForAmazonSSM_OpsInsights` confia nos seguintes serviços para aceitar o perfil:

- `opsinsights.ssm.amazonaws.com`

A política de permissões da função permite que o Systems Manager conclua as seguintes ações nos recursos especificados:

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AllowCreateOpsItem",
   "Effect": "Allow",
   "Action": [
    "ssm:CreateOpsItem",
    "ssm:AddTagsToResource"
   ],
   "Resource": "*"
  },
  {
   "Sid": "AllowAccessOpsItem",
   "Effect": "Allow",
   "Action": [
    "ssm:UpdateOpsItem",
```

```
     "ssm:GetOpsItem"
    ],
    "Resource": "*",
    "Condition": {
     "StringEquals": {
      "aws:ResourceTag/SsmOperationalInsight": "true"
     }
    }
   }
  ]
}
```

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua um perfil vinculado a serviço. Para obter mais informações, consulte Permissões de perfil vinculado ao serviço no Guia do usuário do IAM.

Criar uma função vinculada ao serviço **AWSServiceRoleForAmazonSSM_OpsInsights** para o Systems Manager

Crie uma função vinculada ao serviço. Se você habilitar insights operacionais usandoSystems ManagernoAWS Management Console, é possível criar a função vinculada ao serviço escolhendo a opçãoHabilitar o.

Editar uma função vinculada ao serviço **AWSServiceRoleForAmazonSSM_OpsInsights** para o Systems Manager

O Systems Manager não permite que você edite a função vinculada ao serviço `AWSServiceRoleForAmazonSSM_OpsInsights`. Depois que criar um perfil vinculado ao serviço, você não poderá alterar o nome do perfil, pois várias entidades podem fazer referência a ele. No entanto, será possível editar a descrição do perfil usando o IAM. Para obter mais informações, consulte Editar um perfil vinculado ao serviço no Guia do usuário do IAM.

Excluir uma função vinculada ao serviço **AWSServiceRoleForAmazonSSM_OpsInsights** para o Systems Manager

Se você não precisar mais usar um recurso ou serviço que requer um perfil vinculado ao serviço, é recomendável excluí-lo. Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar suo perfil vinculado ao serviço para excluí-la manualmente.

Limpar a função vinculada ao serviço **AWSServiceRoleForAmazonSSM_OpsInsights**

Antes de usar o IAM para excluir um perfil vinculado ao serviço `AWSServiceRoleForAmazonSSM_OpsInsights`, primeiro é necessário desativar os insights operacionais no OpsCenter do Systems Manager. Para obter mais informações, consulte Analisar insights operacionais para reduzir OpsItems.

Excluir manualmente a função vinculada ao serviço **AWSServiceRoleForAmazonSSM_OpsInsights**

Use o console do IAM, a AWS CLI ou a API da AWS para excluir o perfil vinculado ao serviço `AWSServiceRoleForAmazonSSM_OpsInsights`. Para obter mais informações, consulte Excluir um perfil vinculado ao serviço no Guia do usuário do IAM.

Regiões compatíveis com o perfil vinculado ao serviço **AWSServiceRoleForAmazonSSM_OpsInsights** do Systems Manager

O Systems Manager não oferece suporte ao uso de funções vinculadas a serviços em todas as regiões em que o serviço está disponível. Você pode usar a função WSServiceRoleForAmazonSSM_OpsInsights nas regiões a seguir.

Nome da região	Identidade da região	Suporte no Systems Manager
Leste dos EUA (Norte da Virgínia)	us-east-1	Sim
Leste dos EUA (Ohio)	us-east-2	Sim
Oeste dos EUA (N. da Califórnia)	us-west-1	Sim
Oeste dos EUA (Oregon)	us-west-2	Sim
Ásia-Pacífico (Mumbai)	ap-south-1	Sim
Ásia-Pacífico (Tóquio)	ap-northeast-1	Sim
Ásia-Pacífico (Seul)	ap-northeast-2	Sim
Ásia-Pacífico (Singapura)	ap-southeast-1	Sim
Ásia-Pacífico (Sydney)	ap-southeast-2	Sim

Nome da região	Identidade da região	Suporte no Systems Manager
Ásia-Pacífico (Hong Kong)	ap-east-1	Sim
Canadá (Central)	ca-central-1	Sim
Europa (Frankfurt)	eu-central-1	Sim
Europa (Irlanda)	eu-west-1	Sim
Europa (Londres)	eu-west-2	Sim
Europa (Paris)	eu-west-3	Sim
Europa (Estocolmo)	eu-north-1	Sim
Europa (Milão)	eu-south-1	Sim
América do Sul (São Paulo)	sa-east-1	Sim
Oriente Médio (Barém)	me-south-1	Sim
África (Cidade do Cabo)	af-south-1	Sim
AWS GovCloud (US)	us-gov-west-1	Sim
AWS GovCloud (US)	us-gov-east-1	Sim

Usar funções para manter a integridade e a uniformidade dos recursos provisionados da Quick Setup

O Systems Manager usa a função vinculada ao serviço chamada **`AWSServiceRoleForSSMQuickSetup`**.

Permissões de função vinculada a serviços do **`AWSServiceRoleForSSMQuickSetup`** para o Systems Manager

O perfil vinculado ao serviço `AWSServiceRoleForSSMQuickSetup` confia nos seguintes serviços para aceitar o perfil:

- `ssm-quicksetup.amazonaws.com`

O AWS Systems Manager usa esse perfil de serviço do IAM para verificar a integridade da configuração, garantir o uso uniforme dos parâmetros e dos recursos provisionados e [corrigir os recursos quando o desvio é detectado.

A política de permissões da função permite que o Systems Manager conclua as seguintes ações nos recursos especificados:

- `ssm` (Systems Manager): lê informações sobre o estado em que os recursos configurados devem estar, inclusive nas contas de administrador delegado.
- `iam` (AWS Identity and Access Management): isso é necessário para que as sincronizações de dados de recursos sejam acessíveis em organizações inteiras no AWS Organizations.
- `organizations` (AWS Organizations) lê informações sobre as contas-membros que pertencem a uma organização, conforme configurado em Organizations.
- `cloudformation` (AWS CloudFormation): lê informações sobre as pilhas do CloudFormation usadas para gerenciar o estado dos recursos e as operações do conjunto de pilhas do CloudFormation.

A política gerida que é utilizada para fornecer permissões para o`AWSServiceRoleForSSMQuickSetup`função é`SSMQuickSetupRolePolicy`. Para obter detalhes sobre as permissões necessárias, consulte Política gerenciada da AWS: SSMQuickSetupRolePolicy.

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua um perfil vinculado a serviço. Para obter mais informações, consulte Permissões de perfil vinculado ao serviço no Guia do usuário do IAM.

Criar uma função vinculada ao serviço **`AWSServiceRoleForSSMQuickSetup`** para o Systems Manager

Você não precisa criar manualmente o perfil vinculado a serviços AWSServiceRoleForSSMQuickSetup. Quando você cria uma configuração do Quick Setup no AWS Management Console, o Systems Manager cria a função vinculada a serviços para você.

Editar uma função vinculada ao serviço **AWSServiceRoleForSSMQuickSetup** para o Systems Manager

O Systems Manager não permite que você edite a função vinculada ao serviço `AWSServiceRoleForSSMQuickSetup`. Depois que criar um perfil vinculado ao serviço, você não poderá alterar o nome do perfil, pois várias entidades podem fazer referência a ele. No entanto, será possível editar a descrição do perfil usando o IAM. Para obter mais informações, consulte Editar um perfil vinculado ao serviço no Guia do usuário do IAM.

Excluir uma função vinculada ao serviço **AWSServiceRoleForSSMQuickSetup** para o Systems Manager

Se você não precisar mais usar um atributo ou serviço que requer uma função vinculada a serviço, é recomendável excluí-la. Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar suo perfil vinculado ao serviço para excluí-la manualmente.

Limpar a função vinculada ao serviço **AWSServiceRoleForSSMQuickSetup**

Antes de poder usar o IAM para excluir uma função vinculada a serviços do `AWSServiceRoleForSSMQuickSetup`, você deve primeiro excluir as configurações do Quick Setup que estão usando a função. Para ter mais informações, consulte Editar e excluir a configuração.

Excluir manualmente a função vinculada ao serviço **AWSServiceRoleForSSMQuickSetup**

Use o console do IAM, a AWS CLI ou a API da AWS para excluir o perfil vinculado ao serviço `AWSServiceRoleForSSMQuickSetup`. Para obter mais informações, consulte os tópicos a seguir.

- Exclusão de uma função vinculada a serviços no Guia do usuário do IAM
- delete-configuration-manager na seção da Quick Setup da Referência da AWS CLI
- DeleteConfigurationManager na Referência de API do Quick Setup

Regiões compatíveis com o perfil vinculado ao serviço **AWSServiceRoleForSSMQuickSetup** do Systems Manager

O Systems Manager não oferece suporte ao uso de funções vinculadas a serviços em todas as regiões em que o serviço está disponível. Você pode usar o perfil AWSServiceRoleForSSMQuickSetup nas regiões a seguir.

- Leste dos EUA (Ohio)
- Leste dos EUA (Norte da Virgínia)
- Oeste dos EUA (Norte da Califórnia)
- Oeste dos EUA (Oregon)
- Ásia-Pacífico (Mumbai)
- Ásia-Pacífico (Seul)
- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Tóquio)
- Canadá (Central)
- Europa (Frankfurt)
- Europa (Estocolmo)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Paris)
- América do Sul (São Paulo)

Usar perfis para exportar Explorer OpsData

O AWS Systems Manager Explorer usa o perfil de serviço AmazonSSMExplorerExportRole para exportar dados de operações (OpsData) usando o runbook de automação `AWS-ExportOpsDataToS3`.

Permissões de função vinculada ao serviço Explorer

A função vinculada a serviço `AmazonSSMExplorerExportRole` confia somente em `ssm.amazonaws.com` para assumir essa função.

Você pode usar a o perfil vinculado ao serviço `AmazonSSMExplorerExportRole` para exportar dados de operações (OpsData) usando o runbook de automação `AWS-ExportOpsDataToS3`. É possível exportar cinco mil relatórios de OpsData do Explorer, como um arquivo de valores separados por vírgula (.csv), para um bucket do Amazon Simple Storage Service (Amazon S3).

A política de permissões da função permite que o Systems Manager conclua as seguintes ações nos recursos especificados:

- `s3:PutObject`
- `s3:GetBucketAcl`
- `s3:GetBucketLocation`
- `sns:Publish`
- `logs:DescribeLogGroups`
- `logs:DescribeLogStreams`
- `logs:CreateLogGroup`
- `logs:PutLogEvents`
- `logs:CreateLogStream`
- `ssm:GetOpsSummary`

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua uma função vinculada a serviço. Para obter mais informações, consulte Permissões de função vinculada a serviços no Guia do Usuário do IAM.

Criar uma função vinculada ao serviço **`AmazonSSMExplorerExportRole`** para o Systems Manager

O Systems Manager cria o perfil vinculado ao serviço `AmazonSSMExplorerExportRole` quando você exporta o OpsData usando Explorer no console do Systems Manager. Para ter mais informações, consulte Exportar OpsData do Systems Manager Explorer.

Se excluir essa função vinculada ao serviço e precisar criá-la novamente, você poderá usar esse mesmo processo para recriar a função em sua conta.

Editar uma função vinculada ao serviço **`AmazonSSMExplorerExportRole`** para o Systems Manager

O Systems Manager não permite que você edite a função vinculada a serviço `AmazonSSMExplorerExportRole`. Depois de criar uma função vinculada a serviço, você não poderá alterar o nome da função, já que várias entidades poderão fazer referência à função. No entanto, você poderá editar a descrição do perfil usando o IAM. Para ter mais informações, consulte Editar uma função vinculada a serviço no Guia do usuário do IAM.

Excluir uma função vinculada ao serviço **AmazonSSMExplorerExportRole** para o Systems Manager

Se você não precisar mais usar um recurso ou serviço que exija uma função vinculada ao serviço, é recomendável exclui-la. Dessa forma, você não terá uma entidade não utilizada que não seja monitorada ativamente ou mantida. Também é possível usar o console do IAM, a AWS CLI ou a API do IAM para excluir manualmente a função vinculada ao serviço. Para isso, primeiro você deve limpar manualmente os recursos de sua função vinculada ao serviço e, em seguida, excluí-la manualmente.

> **Note**
>
> Se o serviço Systems Manager estiver usando o perfil quando você tentar excluir etiquetas ou grupos de recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Para excluir recursos do Systems Manager usados por **AmazonSSMExplorerExportRole**

1. Para excluir etiquetas, consulte Adicionar e excluir etiquetas em um recurso individual.
2. Para excluir grupos de recursos, consulte Excluir grupos do AWS Resource Groups.

Para excluir manualmente a função vinculada ao serviço **AmazonSSMExplorerExportRole** usando o IAM

Use o console do IAM, a AWS CLI ou a API do IAM para excluir a função vinculada ao serviço `AmazonSSMExplorerExportRole`. Para obter mais informações, consulte Excluir uma função vinculada ao serviço no Guia do usuário do IAM.

Regiões compatíveis com o perfil vinculado ao serviço **AmazonSSMExplorerExportRole** do Systems Manager

O Systems Manager é compatível com a função vinculadas ao serviço `AmazonSSMExplorerExportRole` em todas as Regiões da AWS em que o serviço está disponível. Para obter mais informações, consulte Endpoints e cotas do AWS Systems Manager.

Uso de perfis para habilitar o acesso a nós just-in-time

O Systems Manager usa o perfil vinculado ao serviço denominado **`AWSServiceRoleForSystemsManagerJustInTimeAccess`**. O AWS Systems Manager usa esse perfil de serviço do IAM para permitir o acesso a nós just-in-time.

Permissões de perfil vinculado ao serviço para acesso a nós just-in-time do Systems Manager

O perfil vinculado ao serviço `AWSServiceRoleForSystemsManagerJustInTimeAccess` confia nos seguintes serviços para aceitar o perfil:

- `ssm.amazonaws.com`

A política de permissões da função permite que o Systems Manager conclua as seguintes ações nos recursos especificados:

- `ssm:CreateOpsItem`
- `ssm:GetOpsItem`
- `ssm:UpdateOpsItem`
- `ssm:DescribeOpsItems`
- `ssm:DescribeSessions`
- `ssm:ListTagsForResource`
- `ssm-guiconnect:ListConnections`
- `identitystore:ListGroupMembershipsForMember`
- `identitystore:DescribeUser`
- `identitystore:GetGroupId`
- `identitystore:GetUserId`
- `sso-directory:DescribeUsers`
- `sso-directory:IsMemberInGroup`
- `sso:ListInstances`
- `sso:DescribeRegisteredRegions`
- `sso:ListDirectoryAssociations`

- `ec2:DescribeTags`

A política gerida que é utilizada para fornecer permissões para o`AWSServiceRoleForSystemsManagerJustInTimeAccess`função é`AWSSystemsManagerEnableJustInTimeAccessPolicy`. Para obter detalhes sobre as permissões necessárias, consulte Política gerenciada pela AWS: AWSSystemsManagerJustInTimeAccessServicePolicy.

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua um perfil vinculado a serviço. Para obter mais informações, consulte Permissões de perfil vinculado ao serviço no Guia do usuário do IAM.

Criar uma função vinculada ao serviço **`AWSServiceRoleForSystemsManagerJustInTimeAccess`** para o Systems Manager

Não é necessário criar manualmente um perfil vinculado ao serviço. Quando você habilita o acesso a nós just-in-time no AWS Management Console, o Systems Manager cria o perfil vinculado ao serviço para você.

> **Important**
>
> Essa função vinculada ao serviço pode ser exibida em sua conta se você concluiu uma ação em outro serviço que usa os recursos compatíveis com essa função. Além disso, se você estava usando o serviço do Systems Manager antes de 19 de novembro de 2024, quando começou a compatibilidade com os perfis vinculados ao serviço, então o Systems Manager criou o perfil `AWSServiceRoleForSystemsManagerJustInTimeAccess` em sua conta. Para saber mais, consulte Uma nova função apareceu na minha conta do IAM.

Se excluir essa função vinculada ao serviço e precisar criá-la novamente, você pode usar esse mesmo processo para recriar a função na sua conta. Quando você habilita o acesso a nós just-in-time no AWS Management Console, o Systems Manager cria o perfil vinculado ao serviço para você novamente.

Você também pode usar o console do IAM para criar um perfil vinculado ao serviço com o caso de uso Perfil de serviço da AWS que permite que o Systems Manager habilite o acesso a nós

just-in-time. Na AWS CLI ou na API do AWS, crie um perfil vinculado a serviço com o nome de serviço `ssm.amazonaws.com`. Para obter mais informações, consulte Criar uma função vinculada ao serviço no Manual do usuário do IAM. Se você excluir essa função vinculada ao serviço, será possível usar esse mesmo processo para criar a função novamente.

Editar uma função vinculada ao serviço **`AWSServiceRoleForSystemsManagerJustInTimeAccess`** para o Systems Manager

O Systems Manager não permite que você edite a função vinculada a serviço `AWSServiceRoleForSystemsManagerJustInTimeAccess`. Depois de criar uma função vinculada a serviço, você não poderá alterar o nome da função, já que várias entidades poderão fazer referência à função. No entanto, você poderá editar a descrição do perfil usando o IAM. Para obter mais informações, consulte Editar um perfil vinculado ao serviço no Guia do usuário do IAM.

Excluir uma função vinculada ao serviço **`AWSServiceRoleForSystemsManagerJustInTimeAccess`** para o Systems Manager

Se você não precisar mais usar um atributo ou serviço que requer um perfil vinculado ao serviço, é recomendável excluí-la. Dessa forma, você não terá uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar os recursos de sua função vinculada ao serviço antes de excluí-la manualmente.

> **Note**
>
> Se o serviço Systems Manager estiver usando a função quando você tenta excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Para excluir manualmente a função vinculada ao serviço **`AWSServiceRoleForSystemsManagerJustInTimeAccess`** usando o IAM

Use o console do IAM, a AWS CLI ou a API da AWS para excluir a função vinculada ao serviço `AWSServiceRoleForSystemsManagerJustInTimeAccess`. Para obter mais informações, consulte Excluir um perfil vinculado ao serviço no Guia do usuário do IAM.

Regiões compatíveis com o perfil vinculado ao serviço **`AWSServiceRoleForSystemsManagerJustInTimeAccess`** do Systems Manager

Nome da Região da AWS	Identidade da região	Suporte no Systems Manager
Leste dos EUA (Norte da Virgínia)	us-east-1	Sim
Leste dos EUA (Ohio)	us-east-2	Sim
Oeste dos EUA (N. da Califórnia)	us-west-1	Sim
Oeste dos EUA (Oregon)	us-west-2	Sim
Ásia-Pacífico (Mumbai)	ap-south-1	Sim
Ásia Pacifico (Osaka)	ap-northeast-3	Sim
Ásia-Pacífico (Seul)	ap-northeast-2	Sim
Ásia-Pacífico (Singapura)	ap-southeast-1	Sim
Ásia-Pacífico (Sydney)	ap-southeast-2	Sim
Ásia-Pacífico (Tóquio)	ap-northeast-1	Sim
Canadá (Central)	ca-central-1	Sim
Europa (Frankfurt)	eu-central-1	Sim
Europa (Irlanda)	eu-west-1	Sim
Europa (Londres)	eu-west-2	Sim
Europa (Paris)	eu-west-3	Sim
Europa (Estocolmo)	eu-north-1	Sim
América do Sul (São Paulo)	sa-east-1	Sim
AWS GovCloud (US)	us-gov-west-1	Não

Uso de perfis para enviar notificações de solicitação de acesso a nós just-in-time

O Systems Manager usa o perfil vinculado ao serviço denominado **`AWSServiceRoleForSystemsManagerNotifications`**. O AWS Systems Manager usa esse perfil de serviço do IAM para enviar notificações aos aprovadores de solicitações de acesso.

Permissões de perfil vinculado ao serviço para notificações de acesso a nós just-in-time do Systems Manager

O perfil vinculado ao serviço `AWSServiceRoleForSystemsManagerNotifications` confia nos seguintes serviços para aceitar o perfil:

- `ssm.amazonaws.com`

A política de permissões da função permite que o Systems Manager conclua as seguintes ações nos recursos especificados:

- `identitystore:ListGroupMembershipsForMember`
- `identitystore:ListGroupMemberships`
- `identitystore:DescribeUser`
- `sso:ListInstances`
- `sso:DescribeRegisteredRegions`
- `sso:ListDirectoryAssociations`
- `sso-directory:DescribeUser`
- `sso-directory:ListMembersInGroup`
- `iam:GetRole`

A política gerida que é utilizada para fornecer permissões para o`AWSServiceRoleForSystemsManagerNotifications`função é`AWSSystemsManagerNotificationsServicePolicy`. Para obter detalhes sobre as permissões necessárias, consulte Política gerenciada pela AWS: AWSSystemsManagerNotificationsServicePolicy.

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua um perfil vinculado a serviço. Para obter mais informações, consulte Permissões de perfil vinculado ao serviço no Guia do usuário do IAM.

Criar uma função vinculada ao serviço **AWSServiceRoleForSystemsManagerNotifications** para o Systems Manager

Não é necessário criar manualmente um perfil vinculado ao serviço. Quando você habilita o acesso a nós just-in-time no AWS Management Console, o Systems Manager cria o perfil vinculado ao serviço para você.

> **Important**
>
> Essa função vinculada ao serviço pode ser exibida em sua conta se você concluiu uma ação em outro serviço que usa os recursos compatíveis com essa função. Além disso, se você estava usando o serviço do Systems Manager antes de 19 de novembro de 2024, quando começou a compatibilidade com os perfis vinculados ao serviço, então o Systems Manager criou o perfil `AWSServiceRoleForSystemsManagerNotifications` em sua conta. Para saber mais, consulte Uma nova função apareceu na minha conta do IAM.

Se excluir essa função vinculada ao serviço e precisar criá-la novamente, você pode usar esse mesmo processo para recriar a função na sua conta. Quando você habilita o acesso a nós just-in-time no AWS Management Console, o Systems Manager cria o perfil vinculado ao serviço para você novamente.

Você também pode usar o console do IAM para criar um perfil vinculado ao serviço com o caso de uso Perfil de serviço da AWS que permite que o Systems Manager envie notificações aos aprovadores de solicitações de acesso. Na AWS CLI ou na API do AWS, crie um perfil vinculado a serviço com o nome de serviço `ssm.amazonaws.com`. Para obter mais informações, consulte Criar uma função vinculada ao serviço no Manual do usuário do IAM. Se você excluir essa função vinculada ao serviço, será possível usar esse mesmo processo para criar a função novamente.

Editar uma função vinculada ao serviço **AWSServiceRoleForSystemsManagerNotifications** para o Systems Manager

O Systems Manager não permite que você edite a função vinculada a serviço `AWSServiceRoleForSystemsManagerNotifications`. Depois de criar uma função vinculada a

serviço, você não poderá alterar o nome da função, já que várias entidades poderão fazer referência à função. No entanto, você poderá editar a descrição do perfil usando o IAM. Para obter mais informações, consulte Editar um perfil vinculado ao serviço no Guia do usuário do IAM.

Excluir uma função vinculada ao serviço **AWSServiceRoleForSystemsManagerNotifications** para o Systems Manager

Se você não precisar mais usar um atributo ou serviço que requer um perfil vinculado ao serviço, é recomendável excluí-la. Dessa forma, você não terá uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar os recursos de sua função vinculada ao serviço antes de excluí-la manualmente.

> **Note**
>
> Se o serviço Systems Manager estiver usando a função quando você tenta excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Para excluir manualmente a função vinculada ao serviço **AWSServiceRoleForSystemsManagerNotifications** usando o IAM

Use o console do IAM, a AWS CLI ou a API da AWS para excluir a função vinculada ao serviço `AWSServiceRoleForSystemsManagerNotifications`. Para obter mais informações, consulte Excluir um perfil vinculado ao serviço no Guia do usuário do IAM.

Regiões compatíveis com o perfil vinculado ao serviço **AWSServiceRoleForSystemsManagerNotifications** do Systems Manager

Nome da Região da AWS	Identidade da região	Suporte no Systems Manager
Leste dos EUA (Norte da Virgínia)	us-east-1	Sim
Leste dos EUA (Ohio)	us-east-2	Sim
Oeste dos EUA (N. da Califórnia)	us-west-1	Sim
Oeste dos EUA (Oregon)	us-west-2	Sim

Nome da Região da AWS	Identidade da região	Suporte no Systems Manager
Ásia-Pacífico (Mumbai)	ap-south-1	Sim
Ásia Pacifico (Osaka)	ap-northeast-3	Sim
Ásia-Pacífico (Seul)	ap-northeast-2	Sim
Ásia-Pacífico (Singapura)	ap-southeast-1	Sim
Ásia-Pacífico (Sydney)	ap-southeast-2	Sim
Ásia-Pacífico (Tóquio)	ap-northeast-1	Sim
Canadá (Central)	ca-central-1	Sim
Europa (Frankfurt)	eu-central-1	Sim
Europa (Irlanda)	eu-west-1	Sim
Europa (Londres)	eu-west-2	Sim
Europa (Paris)	eu-west-3	Sim
Europa (Estocolmo)	eu-north-1	Sim
América do Sul (São Paulo)	sa-east-1	Sim
AWS GovCloud (US)	us-gov-west-1	Não

Registrar em log e monitorar no AWS Systems Manager

O monitoramento é uma parte importante da manutenção da confiabilidade, da disponibilidade e da performance do AWS Systems Manager e de suas soluções da AWS. É necessário coletar dados de monitoramento de todas as partes da solução da AWS para depurar uma falha de vários pontos com mais facilidade, caso ocorra. A AWS fornece várias ferramentas para monitorar os recursos do Systems Manager e responder a possíveis incidentes.

Logs do AWS CloudTrail

O CloudTrail fornece um registro de ações executadas por um usuário, uma função ou um AWS service (Serviço da AWS) no Systems Manager. Ao fazer uso das informações coletadas pelo CloudTrail, é possível determinar a solicitação feita a Systems Manager, o endereço IP no qual a solicitação foi feita, quem fez a solicitação e quando foi feita, além de detalhes adicionais. Para ter mais informações, consulte Registrar em log chamadas de API do AWS Systems Manager com o AWS CloudTrail.

Alarmes do Amazon CloudWatch

Usando alarmes do Amazon CloudWatch, você observa uma única métrica durante um período especificado para suas instâncias do Amazon Elastic Compute Cloud (Amazon EC2) e outros recursos. Se a métrica exceder determinado limite, uma notificação será enviada para um tópico do Amazon Simple Notification Service (Amazon SNS) ou para uma política do AWS Auto Scaling. Os alarmes do CloudWatch não invocam ações só porque estão em um determinado estado. O estado deve ter sido alterado e mantido por uma quantidade especificada de períodos. Para obter mais informações, consulte Uso de alarmes do Amazon CloudWatch no Manual do usuário do Amazon CloudWatch.

Painéis do Amazon CloudWatch

Os painéis do CloudWatch são páginas iniciais personalizáveis no console do CloudWatch que você pode usar para monitorar seus recursos em uma única visualização, mesmo os recursos distribuídos em Regiões da AWS diferentes. Você pode usar os painéis do CloudWatch para criar visualizações personalizadas das métricas e dos alarmes para os recursos da AWS. Para ter mais informações, consulte Usar painéis do Amazon CloudWatch hospedados pelo Systems Manager.

Amazon EventBridge

Usando o Amazon EventBridge, você pode configurar regras do para alertá-lo sobre alterações nos recursos do Systems Manager e direcionar o EventBridge para executar ações com base no conteúdo desses eventos. O EventBridge fornece suporte a uma série de eventos que são emitidos por várias ferramentas do Systems Manager. Para ter mais informações, consulte Monitorar eventos do Systems Manager com o Amazon EventBridge.

Amazon CloudWatch Logs e logs do SSM Agent

O SSM Agent grava informações sobre execuções, ações programadas, erros e status de integridade em arquivos de log para cada nó. É possível visualizar arquivos de log conectando-

se manualmente a um nó. Recomendamos enviar automaticamente dados de log do agente para um grupo de log no CloudWatch Logs para análise. Para ter mais informações, consulte Enviar logs de nós para o CloudWatch Logs unificado (agente do CloudWatch) e Visualizar logs do SSM Agent.

AWS Systems Manager Compatibilidade

Você pode usar o Compliance, uma ferramenta do AWS Systems Manager, para verificar a conformidade dos patches e as inconsistências de configuração em sua frota de nós gerenciados. Você pode coletar e agregar dados de várias Contas da AWS e Regiões da AWS e depois fazer buscas detalhadas em recursos específicos que não forem compatíveis. Por padrão, o Compliance exibe dados de conformidade atuais sobre a aplicação de patches no Patch Manager, uma ferramenta do AWS Systems Manager e associações no State Manager, uma ferramenta do AWS Systems Manager. Para ter mais informações, consulte AWS Systems Manager Compliance.

AWS Systems Manager Explorer

O Explorer, uma ferramenta do AWS Systems Manager, é um painel de operações personalizável que relata informações sobre seus recursos da AWS. O Explorer exibe uma visualização agregada dos dados de operações (OpsData) para suas Contas da AWS e em todas as Regiões da AWS. No Explorer, os OpsData incluem metadados sobre suas instâncias do EC2, detalhes de conformidade de patches e itens de trabalho operacionais (OpsItems). O Explorer fornece contexto sobre como os OpsItems são distribuídos em suas unidades de negócios ou aplicativos, a tendência ao longo do tempo e como eles variam de acordo com a categoria. Você pode agrupar e filtrar informações no Explorer para se concentrar em itens que são relevantes para você e que exigem ação. Para ter mais informações, consulte AWS Systems Manager Explorer.

AWS Systems Manager OpsCenter

O OpsCenter, uma ferramenta do AWS Systems Manager, fornece um local central onde engenheiros de operações e profissionais de TI podem visualizar, investigar e resolver itens de trabalho operacionais (OpsItems) relacionados aos recursos da AWS. O OpsCenter agrega e padroniza OpsItems em todos os serviços, fornecendo dados de investigação contextual sobre cada OpsItem, OpsItems relacionados e recursos relacionados. O OpsCenter também fornece runbooks do Automation, uma ferramenta do AWS Systems Manager que pode ser usada para resolver problemas rapidamente. O OpsCenter é integrado ao Amazon EventBridge. Isso significa que você pode criar regras do EventBridge que criam automaticamente OpsItems para qualquer AWS service (Serviço da AWS) que publique eventos no EventBridge. Para ter mais informações, consulte AWS Systems Manager OpsCenter.

Amazon Simple Notification Service

É possível configurar o Amazon Simple Notification Service (Amazon SNS) para enviar notificações sobre o status dos comandos que você envia usando o Run Command ou Maintenance Windows, ambos ferramentas do AWS Systems Manager. O Amazon SNS coordena e gerencia a entrega e o envio de notificações a clientes e endpoints que assinam tópicos do Amazon SNS. Você pode receber uma notificação sempre que um comando muda para um novo estado ou atinge um estado específico, como `Failed` ou `Timed Out`. Nos casos em que você envia um comando para vários nós, você pode receber uma notificação para cada cópia do comando enviado para um nó específico. Para ter mais informações, consulte Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS.

AWS Trusted Advisor e AWS Health Dashboard

O Trusted Advisor conta com as práticas recomendadas aprendidas com o atendimento a centenas de milhões de clientes da AWS. O Trusted Advisor inspeciona seu ambiente da AWS e faz recomendações quando há oportunidades para economizar dinheiro, melhorar a performance e a disponibilidade do sistema e ajuda a corrigir falhas de segurança. Todos os clientes da AWS têm acesso a cinco verificações do Trusted Advisor. Os clientes com um plano de suporte AWS Support Business ou Enterprise podem ver todas as verificações do Trusted Advisor. Para obter mais informações, consulte AWS Trusted Advisor no Guia do usuário doAWS Support e o Guia do usuário do .AWS Health

Mais informações

- Registrar em log e monitorar no AWS Systems Manager

Validação de conformidade do AWS Systems Manager

Este tópico aborda a conformidade do AWS Systems Manager com programas de garantia de terceiros. Para obter informações sobre como visualizar dados de conformidade para os nós gerenciados, consulte AWS Systems Manager Compliance.

Auditores de terceiros avaliam a segurança e a conformidade do Systems Manager como parte de vários programas de conformidade da AWS. Isso inclui SOC, PCI, FedRAMP, HIPAA e outros.

Para obter uma lista de Serviços da AWS no escopo de programas de conformidade específicos, consulte Serviços da AWS no escopo pelo programa de conformidade. Para obter informações gerais, consulte Programas de Conformidade da AWS.

É possível fazer download de relatórios de auditoria de terceiros usando o AWS Artifact. Para obter mais informações, consulte Baixar relatórios no AWS Artifact.

Sua responsabilidade de conformidade ao usar o Systems Manager é determinada pela confidencialidade dos seus dados, pelos objetivos de conformidade da sua empresa e pelos regulamentos e leis aplicáveis. A AWS fornece os seguintes recursos para ajudar com a conformidade:

- Guias de início rápido de segurança e compatibilidade: esses guias de implantação abordam as considerações de arquitetura e fornecem etapas para implantação de ambientes de linha de base focados em conformidade e segurança na AWS.
- Whitepaper Architecting for HIPAA Security and Compliance (Arquitetura para segurança e conformidade com a HIPAA): este whitepaper descreve como as empresas podem usar a AWS para criar aplicações em conformidade com a HIPAA.
- Recursos de conformidade da AWS: essa coleção de manuais e guias pode ser aplicada a seu setor e local.
- Avaliar atributos com regras no Guia do desenvolvedor do AWS Config: o serviço AWS Config avalia como as configurações de atributos estão em conformidade com práticas internas, diretrizes do setor e regulamentos.
- AWS Security Hub: esse AWS service (Serviço da AWS) fornece uma visão abrangente do estado de sua segurança na AWS que ajuda você a conferir sua conformidade com padrões e práticas recomendadas de segurança do setor.

Resiliência no AWS Systems Manager

A infraestrutura global da AWS se baseia em Regiões da AWS e zonas de disponibilidade. A Regiões da AWS oferece várias zonas de disponibilidade separadas e isoladas fisicamente que são conectadas com baixa latência, throughputs elevadas e em redes altamente redundantes. Com as zonas de disponibilidade, é possível projetar e operar aplicativos e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre Regiões da AWS e zonas de disponibilidade, consulte Infraestrutura global da AWS.

Segurança da infraestrutura no AWS Systems Manager

Por ser um serviço gerenciado, o AWS Systems Manager é protegido pela segurança da rede global da AWS. Para obter informações sobre serviços de segurança da AWS e como a AWS protege a infraestrutura, consulte Segurança na Nuvem AWS. Para projetar seu ambiente da AWS usando as práticas recomendadas de segurança da infraestrutura, consulte Proteção de Infraestrutura em Pilar de Segurança: AWS Well-Architected Framework.

Você usa Systems Manager chamadas de API publicadas pela para acessarAWS o por meio da rede. Os clientes devem oferecer suporte para:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com perfect forward secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou você pode usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Análise de vulnerabilidade e configuração no AWS Systems Manager

A AWS cuida das tarefas básicas de segurança, como configuração de firewall e recuperação de desastres. Esses procedimentos foram revisados e certificados por terceiros certificados. Para obter mais detalhes, consulte os seguintes recursos da :

- Validação de conformidade do AWS Systems Manager
- Modelo de responsabilidade compartilhada
- Práticas recomendadas de segurança, identidade e conformidade

Melhores práticas de segurança do Systems Manager

O AWS Systems Manager oferece uma série de recursos de segurança a serem considerados no desenvolvimento e na implementação das suas próprias políticas de segurança. As práticas

recomendadas a seguir são diretrizes gerais e não representam uma solução completa de segurança. Como essas práticas recomendadas podem não ser adequadas ou suficientes para o seu ambiente, trate-as como considerações úteis em vez de prescrições.

Tópicos

- Melhores práticas de segurança preventivas do Systems Manager
- Melhores práticas de auditoria e monitoramento do Systems Manager

Melhores práticas de segurança preventivas do Systems Manager

As seguintes práticas recomendadas do Systems Manager podem ajudar a evitar incidentes de segurança.

Implemente o acesso de privilégio mínimo

Ao conceder permissões, você decide quem receberá quais permissões para quais recursos do Systems Manager. Você permite ações específicas que quer permitir nesses recursos. Portanto, é necessário conceder somente as permissões necessárias para executar uma tarefa. A implementação do privilégio de acesso mínimo é fundamental para reduzir o risco de segurança e o impacto que pode resultar de erros ou usuários mal-intencionados.

As ferramentas a seguir estão disponíveis para implementar o acesso de privilégio mínimo:

- Políticas do IAM e Limites de permissões para entidades do IAM
- Políticas de controle de serviço

Use as configurações recomendadas para o SSM Agent quando configurado para usar um proxy

Se você configurar o SSM Agent para usar um proxy, use a variável `no_proxy` com o endereço IP do serviço de metadados da instância do Systems Manager para garantir que as chamadas para o Systems Manager não assumam a identidade do serviço de proxy.

Para ter mais informações, consulte Configurar o SSM Agent para usar um proxy em nós do Linux e Configurar o SSM Agent para usar um proxy para instâncias do Windows Server.

Use os parâmetros SecureString para criptografar e proteger dados secretos

No Parameter Store, uma ferramenta do AWS Systems Manager, um parâmetro `SecureString` representa quaisquer dados sigilosos que precisam ser armazenados e referenciados com segurança. Se você tiver dados que não deseja que os usuários alterem ou façam referência em

texto simples, como senhas ou chaves de licença, crie esses parâmetros usando o tipo de dados `SecureString`. O Parameter Store usa uma AWS KMS key no AWS Key Management Service (AWS KMS) para criptografar o valor do parâmetro. O AWS KMS usa uma chave gerenciada pelo cliente ou uma Chave gerenciada pela AWS ao criptografar o valor do parâmetro. Para segurança máxima, recomendamos usar sua própria chave do KMS. Se você usar a Chave gerenciada pela AWS, qualquer usuário com permissão para executar as ações GetParameter e GetParameters em sua conta poderá visualizar ou recuperar o conteúdo de todos os parâmetros `SecureString`. Se você estiver usando chaves gerenciadas pelo cliente para criptografar seus valores `SecureString` de segurança, será possível usar políticas de chave e políticas do IAM para gerenciar permissões para criptografar e descriptografar parâmetros.

É mais difícil estabelecer políticas de controle de acesso para essas operações ao usar uma Chave gerenciada pela AWS. Por exemplo, se você usar uma Chave gerenciada pela AWS para criptografar parâmetros `SecureString` e não quiser que os usuários trabalhem com parâmetros `SecureString`, as políticas do IAM dos usuários devem negar explicitamente o acesso à chave padrão.

Para obter mais informações, consulte Restringir o acesso a parâmetros do Parameter Store usando políticas do IAM e Como o AWS Systems ManagerParameter Store usa o AWS KMS no Manual do desenvolvedor do AWS Key Management Service.

Defina allowedValues e allowedPattern para parâmetros do documento

Você pode validar a entrada do usuário de parâmetros de documentos do Systems Manager (documentos SSM) definindo `allowedValues` e `allowedPattern`. Para `allowedValues`, defina uma matriz de valores permitidos para o parâmetro. Se um usuário inserir um valor que não é permitido, a execução falhará ao iniciar. Para `allowedPattern`, defina uma expressão regular que valida se a entrada do usuário corresponde ao padrão definido para o parâmetro. Se a entrada do usuário não corresponder ao padrão permitido, a execução não será iniciada.

Para obter mais informações sobre `allowedValues` e `allowedPattern`, consulte Elementos e parâmetros de dados.

Bloquear compartilhamento público de documentos

A menos que seu caso de uso exija que o compartilhamento público seja permitido, recomendamos ativar a configuração de bloqueio de compartilhamento público para seus documentos do SSM na seção Preferências do console de documentos do Systems Manager.

Usar endpoints da Amazon Virtual Private Cloud (Amazon VPC) e VPC

É possível usar a Amazon VPC para executar os recursos da AWS em uma rede virtual definida por você. Essa rede virtual se assemelha a uma rede tradicional que você operaria no seu data center, com os benefícios de usar a infraestrutura dimensionável da AWS.

Ao implementar um endpoint da VPC, você pode conectar de forma privada a VPC aos Serviços da AWS compatíveis e aos serviços do endpoint da VPC desenvolvidos pelo AWS PrivateLink sem exigir um gateway da Internet, um dispositivo NAT, uma conexão VPN ou uma conexão do AWS Direct Connect. As instâncias na sua VPC não exigem que endereços IP públicos se comuniquem com recursos no serviço. O tráfego entre a sua VPC e os outros serviços não sai da rede da Amazon.

Para obter mais informações sobre a segurança da Amazon VPC, consulte Melhorar a segurança das instâncias do EC2 usando endpoints da VPC para o Systems Manager e Privacidade do tráfego de redes no Amazon VPC no Guia do usuário do Amazon VPC.

Restrinja usuários do Session Manager para sessões usando comandos interativos e documentos de sessão do SSM específicos

O Session Manager, uma ferramenta do AWS Systems Manager, fornece vários métodos para iniciar sessões em seus nós gerenciados. Para as conexões mais seguras, é possível exigir que os usuários se conectem usando o método de comandos interativos para limitar a interação do usuário a uma sequência de comandos ou um comando específico. Isso ajuda a gerenciar as ações interativas que um usuário pode executar. Para ter mais informações, consulte Iniciar uma sessão (comandos interativos e não interativos).

Para reforçar a segurança, você pode limitar o acesso do Session Manager a instâncias específicas do Amazon EC2 e a documentos de sessão do Session Manager específicos. Você concede ou revoga o acesso do Session Manager dessa forma usando políticas do AWS Identity and Access Management (IAM). Para ter mais informações, consulte Etapa 3: controlar o acesso da sessão pelos nós gerenciados.

Forneça permissões de nó temporárias para fluxos de trabalho de automação

Durante um fluxo de trabalho do Automation, uma ferramenta do AWS Systems Manager, os nós podem precisar de permissões necessárias apenas para essa execução, mas não para outras operações do Systems Manager. Por exemplo, um fluxo de trabalho do Automation pode exigir que um nó gerenciado chame uma operação específica da API ou acesse um recurso da AWS especificamente durante o fluxo de trabalho. Se essas chamadas ou recursos forem aqueles aos quais você deseja limitar o acesso, é possível fornecer permissões temporárias e

complementares para os nós dentro do próprio runbook do Automation, em vez de adicionar as permissões ao seu perfil da instância do IAM. No final do fluxo de trabalho da automação, as permissões temporárias serão removidas. Para obter mais informações, consulte Fornecer permissões de instância temporária com automações do AWS Systems Manager no Blog de gerenciamento e governança da AWS.

Mantenha as ferramentas da AWS e do Systems Manager atualizadas

A AWS lança regularmente versões atualizadas de ferramentas e plugins que é possível usar em suas operações da AWS e do Systems Manager. Manter esses recursos atualizados garante que os usuários e nós em sua conta tenham acesso às funcionalidades e recursos de segurança mais recentes nessas ferramentas.

- SSM Agent – AWS Systems Manager Agent (SSM Agent) é um software da Amazon que pode ser instalado e configurado em uma instância do Amazon Elastic Compute Cloud (Amazon EC2), em um servidor on-premises ou em uma máquina virtual (VM). O SSM Agent possibilita que o Systems Manager atualize, gerencie e configure esses recursos. Recomendamos verificar se há novas versões ou automatizar atualizações para o agente pelo menos a cada duas semanas. Para ter mais informações, consulte Automatizar atualizações do SSM Agent. Também recomendamos verificar a assinatura do SSM Agent como parte do processo de atualização. Para ter mais informações, consulte Verificar a assinatura do SSM Agent.
- AWS CLI: a AWS Command Line Interface (AWS CLI) é uma ferramenta de código aberto que permite interagir com os Serviços da AWS usando comandos no shell da linha de comando. Para atualizar a AWS CLI, execute o mesmo comando usado para instalar a AWS CLI. Recomendamos criar uma tarefa programada em sua máquina local para executar o comando adequado para o sistema operacional pelo menos uma vez a cada duas semanas. Para obter informações sobre como instalar comandos, consulte Instalar a versão 2 da AWS CLI no Guia do usuário da AWS Command Line Interface.
- AWS Tools for Windows PowerShell: o Tools for Windows PowerShell é um conjunto de módulos do PowerShell, criado com base na funcionalidade exposta pelo AWS SDK para .NET. O AWS Tools for Windows PowerShell permite que você faça script de operações em seus recursos da AWS na linha de comando do PowerShell. Periodicamente, à medida que versões atualizadas do Tools for Windows PowerShell são lançadas, você deve atualizar a versão que está sendo executada localmente. Para obter mais informações, consulte Atualizar o AWS Tools for Windows PowerShell no Windows ou Atualizar o AWS Tools for Windows PowerShell no Linux ou macOS no Guia do usuário do IAM Policy Simulator.
- Plugin do Session Manager: se os usuários em sua organização com permissões para usar o Session Manager quiserem se conectar a um nó usando a AWS CLI, eles deverão primeiro

instalar o plugin do Session Manager em suas máquinas locais. Para atualizar o plugin, execute o mesmo comando usado para instalar o plugin. Recomendamos criar uma tarefa programada em sua máquina local para executar o comando adequado para o sistema operacional pelo menos uma vez a cada duas semanas. Para ter mais informações, consulte Instalar o plug-in do Session Manager para a AWS CLI.

- Agente do CloudWatch: você pode configurar e usar o agente do CloudWatch para coletar métricas e logs das instâncias do EC2, instâncias on-premises e máquinas virtuais (VMs). Esses logs podem ser enviados para o Amazon CloudWatch Logs para monitoramento e análise. Recomendamos verificar se há novas versões ou automatizar atualizações para o agente pelo menos a cada duas semanas. Para obter as atualizações mais simples, use a Configuração rápida do AWS Systems Manager. Para ter mais informações, consulte AWS Systems Manager Quick Setup.

Melhores práticas de auditoria e monitoramento do Systems Manager

As práticas recomendadas a seguir para o Systems Manager podem ajudar a detectar pontos fracos e incidentes potenciais de segurança.

Identificar e auditar todos os seus recursos do Systems Manager

A identificação de seus ativos de TI é um aspecto essencial de governança e segurança. É necessário identificar todos os seus recursos do Systems Manager para avaliar sua postura de segurança e agir em possíveis áreas de pontos fracos.

Use o Tag Editor para identificar recursos sensíveis quanto a segurança ou auditoria, depois use essas tags quando precisar procurar por esses recursos. Para obter mais informações, consulte Localizar recursos para etiquetar no Guia do usuário do AWS Resource Groups.

Crie grupos de recursos para seus recursos do Systems Manager. Para obter mais informações, consulte O que são grupos de recursos?

Implementar monitoramento usando ferramentas de monitoramento do Amazon CloudWatch

O monitoramento é uma parte importante da manutenção da confiabilidade, da segurança, da disponibilidade e da performance do Systems Manager e das suas soluções da AWS. O Amazon CloudWatch fornece várias ferramentas e serviços para ajudar você a monitorar o Systems Manager e seus outros Serviços da AWS. Para ter mais informações, consulte Enviar logs de nós para o CloudWatch Logs unificado (agente do CloudWatch) e Monitorar eventos do Systems Manager com o Amazon EventBridge.

Uso do CloudTrail

O AWS CloudTrail fornece um registro das ações executadas por um usuário, uma função ou um AWS service (Serviço da AWS) no Systems Manager. Ao fazer uso das informações coletadas pelo CloudTrail, é possível determinar a solicitação feita a Systems Manager, o endereço IP no qual a solicitação foi feita, quem fez a solicitação e quando foi feita, além de detalhes adicionais. Para ter mais informações, consulte Registrar em log chamadas de API do AWS Systems Manager com o AWS CloudTrail.

Ativar o AWS Config

O AWS Config permite analisar, auditar e avaliar as configurações dos recursos da AWS. O AWS Config monitora as configurações de recursos, possibilitando que você avalie as configurações registradas em relação às configurações seguras requeridas. Com o AWS Config, você pode analisar alterações feitas nas configurações e relacionamentos entre os recursos da AWS, examinar os detalhes do histórico de configuração de recursos e determinar a conformidade geral em relação às configurações especificadas em diretrizes internas. Isso pode ajudar a simplificar a auditoria de conformidade, a análise de segurança, o gerenciamento de alterações e a solução de problemas operacionais. Para obter mais informações, consulte Configuração do AWS Config com o console no Guia do desenvolvedor do AWS Config. Ao especificar os tipos de recurso que devem ser gravados, inclua os recursos do Systems Manager.

Monitorar as recomendações de segurança da AWS

Verifique regularmente as recomendações de segurança publicadas no Trusted Advisor para sua Conta da AWS. Você pode fazer isso programaticamente usando describe-trusted-advisor-checks.

Além disso, monitore ativamente o endereço de e-mail registrado como principal para cada uma de suas Contas da AWS. A AWS usará esse e-mail para entrar em contato e notificá-lo sobre os problemas de segurança que surgirem e que possam afetar você.

Problemas operacionais da AWS com grande impacto são publicados no AWS Service Health Dashboard. Problemas operacionais também são publicados em contas individuais por meio do Personal Health Dashboard. Para obter mais informações, consulte a documentação da AWS Health.

Mais informações

- Práticas recomendadas de segurança, identidade e conformidade

- Getting Started: Follow Security Best Practices as You Configure Your AWS Resources (Blog de segurança da AWS)
- Práticas recomendadas de segurança no IAM
- Práticas recomendadas de segurança no AWS CloudTrail
- Práticas recomendadas de segurança para o Amazon S3
- Práticas recomendadas de segurança para o AWS Key Management Service

Exemplos de código para o Systems Manager usando SDKs da AWS

Os exemplos de código a seguir mostram como usar o Systems Manager com um kit de desenvolvimento de software (SDK) da AWS.

As noções básicas são exemplos de código que mostram como realizar as operações essenciais em um serviço.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, é possível ver as ações no contexto em seus cenários relacionados.

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Conceitos básicos

Hello Systems Manager

Os exemplos de código a seguir mostram como começar a usar o Systems Manager.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.ssm.SsmClient;
import software.amazon.awssdk.services.ssm.model.DocumentFilter;
import software.amazon.awssdk.services.ssm.model.ListDocumentsRequest;
import software.amazon.awssdk.services.ssm.model.ListDocumentsResponse;

public class HelloSSM {
```

```
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <awsAccount>

                Where:
                    awsAccount - Your AWS Account number.
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String awsAccount = args[0] ;
        Region region = Region.US_EAST_1;
        SsmClient ssmClient = SsmClient.builder()
            .region(region)
            .build();

        listDocuments(ssmClient, awsAccount);
    }

    /*
    This code automatically fetches the next set of results using the `nextToken`
and
    stops once the desired maxResults (20 in this case) have been reached.
    */
    public static void listDocuments(SsmClient ssmClient, String awsAccount) {
        String nextToken = null;
        int totalDocumentsReturned = 0;
        int maxResults = 20;
        do {
            ListDocumentsRequest request = ListDocumentsRequest.builder()
                .documentFilterList(
                    DocumentFilter.builder()
                        .key("Owner")
                        .value(awsAccount)
                        .build()
                    )
                .maxResults(maxResults)
                .nextToken(nextToken)
```

```
                .build();

            ListDocumentsResponse response = ssmClient.listDocuments(request);
            response.documentIdentifiers().forEach(identifier ->
 System.out.println("Document Name: " + identifier.name()));
            nextToken = response.nextToken();
            totalDocumentsReturned += response.documentIdentifiers().size();
        } while (nextToken != null && totalDocumentsReturned < maxResults);
    }
}
```

- Para obter detalhes da API, consulte ListDocuments na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
import { paginateListDocuments, SSMClient } from "@aws-sdk/client-ssm";

// Call ListDocuments and display the result.
export const main = async () => {
  const client = new SSMClient();
  const listDocumentsPaginated = [];
  console.log(
    "Hello, AWS Systems Manager! Let's list some of your documents:\n",
  );
  try {
    // The paginate function is a wrapper around the base command.
    const paginator = paginateListDocuments({ client }, { MaxResults: 5 });
    for await (const page of paginator) {
      listDocumentsPaginated.push(...page.DocumentIdentifiers);
    }
  } catch (caught) {
```

```
    console.error(`There was a problem saying hello: ${caught.message}`);
    throw caught;
  }

  for (const { Name, DocumentFormat, CreatedDate } of listDocumentsPaginated) {
    console.log(`${Name} - ${DocumentFormat} - ${CreatedDate}`);
  }
};

// Call function if run directly.
import { fileURLToPath } from "node:url";
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main();
}
```

- Para obter detalhes da API, consulte ListDocuments na Referência da API do AWS SDK para JavaScript.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
import boto3
from botocore.exceptions import ClientError


def hello_systems_manager(ssm_client):
    """
    Use the AWS SDK for Python (Boto3) to create an AWS Systems Manager
    client and list the first 5 documents in your account.
    This example uses the default settings specified in your shared credentials
    and config files.
```

```
    :param ssm_client: A Boto3 AWS Systems Manager Client object. This object
 wraps
                            the low-level AWS Systems Manager service API.
    """
    print("Hello, AWS Systems Manager! Let's list some of your documents:\n")

    paginator = ssm_client.get_paginator("list_documents")
    page_iterator = paginator.paginate(PaginationConfig={"MaxItems": 5})
    for page in page_iterator:
        for document in page["DocumentIdentifiers"]:
            print(f"  {document['Name']}")


if __name__ == "__main__":
    try:
        hello_systems_manager(boto3.client("ssm"))
    except ClientError as err:
        print("Hello systems manager had an error.")
        print(err.response["Error"]["Code"])
        print(err.response["Error"]["Message"])
```

- Para obter detalhes da API, consulte ListDocuments na Referência da API do AWS SDK para Python (Boto3).

Exemplos de código

- Exemplos básicos para o Systems Manager usando SDKs da AWS
  - Hello Systems Manager
  - Conheça os conceitos básicos do Systems Manager com um SDK da AWS
  - Ações para o Systems Manager usando SDKs da AWS
    - Usar AddTagsToResource com uma CLI
    - Usar CancelCommand com uma CLI
    - Usar CreateActivation com uma CLI
    - Usar CreateAssociation com uma CLI
    - Usar CreateAssociationBatch com uma CLI
    - Usar CreateDocument com o AWS SDK ou a CLI
    - Usar CreateMaintenanceWindow com o AWS SDK ou a CLI

- Usar CreateOpsItem com o AWS SDK ou a CLI
- Usar CreatePatchBaseline com uma CLI
- Usar DeleteActivation com uma CLI
- Usar DeleteAssociation com uma CLI
- Usar DeleteDocument com o AWS SDK ou a CLI
- Usar DeleteMaintenanceWindow com o AWS SDK ou a CLI
- Usar DeleteOpsItem com um SDK da AWS
- Usar DeleteParameter com uma CLI
- Usar DeletePatchBaseline com uma CLI
- Usar DeregisterManagedInstance com uma CLI
- Usar DeregisterPatchBaselineForPatchGroup com uma CLI
- Usar DeregisterTargetFromMaintenanceWindow com uma CLI
- Usar DeregisterTaskFromMaintenanceWindow com uma CLI
- Usar DescribeActivations com uma CLI
- Usar DescribeAssociation com uma CLI
- Usar DescribeAssociationExecutionTargets com uma CLI
- Usar DescribeAssociationExecutions com uma CLI
- Usar DescribeAutomationExecutions com uma CLI
- Usar DescribeAutomationStepExecutions com uma CLI
- Usar DescribeAvailablePatches com uma CLI
- Usar DescribeDocument com uma CLI
- Usar DescribeDocumentPermission com uma CLI
- Usar DescribeEffectiveInstanceAssociations com uma CLI
- Usar DescribeEffectivePatchesForPatchBaseline com uma CLI
- Usar DescribeInstanceAssociationsStatus com uma CLI
- Usar DescribeInstanceInformation com uma CLI
- Usar DescribeInstancePatchStates com uma CLI
- Usar DescribeInstancePatchStatesForPatchGroup com uma CLI
- Usar DescribeInstancePatches com uma CLI

- Usar DescribeMaintenanceWindowExecutionTaskInvocations com uma CLI

- Usar DescribeMaintenanceWindowExecutionTasks com uma CLI
- Usar DescribeMaintenanceWindowExecutions com uma CLI
- Usar DescribeMaintenanceWindowTargets com uma CLI
- Usar DescribeMaintenanceWindowTasks com uma CLI
- Usar DescribeMaintenanceWindows com uma CLI
- Usar DescribeOpsItems com o AWS SDK ou a CLI
- Usar DescribeParameters com o AWS SDK ou a CLI
- Usar DescribePatchBaselines com uma CLI
- Usar DescribePatchGroupState com uma CLI
- Usar DescribePatchGroups com uma CLI
- Usar GetAutomationExecution com uma CLI
- Usar GetCommandInvocation com uma CLI
- Usar GetConnectionStatus com uma CLI
- Usar GetDefaultPatchBaseline com uma CLI
- Usar GetDeployablePatchSnapshotForInstance com uma CLI
- Usar GetDocument com uma CLI
- Usar GetInventory com uma CLI
- Usar GetInventorySchema com uma CLI
- Usar GetMaintenanceWindow com uma CLI
- Usar GetMaintenanceWindowExecution com uma CLI
- Usar GetMaintenanceWindowExecutionTask com uma CLI
- Usar GetParameter com o AWS SDK ou a CLI
- Usar GetParameterHistory com uma CLI
- Usar GetParameters com uma CLI
- Usar GetPatchBaseline com uma CLI
- Usar GetPatchBaselineForPatchGroup com uma CLI
- Usar ListAssociationVersions com uma CLI
- Usar ListAssociations com uma CLI
- Usar ListCommandInvocations com o AWS SDK ou a CLI
- Usar ListCommands com uma CLI

- Usar ListComplianceItems com uma CLI
- Usar ListComplianceSummaries com uma CLI
- Usar ListDocumentVersions com uma CLI
- Usar ListDocuments com uma CLI
- Usar ListInventoryEntries com uma CLI
- Usar ListResourceComplianceSummaries com uma CLI
- Usar ListTagsForResource com uma CLI
- Usar ModifyDocumentPermission com uma CLI
- Usar PutComplianceItems com uma CLI
- Usar PutInventory com uma CLI
- Usar PutParameter com o AWS SDK ou a CLI
- Usar RegisterDefaultPatchBaseline com uma CLI
- Usar RegisterPatchBaselineForPatchGroup com uma CLI
- Usar RegisterTargetWithMaintenanceWindow com uma CLI
- Usar RegisterTaskWithMaintenanceWindow com uma CLI
- Usar RemoveTagsFromResource com uma CLI
- Usar SendCommand com o AWS SDK ou a CLI
- Usar StartAutomationExecution com uma CLI
- Usar StartSession com uma CLI
- Usar StopAutomationExecution com uma CLI
- Usar UpdateAssociation com uma CLI
- Usar UpdateAssociationStatus com uma CLI
- Usar UpdateDocument com uma CLI
- Usar UpdateDocumentDefaultVersion com uma CLI
- Usar UpdateMaintenanceWindow com o AWS SDK ou a CLI
- Usar UpdateManagedInstanceRole com uma CLI
- Usar UpdateOpsItem com o AWS SDK ou a CLI
- Usar UpdatePatchBaseline com uma CLI

Exemplos básicos para o Systems Manager usando SDKs da AWS

Os exemplos de código a seguir mostram como usar os conceitos básicos do AWS Systems Manager com SDKs da AWS.

Exemplos

- Hello Systems Manager
- Conheça os conceitos básicos do Systems Manager com um SDK da AWS
- Ações para o Systems Manager usando SDKs da AWS
  - Usar AddTagsToResource com uma CLI
  - Usar CancelCommand com uma CLI
  - Usar CreateActivation com uma CLI
  - Usar CreateAssociation com uma CLI
  - Usar CreateAssociationBatch com uma CLI
  - Usar CreateDocument com o AWS SDK ou a CLI
  - Usar CreateMaintenanceWindow com o AWS SDK ou a CLI
  - Usar CreateOpsItem com o AWS SDK ou a CLI
  - Usar CreatePatchBaseline com uma CLI
  - Usar DeleteActivation com uma CLI
  - Usar DeleteAssociation com uma CLI
  - Usar DeleteDocument com o AWS SDK ou a CLI
  - Usar DeleteMaintenanceWindow com o AWS SDK ou a CLI
  - Usar DeleteOpsItem com um SDK da AWS
  - Usar DeleteParameter com uma CLI
  - Usar DeletePatchBaseline com uma CLI
  - Usar DeregisterManagedInstance com uma CLI
  - Usar DeregisterPatchBaselineForPatchGroup com uma CLI
  - Usar DeregisterTargetFromMaintenanceWindow com uma CLI
  - Usar DeregisterTaskFromMaintenanceWindow com uma CLI
  - Usar DescribeActivations com uma CLI
  - Usar DescribeAssociation com uma CLI
  - Usar DescribeAssociationExecutionTargets com uma CLI

- Usar DescribeAssociationExecutions com uma CLI
- Usar DescribeAutomationExecutions com uma CLI
- Usar DescribeAutomationStepExecutions com uma CLI
- Usar DescribeAvailablePatches com uma CLI
- Usar DescribeDocument com uma CLI
- Usar DescribeDocumentPermission com uma CLI
- Usar DescribeEffectiveInstanceAssociations com uma CLI
- Usar DescribeEffectivePatchesForPatchBaseline com uma CLI
- Usar DescribeInstanceAssociationsStatus com uma CLI
- Usar DescribeInstanceInformation com uma CLI
- Usar DescribeInstancePatchStates com uma CLI
- Usar DescribeInstancePatchStatesForPatchGroup com uma CLI
- Usar DescribeInstancePatches com uma CLI
- Usar DescribeMaintenanceWindowExecutionTaskInvocations com uma CLI
- Usar DescribeMaintenanceWindowExecutionTasks com uma CLI
- Usar DescribeMaintenanceWindowExecutions com uma CLI
- Usar DescribeMaintenanceWindowTargets com uma CLI
- Usar DescribeMaintenanceWindowTasks com uma CLI
- Usar DescribeMaintenanceWindows com uma CLI
- Usar DescribeOpsItems com o AWS SDK ou a CLI
- Usar DescribeParameters com o AWS SDK ou a CLI
- Usar DescribePatchBaselines com uma CLI
- Usar DescribePatchGroupState com uma CLI
- Usar DescribePatchGroups com uma CLI
- Usar GetAutomationExecution com uma CLI
- Usar GetCommandInvocation com uma CLI
- Usar GetConnectionStatus com uma CLI
- Usar GetDefaultPatchBaseline com uma CLI
- Usar GetDeployablePatchSnapshotForInstance com uma CLI

- Usar GetDocument com uma CLI

- Usar GetInventory com uma CLI
- Usar GetInventorySchema com uma CLI
- Usar GetMaintenanceWindow com uma CLI
- Usar GetMaintenanceWindowExecution com uma CLI
- Usar GetMaintenanceWindowExecutionTask com uma CLI
- Usar GetParameter com o AWS SDK ou a CLI
- Usar GetParameterHistory com uma CLI
- Usar GetParameters com uma CLI
- Usar GetPatchBaseline com uma CLI
- Usar GetPatchBaselineForPatchGroup com uma CLI
- Usar ListAssociationVersions com uma CLI
- Usar ListAssociations com uma CLI
- Usar ListCommandInvocations com o AWS SDK ou a CLI
- Usar ListCommands com uma CLI
- Usar ListComplianceItems com uma CLI
- Usar ListComplianceSummaries com uma CLI
- Usar ListDocumentVersions com uma CLI
- Usar ListDocuments com uma CLI
- Usar ListInventoryEntries com uma CLI
- Usar ListResourceComplianceSummaries com uma CLI
- Usar ListTagsForResource com uma CLI
- Usar ModifyDocumentPermission com uma CLI
- Usar PutComplianceItems com uma CLI
- Usar PutInventory com uma CLI
- Usar PutParameter com o AWS SDK ou a CLI
- Usar RegisterDefaultPatchBaseline com uma CLI
- Usar RegisterPatchBaselineForPatchGroup com uma CLI
- Usar RegisterTargetWithMaintenanceWindow com uma CLI
- Usar RegisterTaskWithMaintenanceWindow com uma CLI
- Usar RemoveTagsFromResource com uma CLI

- Usar SendCommand com o AWS SDK ou a CLI
- Usar StartAutomationExecution com uma CLI
- Usar StartSession com uma CLI
- Usar StopAutomationExecution com uma CLI
- Usar UpdateAssociation com uma CLI
- Usar UpdateAssociationStatus com uma CLI
- Usar UpdateDocument com uma CLI
- Usar UpdateDocumentDefaultVersion com uma CLI
- Usar UpdateMaintenanceWindow com o AWS SDK ou a CLI
- Usar UpdateManagedInstanceRole com uma CLI
- Usar UpdateOpsItem com o AWS SDK ou a CLI
- Usar UpdatePatchBaseline com uma CLI

Hello Systems Manager

Os exemplos de código a seguir mostram como começar a usar o Systems Manager.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.ssm.SsmClient;
import software.amazon.awssdk.services.ssm.model.DocumentFilter;
import software.amazon.awssdk.services.ssm.model.ListDocumentsRequest;
import software.amazon.awssdk.services.ssm.model.ListDocumentsResponse;

public class HelloSSM {

    public static void main(String[] args) {
```

```
        final String usage = """

                Usage:
                    <awsAccount>

                Where:
                    awsAccount - Your AWS Account number.
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String awsAccount = args[0] ;
        Region region = Region.US_EAST_1;
        SsmClient ssmClient = SsmClient.builder()
            .region(region)
            .build();

        listDocuments(ssmClient, awsAccount);
    }

    /*
    This code automatically fetches the next set of results using the `nextToken`
and
    stops once the desired maxResults (20 in this case) have been reached.
    */
    public static void listDocuments(SsmClient ssmClient, String awsAccount) {
        String nextToken = null;
        int totalDocumentsReturned = 0;
        int maxResults = 20;
        do {
            ListDocumentsRequest request = ListDocumentsRequest.builder()
                .documentFilterList(
                    DocumentFilter.builder()
                        .key("Owner")
                        .value(awsAccount)
                        .build()
                    )
                .maxResults(maxResults)
                .nextToken(nextToken)
                .build();
```

```
            ListDocumentsResponse response = ssmClient.listDocuments(request);
            response.documentIdentifiers().forEach(identifier ->
 System.out.println("Document Name: " + identifier.name()));
            nextToken = response.nextToken();
            totalDocumentsReturned += response.documentIdentifiers().size();
        } while (nextToken != null && totalDocumentsReturned < maxResults);
    }
}
```

- Para obter detalhes da API, consulte ListDocuments na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
import { paginateListDocuments, SSMClient } from "@aws-sdk/client-ssm";

// Call ListDocuments and display the result.
export const main = async () => {
  const client = new SSMClient();
  const listDocumentsPaginated = [];
  console.log(
    "Hello, AWS Systems Manager! Let's list some of your documents:\n",
  );
  try {
    // The paginate function is a wrapper around the base command.
    const paginator = paginateListDocuments({ client }, { MaxResults: 5 });
    for await (const page of paginator) {
      listDocumentsPaginated.push(...page.DocumentIdentifiers);
    }
  } catch (caught) {
    console.error(`There was a problem saying hello: ${caught.message}`);
    throw caught;
```

```
  }

  for (const { Name, DocumentFormat, CreatedDate } of listDocumentsPaginated) {
    console.log(`${Name} - ${DocumentFormat} - ${CreatedDate}`);
  }
};

// Call function if run directly.
import { fileURLToPath } from "node:url";
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main();
}
```

- Para obter detalhes da API, consulte ListDocuments na Referência da API do AWS SDK para JavaScript.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
import boto3
from botocore.exceptions import ClientError


def hello_systems_manager(ssm_client):
    """
    Use the AWS SDK for Python (Boto3) to create an AWS Systems Manager
    client and list the first 5 documents in your account.
    This example uses the default settings specified in your shared credentials
    and config files.

    :param ssm_client: A Boto3 AWS Systems Manager Client object. This object
 wraps
                           the low-level AWS Systems Manager service API.
```

```
    """
    print("Hello, AWS Systems Manager! Let's list some of your documents:\n")

    paginator = ssm_client.get_paginator("list_documents")
    page_iterator = paginator.paginate(PaginationConfig={"MaxItems": 5})
    for page in page_iterator:
        for document in page["DocumentIdentifiers"]:
            print(f"  {document['Name']}")


if __name__ == "__main__":
    try:
        hello_systems_manager(boto3.client("ssm"))
    except ClientError as err:
        print("Hello systems manager had an error.")
        print(err.response["Error"]["Code"])
        print(err.response["Error"]["Message"])
```

- Para obter detalhes da API, consulte ListDocuments na Referência da API do AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Conheça os conceitos básicos do Systems Manager com um SDK da AWS

Os exemplos de código a seguir mostram como:

- Criar uma janela de manutenção.
- Modificar a programação da janela de manutenção.
- Criar um documento.
- Enviar um comando para uma instância do EC2 especificada.
- Criar um novo OpsItem.
- Atualizar e resolver o OpsItem.
- Excluir a janela de manutenção, o OpsItem e o documento.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
import software.amazon.awssdk.services.ssm.model.DocumentAlreadyExistsException;
import software.amazon.awssdk.services.ssm.model.SsmException;

import java.util.Scanner;
public class SSMScenario {
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");

    public static void main(String[] args) {
        String usage = """
            Usage:
              <instanceId> <title> <source> <category> <severity>

            Where:
                instanceId - The Amazon EC2 Linux/UNIX instance Id that AWS
 Systems Manager uses (ie, i-0149338494ed95f06).
                title - The title of the parameter (default is Disk Space Alert).
                source - The source of the parameter (default is EC2).
                category - The category of the parameter. Valid values are
 'Availability', 'Cost', 'Performance', 'Recovery', 'Security' (default is
 Performance).
                severity - The severity of the parameter. Severity should be a
 number from 1 to 4 (default is 2).
        """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        Scanner scanner = new Scanner(System.in);
        SSMActions actions = new SSMActions();
```

```
        String documentName;
        String windowName;
        String instanceId = args[0];
        String title = "Disk Space Alert" ;
        String source = "EC2" ;
        String category = "Availability" ;
        String severity = "2" ;

        System.out.println(DASHES);
        System.out.println("""
                Welcome to the AWS Systems Manager SDK Basics scenario.
                This Java program demonstrates how to interact with AWS Systems
Manager using the AWS SDK for Java (v2).
                AWS Systems Manager is the operations hub for your AWS
applications and resources and a secure end-to-end management solution.
                The program's primary functionalities include creating a
maintenance window, creating a document, sending a command to a document,
                listing documents, listing commands, creating an OpsItem,
modifying an OpsItem, and deleting AWS SSM resources.
                Upon completion of the program, all AWS resources are cleaned up.
                Let's get started...

                """);
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println("1. Create an SSM maintenance window.");
        System.out.println("Please enter the maintenance window name (default is
ssm-maintenance-window):");
        String win = scanner.nextLine();
        windowName = win.isEmpty() ? "ssm-maintenance-window" : win;
        String winId = null;
        try {
            winId = actions.createMaintenanceWindow(windowName);
            waitForInputToContinue(scanner);
            System.out.println("The maintenance window ID is: " + winId);
        } catch (DocumentAlreadyExistsException e) {
            System.err.println("The SSM maintenance window already exists.
Retrieving existing window ID...");
            String existingWinId = actions.createMaintenanceWindow(windowName);
            System.out.println("Existing window ID: " + existingWinId);
        } catch (SsmException e) {
            System.err.println("SSM error: " + e.getMessage());
            return;
```

```
        } catch (RuntimeException e) {
            System.err.println("Unexpected error: " + e.getMessage());
            return;
        }
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println("2. Modify the maintenance window by changing the
schedule");
        waitForInputToContinue(scanner);
        try {
            actions.updateSSMMaintenanceWindow(winId, windowName);
            waitForInputToContinue(scanner);
            System.out.println("The SSM maintenance window was successfully
updated");
        } catch (SsmException e) {
            System.err.println("SSM error: " + e.getMessage());
            return;
        } catch (RuntimeException e) {
            System.err.println("Unexpected error: " + e.getMessage());
            return;
        }
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println("3. Create an SSM document that defines the actions
that Systems Manager performs on your managed nodes.");
        System.out.println("Please enter the document name (default is
ssmdocument):");
        String doc = scanner.nextLine();
        documentName = doc.isEmpty() ? "ssmdocument" : doc;
        try {
            actions.createSSMDoc(documentName);
            waitForInputToContinue(scanner);
            System.out.println("The SSM document was successfully created");
        } catch (DocumentAlreadyExistsException e) {
            System.err.println("The SSM document already exists. Moving on");
        } catch (SsmException e) {
            System.err.println("SSM error: " + e.getMessage());
            return;
        } catch (RuntimeException e) {
            System.err.println("Unexpected error: " + e.getMessage());
        }
        waitForInputToContinue(scanner);
```

```
        System.out.println(DASHES);

        System.out.println("4. Now we are going to run a command on an EC2
instance");
        waitForInputToContinue(scanner);
        String commandId="";
        try {
            commandId = actions.sendSSMCommand(documentName, instanceId);
            waitForInputToContinue(scanner);
            System.out.println("The command was successfully sent. Command ID: "
+ commandId);
        } catch (SsmException e) {
            System.err.println("SSM error: " + e.getMessage());
        } catch (InterruptedException e) {
            System.err.println("Thread was interrupted: " + e.getMessage());
        } catch (RuntimeException e) {
            System.err.println("Unexpected error: " + e.getMessage());
        }
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println("5. Lets get the time when the specific command was
sent to the specific managed node");
        waitForInputToContinue(scanner);
        try {
            actions.displayCommands(commandId);
            System.out.println("The command invocations were successfully
displayed.");
        } catch (SsmException e) {
            System.err.println("SSM error: " + e.getMessage());
            return;
        } catch (RuntimeException e) {
            System.err.println("Unexpected error: " + e.getMessage());
            return;
        }
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("""
             6. Now we will create an SSM OpsItem.
             A SSM OpsItem is a feature provided by Amazon's Systems Manager
(SSM) service.
```

```
            It is a type of operational data item that allows you to manage and
track various operational issues,
            events, or tasks within your AWS environment.

            You can create OpsItems to track and manage operational issues as
they arise.
            For example, you could create an OpsItem whenever your application
detects a critical error
            or an anomaly in your infrastructure.
           """);

       waitForInputToContinue(scanner);
       String opsItemId;
       try {
           opsItemId = actions.createSSMOpsItem(title, source, category,
severity);
           System.out.println(opsItemId + " was created");
       } catch (SsmException e) {
           System.err.println("SSM error: " + e.getMessage());
           return;
       } catch (RuntimeException e) {
           System.err.println("Unexpected error: " + e.getMessage());
           return;
       }
       waitForInputToContinue(scanner);
       System.out.println(DASHES);

       System.out.println(DASHES);
       System.out.println("7. Now we will update the SSM OpsItem "+opsItemId);
       waitForInputToContinue(scanner);
       String description = "An update to "+opsItemId ;
       try {
           actions.updateOpsItem(opsItemId, title, description);
       } catch (SsmException e) {
           System.err.println("SSM error: " + e.getMessage());
           return;
       } catch (RuntimeException e) {
           System.err.println("Unexpected error: " + e.getMessage());
           return;
       }

       System.out.println(DASHES);
       System.out.println("8. Now we will get the status of the SSM OpsItem
"+opsItemId);
```

```
        waitForInputToContinue(scanner);
        try {
            actions.describeOpsItems(opsItemId);
        } catch (SsmException e) {
            System.err.println("SSM error: " + e.getMessage());
            return;
        } catch (RuntimeException e) {
            System.err.println("Unexpected error: " + e.getMessage());
            return;
        }

        System.out.println(DASHES);
        System.out.println("9. Now we will resolve the SSM OpsItem "+opsItemId);
        waitForInputToContinue(scanner);
        try {
            actions.resolveOpsItem(opsItemId);
        } catch (SsmException e) {
            System.err.println("SSM error: " + e.getMessage());
            return;
        } catch (RuntimeException e) {
            System.err.println("Unexpected error: " + e.getMessage());
            return;
        }

        System.out.println(DASHES);
        System.out.println("10. Would you like to delete the AWS Systems Manager
resources? (y/n)");
        String delAns = scanner.nextLine().trim();
        if (delAns.equalsIgnoreCase("y")) {
            System.out.println("You selected to delete the resources.");
            waitForInputToContinue(scanner);
            try {
                actions.deleteMaintenanceWindow(winId);
                actions.deleteDoc(documentName);
            } catch (SsmException e) {
                System.err.println("SSM error: " + e.getMessage());
                return;
            } catch (RuntimeException e) {
                System.err.println("Unexpected error: " + e.getMessage());
                return;
            }
        } else {
            System.out.println("The AWS Systems Manager resources will not be
deleted");
```

```
        }
        System.out.println(DASHES);

        System.out.println("This concludes the AWS Systems Manager SDK Basics
 scenario.");
        System.out.println(DASHES);
    }

    private static void waitForInputToContinue(Scanner scanner) {
        while (true) {
            System.out.println("");
            System.out.println("Enter 'c' followed by <ENTER> to continue:");
            String input = scanner.nextLine();

            if (input.trim().equalsIgnoreCase("c")) {
                System.out.println("Continuing with the program...");
                System.out.println("");
                break;
            } else {
                // Handle invalid input.
                System.out.println("Invalid input. Please try again.");
            }
        }
    }
}
```

Uma classe de wrapper para métodos do SDK do Systems Manager.

```
public class SSMActions {

    private static SsmAsyncClient ssmAsyncClient;

    private static SsmAsyncClient getAsyncClient() {
        if (ssmAsyncClient == null) {
            SdkAsyncHttpClient httpClient = NettyNioAsyncHttpClient.builder()
                    .maxConcurrency(100)
                    .connectionTimeout(Duration.ofSeconds(60))
                    .readTimeout(Duration.ofSeconds(60))
                    .writeTimeout(Duration.ofSeconds(60))
                    .build();
```

```
            ClientOverrideConfiguration overrideConfig =
ClientOverrideConfiguration.builder()
                    .apiCallTimeout(Duration.ofMinutes(2))
                    .apiCallAttemptTimeout(Duration.ofSeconds(90))
                    .retryPolicy(RetryPolicy.builder()
                            .numRetries(3)
                            .build())
                    .build();

            ssmAsyncClient = SsmAsyncClient.builder()
                    .region(Region.US_EAST_1)
                    .httpClient(httpClient)
                    .overrideConfiguration(overrideConfig)
                    .build();
        }
        return ssmAsyncClient;
    }

    /**
     * Deletes an AWS SSM document asynchronously.
     *
     * @param documentName The name of the document to delete.
     * <p>
     * This method initiates an asynchronous request to delete an SSM document.
     * If an exception occurs, it handles the error appropriately.
     */
    public void deleteDoc(String documentName) {
        DeleteDocumentRequest documentRequest = DeleteDocumentRequest.builder()
                .name(documentName)
                .build();

        CompletableFuture<Void> future = CompletableFuture.runAsync(() -> {
            getAsyncClient().deleteDocument(documentRequest)
                    .thenAccept(response -> {
                        System.out.println("The SSM document was successfully
deleted.");
                    })
                    .exceptionally(ex -> {
                        throw new CompletionException(ex);
                    }).join();
        }).exceptionally(ex -> {
            Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
            if (cause instanceof SsmException) {
```

```
                throw new RuntimeException("SSM error: " + cause.getMessage(),
cause);
            } else {
                throw new RuntimeException("Unexpected error: " +
cause.getMessage(), cause);
            }
        });

        try {
            future.join();
        } catch (CompletionException ex) {
            throw ex.getCause() instanceof RuntimeException ? (RuntimeException)
ex.getCause() : ex;
        }
    }

    /**
     * Deletes an AWS SSM Maintenance Window asynchronously.
     *
     * @param winId The ID of the Maintenance Window to delete.
     * <p>
     * This method initiates an asynchronous request to delete an SSM Maintenance
Window.
     * If an exception occurs, it handles the error appropriately.
     */
    public void deleteMaintenanceWindow(String winId) {
        DeleteMaintenanceWindowRequest windowRequest =
DeleteMaintenanceWindowRequest.builder()
                .windowId(winId)
                .build();

        CompletableFuture<Void> future = CompletableFuture.runAsync(() -> {
            getAsyncClient().deleteMaintenanceWindow(windowRequest)
                    .thenAccept(response -> {
                        System.out.println("The maintenance window was
successfully deleted.");
                    })
                    .exceptionally(ex -> {
                        throw new CompletionException(ex);
                    }).join();
        }).exceptionally(ex -> {
            Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
            if (cause instanceof SsmException) {
```

```
                throw new RuntimeException("SSM error: " + cause.getMessage(),
cause);
            } else {
                throw new RuntimeException("Unexpected error: " +
cause.getMessage(), cause);
            }
        });

        try {
            future.join();
        } catch (CompletionException ex) {
            throw ex.getCause() instanceof RuntimeException ? (RuntimeException)
ex.getCause() : ex;
        }
    }

    /**
     * Resolves an AWS SSM OpsItem asynchronously.
     *
     * @param opsID The ID of the OpsItem to resolve.
     * <p>
     * This method initiates an asynchronous request to resolve an SSM OpsItem.
     * If an exception occurs, it handles the error appropriately.
     */
    public void resolveOpsItem(String opsID) {
        UpdateOpsItemRequest opsItemRequest = UpdateOpsItemRequest.builder()
                .opsItemId(opsID)
                .status(OpsItemStatus.RESOLVED)
                .build();

        CompletableFuture<Void> future = CompletableFuture.runAsync(() -> {
            getAsyncClient().updateOpsItem(opsItemRequest)
                    .thenAccept(response -> {
                        System.out.println("OpsItem resolved successfully.");
                    })
                    .exceptionally(ex -> {
                        throw new CompletionException(ex);
                    }).join();
        }).exceptionally(ex -> {
            Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
            if (cause instanceof SsmException) {
                throw new RuntimeException("SSM error: " + cause.getMessage(),
cause);
```

```
            } else {
                throw new RuntimeException("Unexpected error: " +
cause.getMessage(), cause);
            }
        });

        try {
            future.join();
        } catch (CompletionException ex) {
            throw ex.getCause() instanceof RuntimeException ? (RuntimeException)
ex.getCause() : ex;
        }
    }

    /**
     * Describes AWS SSM OpsItems asynchronously.
     *
     * @param key The key to filter OpsItems by (e.g., OPS_ITEM_ID).
     *
     * This method initiates an asynchronous request to describe SSM OpsItems.
     * If the request is successful, it prints the title and status of each
OpsItem.
     * If an exception occurs, it handles the error appropriately.
     */
    public void describeOpsItems(String key) {
        OpsItemFilter filter = OpsItemFilter.builder()
                .key(OpsItemFilterKey.OPS_ITEM_ID)
                .values(key)
                .operator(OpsItemFilterOperator.EQUAL)
                .build();

        DescribeOpsItemsRequest itemsRequest = DescribeOpsItemsRequest.builder()
                .maxResults(10)
                .opsItemFilters(filter)
                .build();

        CompletableFuture<Void> future = CompletableFuture.runAsync(() -> {
            getAsyncClient().describeOpsItems(itemsRequest)
                    .thenAccept(itemsResponse -> {
                        List<OpsItemSummary> items =
itemsResponse.opsItemSummaries();
                        for (OpsItemSummary item : items) {
                            System.out.println("The item title is " +
item.title() + " and the status is " + item.status().toString());
```

```
                    }
                })
                .exceptionally(ex -> {
                    throw new CompletionException(ex);
                }).join();
        }).exceptionally(ex -> {
            Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
            if (cause instanceof SsmException) {
                throw new RuntimeException("SSM error: " + cause.getMessage(),
cause);
            } else {
                throw new RuntimeException("Unexpected error: " +
cause.getMessage(), cause);
            }
        });

        try {
            future.join();
        } catch (CompletionException ex) {
            throw ex.getCause() instanceof RuntimeException ? (RuntimeException)
ex.getCause() : ex;
        }
    }

    /**
     * Updates the AWS SSM OpsItem asynchronously.
     *
     * @param opsItemId The ID of the OpsItem to update.
     * @param title The new title of the OpsItem.
     * @param description The new description of the OpsItem.
     * <p>
     * This method initiates an asynchronous request to update an SSM OpsItem.
     * If the request is successful, it completes without returning a value.
     * If an exception occurs, it handles the error appropriately.
     */
    public void updateOpsItem(String opsItemId, String title, String description)
{
        Map<String, OpsItemDataValue> operationalData = new HashMap<>();
        operationalData.put("key1",
OpsItemDataValue.builder().value("value1").build());
        operationalData.put("key2",
OpsItemDataValue.builder().value("value2").build());
```

```
        CompletableFuture<Void> future =
getOpsItem(opsItemId).thenCompose(opsItem -> {
            UpdateOpsItemRequest request = UpdateOpsItemRequest.builder()
                    .opsItemId(opsItemId)
                    .title(title)
                    .operationalData(operationalData)
                    .status(opsItem.statusAsString())
                    .description(description)
                    .build();

            return getAsyncClient().updateOpsItem(request).thenAccept(response ->
{
                System.out.println(opsItemId + " updated successfully.");
            }).exceptionally(ex -> {
                throw new CompletionException(ex);
            });
        }).exceptionally(ex -> {
            Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
            if (cause instanceof SsmException) {
                throw new RuntimeException("SSM error: " + cause.getMessage(),
cause);
            } else {
                throw new RuntimeException("Unexpected error: " +
cause.getMessage(), cause);
            }
        });

        try {
            future.join();
        } catch (CompletionException ex) {
            throw ex.getCause() instanceof RuntimeException ? (RuntimeException)
ex.getCause() : ex;
        }
    }


    private static CompletableFuture<OpsItem> getOpsItem(String opsItemId) {
        GetOpsItemRequest request =
GetOpsItemRequest.builder().opsItemId(opsItemId).build();
        return
getAsyncClient().getOpsItem(request).thenApply(GetOpsItemResponse::opsItem);
    }
```

```
    /**
     * Creates an SSM OpsItem asynchronously.
     *
     * @param title The title of the OpsItem.
     * @param source The source of the OpsItem.
     * @param category The category of the OpsItem.
     * @param severity The severity of the OpsItem.
     * @return The ID of the created OpsItem.
     * <p>
     * This method initiates an asynchronous request to create an SSM OpsItem.
     * If the request is successful, it returns the OpsItem ID.
     * If an exception occurs, it handles the error appropriately.
     */
    public String createSSMOpsItem(String title, String source, String category,
String severity) {
        CreateOpsItemRequest opsItemRequest = CreateOpsItemRequest.builder()
                .description("Created by the SSM Java API")
                .title(title)
                .source(source)
                .category(category)
                .severity(severity)
                .build();

        CompletableFuture<CreateOpsItemResponse> future =
getAsyncClient().createOpsItem(opsItemRequest);

        try {
            CreateOpsItemResponse response = future.join();
            return response.opsItemId();
        } catch (CompletionException e) {
            Throwable cause = e.getCause();
            if (cause instanceof SsmException) {
                throw (SsmException) cause;
            } else {
                throw new RuntimeException(cause);
            }
        }
    }

    /**
     * Displays the date and time when the specific command was invoked.
     *
     * @param commandId The ID of the command to describe.
     * <p>
```

```
    * This method initiates an asynchronous request to list command invocations
and prints the date and time of each command invocation.
    * If an exception occurs, it handles the error appropriately.
    */
   public void displayCommands(String commandId) {
       ListCommandInvocationsRequest commandInvocationsRequest =
ListCommandInvocationsRequest.builder()
               .commandId(commandId)
               .build();

       CompletableFuture<ListCommandInvocationsResponse> future =
getAsyncClient().listCommandInvocations(commandInvocationsRequest);
       future.thenAccept(response -> {
           List<CommandInvocation> commandList = response.commandInvocations();
           DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd
HH:mm:ss").withZone(ZoneId.systemDefault());
           for (CommandInvocation invocation : commandList) {
               System.out.println("The time of the command invocation is " +
formatter.format(invocation.requestedDateTime()));
           }
       }).exceptionally(ex -> {
           Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
           if (cause instanceof SsmException) {
               throw (SsmException) cause;
           } else {
               throw new RuntimeException(cause);
           }
       }).join();
   }

   /**
    * Sends a SSM command to a managed node asynchronously.
    *
    * @param documentName The name of the document to use.
    * @param instanceId The ID of the instance to send the command to.
    * @return The command ID.
    * <p>
    * This method initiates asynchronous requests to send a SSM command to a
managed node.
    * It waits until the document is active, sends the command, and checks the
command execution status.
    */
```

```
   public String sendSSMCommand(String documentName, String instanceId) throws
InterruptedException, SsmException {
       // Before we use Document to send a command - make sure it is active.
       CompletableFuture<Void> documentActiveFuture =
CompletableFuture.runAsync(() -> {
           boolean isDocumentActive = false;
           DescribeDocumentRequest request = DescribeDocumentRequest.builder()
                   .name(documentName)
                   .build();

           while (!isDocumentActive) {
               CompletableFuture<DescribeDocumentResponse> response =
getAsyncClient().describeDocument(request);
               String documentStatus =
response.join().document().statusAsString();
               if (documentStatus.equals("Active")) {
                   System.out.println("The SSM document is active and ready to
use.");
                   isDocumentActive = true;
               } else {
                   System.out.println("The SSM document is not active. Status: "
+ documentStatus);
                   try {
                       Thread.sleep(5000);
                   } catch (InterruptedException e) {
                       throw new RuntimeException(e);
                   }
               }
           }
       });

       documentActiveFuture.join();

       // Create the SendCommandRequest.
       SendCommandRequest commandRequest = SendCommandRequest.builder()
               .documentName(documentName)
               .instanceIds(instanceId)
               .build();

       // Send the command.
       CompletableFuture<SendCommandResponse> commandFuture =
getAsyncClient().sendCommand(commandRequest);
       final String[] commandId = {null};
```

```
        commandFuture.whenComplete((commandResponse, ex) -> {
            if (commandResponse != null) {
                commandId[0] = commandResponse.command().commandId();
                System.out.println("Command ID: " + commandId[0]);

                // Wait for the command execution to complete.
                GetCommandInvocationRequest invocationRequest =
GetCommandInvocationRequest.builder()
                        .commandId(commandId[0])
                        .instanceId(instanceId)
                        .build();

                try {
                    System.out.println("Wait 5 secs");
                    TimeUnit.SECONDS.sleep(5);

                    // Retrieve the command execution details.
                    CompletableFuture<GetCommandInvocationResponse>
invocationFuture = getAsyncClient().getCommandInvocation(invocationRequest);
                    invocationFuture.whenComplete((commandInvocationResponse,
invocationEx) -> {
                        if (commandInvocationResponse != null) {
                            // Check the status of the command execution.
                            CommandInvocationStatus status =
commandInvocationResponse.status();
                            if (status == CommandInvocationStatus.SUCCESS) {
                                System.out.println("Command execution
successful");
                            } else {
                                System.out.println("Command execution failed.
Status: " + status);
                            }
                        } else {
                            Throwable invocationCause = (invocationEx instanceof
CompletionException) ? invocationEx.getCause() : invocationEx;
                            throw new CompletionException(invocationCause);
                        }
                    }).join();
                } catch (InterruptedException e) {
                    throw new RuntimeException(e);
                }
            } else {
                Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
```

```
                if (cause instanceof SsmException) {
                    throw (SsmException) cause;
                } else {
                    throw new RuntimeException(cause);
                }
            }
        }).join();

        return commandId[0];
    }

    /**
     * Creates an AWS SSM document asynchronously.
     *
     * @param docName The name of the document to create.
     * <p>
     * This method initiates an asynchronous request to create an SSM document.
     * If the request is successful, it prints the document status.
     * If an exception occurs, it handles the error appropriately.
     */
    public void createSSMDoc(String docName) throws SsmException {
        String jsonData = """
        {
        "schemaVersion": "2.2",
        "description": "Run a simple shell command",
        "mainSteps": [
            {
                "action": "aws:runShellScript",
                "name": "runEchoCommand",
                "inputs": {
                  "runCommand": [
                    "echo 'Hello, world!'"
                  ]
                }
              }
            ]
        }
        """;

        CreateDocumentRequest request = CreateDocumentRequest.builder()
                .content(jsonData)
                .name(docName)
                .documentType(DocumentType.COMMAND)
                .build();
```

```
        CompletableFuture<CreateDocumentResponse> future =
getAsyncClient().createDocument(request);
        future.thenAccept(response -> {
            System.out.println("The status of the SSM document is " +
response.documentDescription().status());
        }).exceptionally(ex -> {
            Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
            if (cause instanceof DocumentAlreadyExistsException) {
                throw new CompletionException(cause);
            } else if (cause instanceof SsmException) {
                throw new CompletionException(cause);
            } else {
                throw new RuntimeException(cause);
            }
        }).join();
    }

    /**
     * Updates an SSM maintenance window asynchronously.
     *
     * @param id The ID of the maintenance window to update.
     * @param name The new name for the maintenance window.
     * <p>
     * This method initiates an asynchronous request to update an SSM maintenance
window.
     * If the request is successful, it prints a success message.
     * If an exception occurs, it handles the error appropriately.
     */
    public void updateSSMMaintenanceWindow(String id, String name) throws
SsmException {
        UpdateMaintenanceWindowRequest updateRequest =
UpdateMaintenanceWindowRequest.builder()
                .windowId(id)
                .allowUnassociatedTargets(true)
                .duration(24)
                .enabled(true)
                .name(name)
                .schedule("cron(0 0 ? * MON *)")
                .build();

        CompletableFuture<UpdateMaintenanceWindowResponse> future =
getAsyncClient().updateMaintenanceWindow(updateRequest);
```

```
        future.whenComplete((response, ex) -> {
            if (response != null) {
                System.out.println("The SSM maintenance window was successfully
updated");
            } else {
                Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
                if (cause instanceof SsmException) {
                    throw new CompletionException(cause);
                } else {
                    throw new RuntimeException(cause);
                }
            }
        }).join();
    }

    /**
     * Creates an SSM maintenance window asynchronously.
     *
     * @param winName The name of the maintenance window.
     * @return The ID of the created or existing maintenance window.
     * <p>
     * This method initiates an asynchronous request to create an SSM maintenance
window.
     * If the request is successful, it prints the maintenance window ID.
     * If an exception occurs, it handles the error appropriately.
     */
    public String createMaintenanceWindow(String winName) throws SsmException,
DocumentAlreadyExistsException {
        CreateMaintenanceWindowRequest request =
CreateMaintenanceWindowRequest.builder()
                .name(winName)
                .description("This is my maintenance window")
                .allowUnassociatedTargets(true)
                .duration(2)
                .cutoff(1)
                .schedule("cron(0 10 ? * MON-FRI *)")
                .build();

        CompletableFuture<CreateMaintenanceWindowResponse> future =
getAsyncClient().createMaintenanceWindow(request);
        final String[] windowId = {null};
        future.whenComplete((response, ex) -> {
            if (response != null) {
```

```
                String maintenanceWindowId = response.windowId();
                System.out.println("The maintenance window id is " +
maintenanceWindowId);
                windowId[0] = maintenanceWindowId;
            } else {
                Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
                if (cause instanceof DocumentAlreadyExistsException) {
                    throw new CompletionException(cause);
                } else if (cause instanceof SsmException) {
                    throw new CompletionException(cause);
                } else {
                    throw new RuntimeException(cause);
                }
            }
        }).join();

        if (windowId[0] == null) {
            MaintenanceWindowFilter filter = MaintenanceWindowFilter.builder()
                    .key("name")
                    .values(winName)
                    .build();

            DescribeMaintenanceWindowsRequest winRequest =
DescribeMaintenanceWindowsRequest.builder()
                    .filters(filter)
                    .build();

            CompletableFuture<DescribeMaintenanceWindowsResponse> describeFuture
= getAsyncClient().describeMaintenanceWindows(winRequest);
            describeFuture.whenComplete((describeResponse, describeEx) -> {
                if (describeResponse != null) {
                    List<MaintenanceWindowIdentity> windows =
describeResponse.windowIdentities();
                    if (!windows.isEmpty()) {
                        windowId[0] = windows.get(0).windowId();
                        System.out.println("Window ID: " + windowId[0]);
                    } else {
                        System.out.println("Window not found.");
                        windowId[0] = "";
                    }
                } else {
                    Throwable describeCause = (describeEx instanceof
CompletionException) ? describeEx.getCause() : describeEx;
```

```
                    throw new RuntimeException("Error describing maintenance
 windows: " + describeCause.getMessage(), describeCause);
                }
            }).join();
        }

        return windowId[0];
    }
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK for Java 2.x.
  - CreateDocument
  - CreateMaintenanceWindow
  - CreateOpsItem
  - DeleteMaintenanceWindow
  - ListCommandInvocations
  - SendCommand
  - UpdateOpsItem

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
import {
  Scenario,
  ScenarioAction,
  ScenarioInput,
  ScenarioOutput,
} from "@aws-doc-sdk-examples/lib/scenario/index.js";
import { fileURLToPath } from "node:url";
```

```
import {
  CreateDocumentCommand,
  CreateMaintenanceWindowCommand,
  CreateOpsItemCommand,
  DeleteDocumentCommand,
  DeleteMaintenanceWindowCommand,
  DeleteOpsItemCommand,
  DescribeOpsItemsCommand,
  DocumentAlreadyExists,
  OpsItemStatus,
  waitUntilCommandExecuted,
  CancelCommandCommand,
  paginateListCommandInvocations,
  SendCommandCommand,
  UpdateMaintenanceWindowCommand,
  UpdateOpsItemCommand,
  SSMClient,
} from "@aws-sdk/client-ssm";
import { parseArgs } from "node:util";

/**
 * @typedef {{
 *   ssmClient: import('@aws-sdk/client-ssm').SSMClient,
 *   documentName?: string
 *   maintenanceWindow?: string
 *   winId?: int
 *   ec2InstanceId?: string
 *   requestedDateTime?: Date
 *   opsItemId?: string
 *   askToDeleteResources?: boolean
 * }} State
 */

const defaultMaintenanceWindow = "ssm-maintenance-window";
const defaultDocumentName = "ssmdocument";
// The timeout duration is highly dependent on the specific setup and environment
 necessary. This example handles only the most common error cases, and uses a
 much shorter duration than most productions systems would use.
const COMMAND_TIMEOUT_DURATION_SECONDS = 30; // 30 seconds

const pressEnter = new ScenarioInput("continue", "Press Enter to continue", {
  type: "confirm",
});
```

```
const greet = new ScenarioOutput(
  "greet",
  `Welcome to the AWS Systems Manager SDK Getting Started scenario.
    This program demonstrates how to interact with Systems Manager using the AWS
 SDK for JavaScript V3.
    Systems Manager is the operations hub for your AWS applications and resources
 and a secure end-to-end management solution.
    The program's primary functions include creating a maintenance window,
 creating a document, sending a command to a document,
    listing documents, listing commands, creating an OpsItem, modifying an
 OpsItem, and deleting Systems Manager resources.
    Upon completion of the program, all AWS resources are cleaned up.
    Let's get started...`,
  { header: true },
);

const createMaintenanceWindow = new ScenarioOutput(
  "createMaintenanceWindow",
  "Step 1: Create a Systems Manager maintenance window.",
);

const getMaintenanceWindow = new ScenarioInput(
  "maintenanceWindow",
  "Please enter the maintenance window name:",
  { type: "input", default: defaultMaintenanceWindow },
);

export const sdkCreateMaintenanceWindow = new ScenarioAction(
  "sdkCreateMaintenanceWindow",
  async (/** @type {State} */ state) => {
    try {
      const response = await state.ssmClient.send(
        new CreateMaintenanceWindowCommand({
          Name: state.maintenanceWindow,
          Schedule: "cron(0 10 ? * MON-FRI *)", //The schedule of the maintenance
 window in the form of a cron or rate expression.
          Duration: 2, //The duration of the maintenance window in hours.
          Cutoff: 1, //The number of hours before the end of the maintenance
 window that Amazon Web Services Systems Manager stops scheduling new tasks for
 execution.
          AllowUnassociatedTargets: true, //Allow the maintenance window to run
 on managed nodes, even if you haven't registered those nodes as targets.
        }),
      );
```

```
      state.winId = response.WindowId;
    } catch (caught) {
      console.error(caught.message);
      console.log(
        `An error occurred while creating the maintenance window. Please fix the
 error and try again. Error message: ${caught.message}`,
      );
      throw caught;
    }
  },
);

const modifyMaintenanceWindow = new ScenarioOutput(
  "modifyMaintenanceWindow",
  "Modify the maintenance window by changing the schedule.",
);

const sdkModifyMaintenanceWindow = new ScenarioAction(
  "sdkModifyMaintenanceWindow",
  async (/** @type {State} */ state) => {
    try {
      await state.ssmClient.send(
        new UpdateMaintenanceWindowCommand({
          WindowId: state.winId,
          Schedule: "cron(0 0 ? * MON *)",
        }),
      );
    } catch (caught) {
      console.error(caught.message);
      console.log(
        `An error occurred while modifying the maintenance window. Please fix the
 error and try again. Error message: ${caught.message}`,
      );
      throw caught;
    }
  },
);

const createSystemsManagerActions = new ScenarioOutput(
  "createSystemsManagerActions",
  "Create a document that defines the actions that Systems Manager performs on
 your EC2 instance.",
);
```

```
const getDocumentName = new ScenarioInput(
  "documentName",
  "Please enter the document: ",
  { type: "input", default: defaultDocumentName },
);

const sdkCreateSSMDoc = new ScenarioAction(
  "sdkCreateSSMDoc",
  async (/** @type {State} */ state) => {
    const contentData = `{
                "schemaVersion": "2.2",
                "description": "Run a simple shell command",
                "mainSteps": [
                    {
                        "action": "aws:runShellScript",
                        "name": "runEchoCommand",
                        "inputs": {
                          "runCommand": [
                            "echo 'Hello, world!'"
                          ]
                        }
                    }
                ]
            }`;
    try {
      await state.ssmClient.send(
        new CreateDocumentCommand({
          Content: contentData,
          Name: state.documentName,
          DocumentType: "Command",
        }),
      );
    } catch (caught) {
      console.log(`Exception type: (${typeof caught})`);
      if (caught instanceof DocumentAlreadyExists) {
        console.log("Document already exists. Continuing...\n");
      } else {
        console.error(caught.message);
        console.log(
          `An error occurred while creating the document. Please fix the error
 and try again. Error message: ${caught.message}`,
        );
        throw caught;
      }
```

```
    }
  },
);

const ec2HelloWorld = new ScenarioOutput(
  "ec2HelloWorld",
  `Now you have the option of running a command on an EC2 instance that echoes
 'Hello, world!'. In order to run this command, you must provide the instance ID
 of a Linux EC2 instance. If you do not already have a running Linux EC2 instance
 in your account, you can create one using the AWS console. For information
 about creating an EC2 instance, see https://docs.aws.amazon.com/AWSEC2/latest/
UserGuide/ec2-launch-instance-wizard.html.`,
);

const enterIdOrSkipEC2HelloWorld = new ScenarioInput(
  "enterIdOrSkipEC2HelloWorld",
  "Enter your EC2 InstanceId or press enter to skip this step: ",
  { type: "input", default: "" },
);

const sdkEC2HelloWorld = new ScenarioAction(
  "sdkEC2HelloWorld",
  async (/** @type {State} */ state) => {
    try {
      const response = await state.ssmClient.send(
        new SendCommandCommand({
          DocumentName: state.documentName,
          InstanceIds: [state.ec2InstanceId],
          TimeoutSeconds: COMMAND_TIMEOUT_DURATION_SECONDS,
        }),
      );
      state.CommandId = response.Command.CommandId;
    } catch (caught) {
      console.error(caught.message);
      console.log(
        `An error occurred while sending the command. Please fix the error and
 try again. Error message: ${caught.message}`,
      );
      throw caught;
    }
  },
  {
    skipWhen: (/** @type {State} */ state) =>
      state.enterIdOrSkipEC2HelloWorld === "",
```

```
  },
);

const sdkGetCommandTime = new ScenarioAction(
  "sdkGetCommandTime",
  async (/** @type {State} */ state) => {
    const listInvocationsPaginated = [];
    console.log(
      "Let's get the time when the specific command was sent to the specific
 managed node.",
    );

    console.log(
      `First, we'll wait for the command to finish executing. This may take up to
 ${COMMAND_TIMEOUT_DURATION_SECONDS} seconds.`,
    );
    const commandExecutedResult = waitUntilCommandExecuted(
      { client: state.ssmClient },
      {
        CommandId: state.CommandId,
        InstanceId: state.ec2InstanceId,
      },
    );
    // This is necessary because the TimeoutSeconds of SendCommandCommand is only
 for the delivery, not execution.
    try {
      await new Promise((_, reject) =>
        setTimeout(
          reject,
          COMMAND_TIMEOUT_DURATION_SECONDS * 1000,
          new Error("Command Timed Out"),
        ),
      );
    } catch (caught) {
      if (caught.message === "Command Timed Out") {
        commandExecutedResult.state = "TIMED_OUT";
      } else {
        throw caught;
      }
    }

    if (commandExecutedResult.state !== "SUCCESS") {
      console.log(
```

```
        `The command with id: ${state.CommandId} did not execute in the allotted
 time. Canceling command.`,
      );
      state.ssmClient.send(
        new CancelCommandCommand({
          CommandId: state.CommandId,
        }),
      );
      state.enterIdOrSkipEC2HelloWorld === "";
      return;
    }

    for await (const page of paginateListCommandInvocations(
      { client: state.ssmClient },
      { CommandId: state.CommandId },
    )) {
      listInvocationsPaginated.push(...page.CommandInvocations);
    }
    /**
     * @type {import('@aws-sdk/client-ssm').CommandInvocation}
     */
    const commandInvocation = listInvocationsPaginated.shift(); // Because the
 call was made with CommandId, there's only one result, so shift it off.
    state.requestedDateTime = commandInvocation.RequestedDateTime;

    console.log(
      `The command invocation happened at: ${state.requestedDateTime}.`,
    );
  },
  {
    skipWhen: (/** @type {State} */ state) =>
      state.enterIdOrSkipEC2HelloWorld === "",
  },
);

const createSSMOpsItem = new ScenarioOutput(
  "createSSMOpsItem",
  `Now we will create a Systems Manager OpsItem. An OpsItem is a feature provided
 by the Systems Manager service. It is a type of operational data item that
 allows you to manage and track various operational issues, events, or tasks
 within your AWS environment.
You can create OpsItems to track and manage operational issues as they arise.
 For example, you could create an OpsItem whenever your application detects a
 critical error or an anomaly in your infrastructure.`,
```

```
);

const sdkCreateSSMOpsItem = new ScenarioAction(
  "sdkCreateSSMOpsItem",
  async (/** @type {State} */ state) => {
    try {
      const response = await state.ssmClient.send(
        new CreateOpsItemCommand({
          Description: "Created by the System Manager Javascript API",
          Title: "Disk Space Alert",
          Source: "EC2",
          Category: "Performance",
          Severity: "2",
        }),
      );
      state.opsItemId = response.OpsItemId;
    } catch (caught) {
      console.error(caught.message);
      console.log(
        `An error occurred while creating the ops item. Please fix the error and
 try again. Error message: ${caught.message}`,
      );
      throw caught;
    }
  },
);

const updateOpsItem = new ScenarioOutput(
  "updateOpsItem",
  (/** @type {State} */ state) =>
    `Now we will update the OpsItem: ${state.opsItemId}`,
);

const sdkUpdateOpsItem = new ScenarioAction(
  "sdkUpdateOpsItem",
  async (/** @type {State} */ state) => {
    try {
      const _response = await state.ssmClient.send(
        new UpdateOpsItemCommand({
          OpsItemId: state.opsItemId,
          Description: `An update to ${state.opsItemId}`,
        }),
      );
    } catch (caught) {
```

```
      console.error(caught.message);
      console.log(
        `An error occurred while updating the ops item. Please fix the error and
 try again. Error message: ${caught.message}`,
      );
      throw caught;
    }
  },
);

const getOpsItemStatus = new ScenarioOutput(
  "getOpsItemStatus",
  (/** @type {State} */ state) =>
    `Now we will get the status of the OpsItem: ${state.opsItemId}`,
);

const sdkOpsItemStatus = new ScenarioAction(
  "sdkGetOpsItemStatus",
  async (/** @type {State} */ state) => {
    try {
      const response = await state.ssmClient.send(
        new DescribeOpsItemsCommand({
          OpsItemId: state.opsItemId,
        }),
      );
      state.opsItemStatus = response.OpsItemStatus;
    } catch (caught) {
      console.error(caught.message);
      console.log(
        `An error occurred while describing the ops item. Please fix the error
 and try again. Error message: ${caught.message}`,
      );
      throw caught;
    }
  },
);

const resolveOpsItem = new ScenarioOutput(
  "resolveOpsItem",
  (/** @type {State} */ state) =>
    `Now we will resolve the OpsItem: ${state.opsItemId}`,
);

const sdkResolveOpsItem = new ScenarioAction(
```

```
  "sdkResolveOpsItem",
  async (/** @type {State} */ state) => {
    try {
      const _response = await state.ssmClient.send(
        new UpdateOpsItemCommand({
          OpsItemId: state.opsItemId,
          Status: OpsItemStatus.RESOLVED,
        }),
      );
    } catch (caught) {
      console.error(caught.message);
      console.log(
        `An error occurred while updating the ops item. Please fix the error and
 try again. Error message: ${caught.message}`,
      );
      throw caught;
    }
  },
);

const askToDeleteResources = new ScenarioInput(
  "askToDeleteResources",
  "Would you like to delete the Systems Manager resources created during this
 example run?",
  { type: "confirm" },
);

const confirmDeleteChoice = new ScenarioOutput(
  "confirmDeleteChoice",
  (/** @type {State} */ state) => {
    if (state.askToDeleteResources) {
      return "You chose to delete the resources.";
    }
    return "The Systems Manager resources will not be deleted. Please delete them
 manually to avoid charges.";
  },
);

export const sdkDeleteResources = new ScenarioAction(
  "sdkDeleteResources",
  async (/** @type {State} */ state) => {
    try {
      await state.ssmClient.send(
        new DeleteOpsItemCommand({
```

```
          OpsItemId: state.opsItemId,
        }),
      );
      console.log(`The ops item: ${state.opsItemId} was successfully deleted.`);
    } catch (caught) {
      console.log(
        `There was a problem deleting the ops item: ${state.opsItemId}. Please
delete it manually. Error: ${caught.message}`,
      );
    }

    try {
      await state.ssmClient.send(
        new DeleteMaintenanceWindowCommand({
          Name: state.maintenanceWindow,
          WindowId: state.winId,
        }),
      );
      console.log(
        `The maintenance window: ${state.maintenanceWindow} was successfully
deleted.`,
      );
    } catch (caught) {
      console.log(
        `There was a problem deleting the maintenance window: ${state.opsItemId}.
Please delete it manually. Error: ${caught.message}`,
      );
    }

    try {
      await state.ssmClient.send(
        new DeleteDocumentCommand({
          Name: state.documentName,
        }),
      );
      console.log(
        `The document: ${state.documentName} was successfully deleted.`,
      );
    } catch (caught) {
      console.log(
        `There was a problem deleting the document: ${state.documentName}. Please
delete it manually. Error: ${caught.message}`,
      );
    }
```

```
  },
  { skipWhen: (/** @type {{}} */ state) => !state.askToDeleteResources },
);

const goodbye = new ScenarioOutput(
  "goodbye",
  "This concludes the Systems Manager Basics scenario for the AWS Javascript SDK
 v3. Thank you!",
);

const myScenario = new Scenario(
  "SSM Basics",
  [
    greet,
    pressEnter,
    createMaintenanceWindow,
    getMaintenanceWindow,
    sdkCreateMaintenanceWindow,
    modifyMaintenanceWindow,
    pressEnter,
    sdkModifyMaintenanceWindow,
    createSystemsManagerActions,
    getDocumentName,
    sdkCreateSSMDoc,
    ec2HelloWorld,
    enterIdOrSkipEC2HelloWorld,
    sdkEC2HelloWorld,
    sdkGetCommandTime,
    pressEnter,
    createSSMOpsItem,
    pressEnter,
    sdkCreateSSMOpsItem,
    updateOpsItem,
    pressEnter,
    sdkUpdateOpsItem,
    getOpsItemStatus,
    pressEnter,
    sdkOpsItemStatus,
    resolveOpsItem,
    pressEnter,
    sdkResolveOpsItem,
    askToDeleteResources,
    confirmDeleteChoice,
    sdkDeleteResources,
```

```
    goodbye,
  ],
  { ssmClient: new SSMClient({}) },
);

/** @type {{ stepHandlerOptions: StepHandlerOptions }} */
export const main = async (stepHandlerOptions) => {
  await myScenario.run(stepHandlerOptions);
};

// Invoke main function if this file was run directly.
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  const { values } = parseArgs({
    options: {
      yes: {
        type: "boolean",
        short: "y",
      },
    },
  });
  main({ confirmAll: values.yes });
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK para JavaScript.
  - CreateDocument
  - CreateMaintenanceWindow
  - CreateOpsItem
  - DeleteMaintenanceWindow
  - ListCommandInvocations
  - SendCommand
  - UpdateOpsItem

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

Execute um cenário interativo em um prompt de comando.

```
class SystemsManagerScenario:
    """Runs an interactive scenario that shows how to get started using Amazon
 Systems Manager."""

    def __init__(self, document_wrapper, maintenance_window_wrapper,
 ops_item_wrapper):
        """
        :param document_wrapper: An object that wraps Systems Manager document
 functions.
        :param maintenance_window_wrapper: An object that wraps Systems Manager
 maintenance window functions.
        :param ops_item_wrapper: An object that wraps Systems Manager OpsItem
 functions.
        """
        self.document_wrapper = document_wrapper
        self.maintenance_window_wrapper = maintenance_window_wrapper
        self.ops_item_wrapper = ops_item_wrapper

    def run(self):
        """Demonstrates how to use the AWS SDK for Python (Boto3) to get started
 with Systems Manager."""
        try:
            print("-" * 88)
            print(
                """
Welcome to the AWS Systems Manager SDK Getting Started scenario.
This program demonstrates how to interact with Systems Manager using the AWS SDK
 for Python (Boto3).
Systems Manager is the operations hub for your AWS applications and resources and
 a secure end-to-end management
```

```
solution. The program's primary functions include creating a maintenance window,
 creating a document, sending a
command to a document, listing documents, listing commands, creating an OpsItem,
 modifying an OpsItem, and deleting
Systems Manager resources. Upon completion of the program, all AWS resources are
 cleaned up.
Let's get started..."""
            )
            q.ask("Please hit Enter")

            print("-" * 88)
            print("Create a Systems Manager maintenance window.")
            maintenance_window_name = q.ask(
                "Please enter the maintenance window name (default is ssm-
maintenance-window):",
            )
            if not maintenance_window_name:
                maintenance_window_name = "ssm-maintenance-window"

            self.maintenance_window_wrapper.create(
                name=maintenance_window_name,
                schedule="cron(0 10 ? * MON-FRI *)",
                duration=2,
                cutoff=1,
                allow_unassociated_targets=True,
            )

            print("-" * 88)
            print("Modify the maintenance window by changing the schedule")
            q.ask("Please hit Enter")

            self.maintenance_window_wrapper.update(
                name=maintenance_window_name,
                schedule="cron(0 0 ? * MON *)",
                duration=24,
                cutoff=1,
                allow_unassociated_targets=True,
                enabled=True,
            )

            print("-" * 88)
            print(
                "Create a document that defines the actions that Systems Manager
 performs on your EC2 instance."
```

```
            )
            document_name = q.ask(
                "Please enter the document name (default is ssmdocument):"
            )

            if not document_name:
                document_name = "ssmdocument"

            self.document_wrapper.create(
                name=document_name,
                content="""
{
    "schemaVersion": "2.2",
    "description": "Run a simple shell command",
    "mainSteps": [
        {
            "action": "aws:runShellScript",
            "name": "runEchoCommand",
            "inputs": {
              "runCommand": [
                "echo 'Hello, world!'"
              ]
            }
        }
    ]
}
            """,
            )

            self.document_wrapper.wait_until_active()

            print(
                """
Now you have the option of running a command on an EC2 instance that echoes
 'Hello, world!'.
In order to run this command, you must provide the instance ID of a Linux EC2
 instance. If you do
not already have a running Linux EC2 instance in your account, you can create one
 using the AWS console.
For information about creating an EC2 instance, see
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-launch-instance-
wizard.html.
            """
            )
```

```
            if q.ask(
                "Would you like to run a command on an EC2 instance? (y/n)",
                q.is_yesno,
            ):
                instance_id = q.ask(
                    "Please enter the instance ID of the EC2 instance:",
 q.non_empty
                )
                command_id = self.document_wrapper.send_command(
                    instance_ids=[instance_id]
                )

                self.document_wrapper.wait_command_executed(
                    command_id=command_id, instance_id=instance_id
                )

                print("-" * 88)
                print(
                    "Lets get the time when the specific command was sent to the
 specific managed node"
                )
                q.ask("Please hit Enter")


 self.document_wrapper.list_command_invocations(instance_id=instance_id)

            print("-" * 88)
            print("-" * 88)
            print(
                """
Now we will create a  Systems Manager OpsItem.
An OpsItem is a feature provided by the Systems Manager service.
It is a type of operational data item that allows you to manage and track various
 operational issues,
events, or tasks within your AWS environment.

You can create OpsItems to track and manage operational issues as they arise.
For example, you could create an OpsItem whenever your application detects a
 critical error
or an anomaly in your infrastructure.
            """
            )
            q.ask("Please hit Enter")
```

```
            self.ops_item_wrapper.create(
                title="Disk Space Alert",
                description="Created by the Systems Manager Python (Boto3) API",
                source="EC2",
                category="Performance",
                severity="2",
            )

            print("-" * 88)
            print("-" * 88)
            print(f"Now we will update  the OpsItem {self.ops_item_wrapper.id}")
            q.ask("Please hit Enter")

            self.ops_item_wrapper.update(
                title="Disk Space Alert",
                description=f"An update to {self.ops_item_wrapper.id}",
            )

            print(
                f"Now we will get the status of the OpsItem
{self.ops_item_wrapper.id}"
            )
            q.ask("Please hit Enter")

            # It may take a second for the ops item to be available
            counter = 0
            while not self.ops_item_wrapper.describe() and counter < 5:
                counter += 1
                time.sleep(1)

            print(f"Now we will resolve the OpsItem {self.ops_item_wrapper.id}")
            q.ask("Please hit Enter")

            self.ops_item_wrapper.update(status="Resolved")

            print("-" * 88)
            print("-" * 88)
            if q.ask(
                "Would you like to delete the Systems Manager resources? (y/n)",
                q.is_yesno,
            ):
                print("You selected to delete the resources.")
                self.cleanup()
```

```
            else:
                print("The Systems Manager resources will not be deleted")

            print("-" * 88)
            print("This concludes the Systems Manager SDK Getting Started
 scenario.")
            print("-" * 88)

        except Exception:
            self.cleanup()
            raise

    def cleanup(self):
        self.maintenance_window_wrapper.delete()
        self.ops_item_wrapper.delete()
        self.document_wrapper.delete()


if __name__ == "__main__":
    try:
        scenario = SystemsManagerScenario(
            DocumentWrapper.from_client(),
            MaintenanceWindowWrapper.from_client(),
            OpsItemWrapper.from_client(),
        )
        scenario.run()
    except Exception:
        logging.exception("Something went wrong with the demo.")
```

Defina uma classe que englobe as ações de documentos e comandos.

```
class DocumentWrapper:
    """Encapsulates AWS Systems Manager Document actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.name = None

    @classmethod
```

```
    def from_client(cls):
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def create(self, content, name):
        """
        Creates a document.

        :param content: The content of the document.
        :param name: The name of the document.
        """
        try:
            self.ssm_client.create_document(
                Name=name, Content=content, DocumentType="Command"
            )
            self.name = name
        except self.ssm_client.exceptions.DocumentAlreadyExists:
            print(f"Document {name} already exists.")
            self.name = name
        except ClientError as err:
            logger.error(
                "Couldn't create %s. Here's why: %s: %s",
                name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def delete(self):
        """
        Deletes an AWS Systems Manager document.
        """
        if self.name is None:
            return

        try:
            self.ssm_client.delete_document(Name=self.name)
            print(f"Deleted document {self.name}.")
            self.name = None
        except ClientError as err:
            logger.error(
                "Couldn't delete %s. Here's why: %s: %s",
```

```
                self.name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def send_command(self, instance_ids):
        """
        Sends a command to one or more instances.

        :param instance_ids: The IDs of the instances to send the command to.
        :return: The ID of the command.
        """
        try:
            response = self.ssm_client.send_command(
                InstanceIds=instance_ids, DocumentName=self.name,
TimeoutSeconds=3600
            )
            return response["Command"]["CommandId"]
        except ClientError as err:
            logger.error(
                "Couldn't send command to %s. Here's why: %s: %s",
                self.name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def describe(self):
        """
        Describes the document.

        :return: Document status.
        """
        try:
            response = self.ssm_client.describe_document(Name=self.name)
            return response["Document"]["Status"]
        except ClientError as err:
            logger.error(
                "Couldn't get %s. Here's why: %s: %s",
                self.name,
                err.response["Error"]["Code"],
```

```
                err.response["Error"]["Message"],
            )
            raise


    def wait_until_active(self, max_attempts=20, delay=5):
        """
        Waits until the document is active.

        :param max_attempts: The maximum number of attempts for checking the
status.
        :param delay: The delay in seconds between each check.
        """
        attempt = 0
        status = ""
        while attempt <= max_attempts:
            status = self.describe()
            if status == "Active":
                break
            attempt += 1
            time.sleep(delay)

        if status != "Active":
            logger.error("Document is not active.")
        else:
            logger.info("Document is active.")

    def wait_command_executed(self, command_id, instance_id):
        """
        Waits until the command is executed on the instance.

        :param command_id: The ID of the command.
        :param instance_id: The ID of the instance.
        """

        waiter = self.ssm_client.get_waiter("command_executed")
        waiter.wait(CommandId=command_id, InstanceId=instance_id)

    def list_command_invocations(self, instance_id):
        """
        Lists the commands for an instance.

        :param instance_id: The ID of the instance.
        :return: The list of commands.
```

```
        """
        try:
            paginator = self.ssm_client.get_paginator("list_command_invocations")
            command_invocations = []
            for page in paginator.paginate(InstanceId=instance_id):
                command_invocations.extend(page["CommandInvocations"])
            num_of_commands = len(command_invocations)
            print(
                f"{num_of_commands} command invocation(s) found for instance
{instance_id}."
            )

            if num_of_commands > 10:
                print("Displaying the first 10 commands:")
                num_of_commands = 10
            date_format = "%A, %d %B %Y %I:%M%p"
            for command in command_invocations[:num_of_commands]:
                print(
                    f"   The time of command invocation is
{command['RequestedDateTime'].strftime(date_format)}"
                )
        except ClientError as err:
            logger.error(
                "Couldn't list commands for %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

Defina uma classe que englobe as ações de itens de operação.

```
class OpsItemWrapper:
    """Encapsulates AWS Systems Manager OpsItem actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
```

```
        self.ssm_client = ssm_client
        self.id = None

    @classmethod
    def from_client(cls):
        """
        :return: A OpsItemWrapper instance.
        """
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def create(self, title, source, category, severity, description):
        """
        Create an OpsItem

        :param title: The OpsItem title.
        :param source: The OpsItem source.
        :param category: The OpsItem category.
        :param severity: The OpsItem severity.
        :param description: The OpsItem description.

        """
        try:
            response = self.ssm_client.create_ops_item(
                Title=title,
                Source=source,
                Category=category,
                Severity=severity,
                Description=description,
            )
            self.id = response["OpsItemId"]
        except self.ssm_client.exceptions.OpsItemLimitExceededException as err:
            logger.error(
                "Couldn't create ops item because you have exceeded your open
OpsItem limit. "
                "Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        except ClientError as err:
            logger.error(
                "Couldn't create ops item %s. Here's why: %s: %s",
```

```
                title,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise

    def delete(self):
        """
        Delete the OpsItem.
        """
        if self.id is None:
            return
        try:
            self.ssm_client.delete_ops_item(OpsItemId=self.id)
            print(f"Deleted ops item with id {self.id}")
            self.id = None
        except ClientError as err:
            logger.error(
                "Couldn't delete ops item %s. Here's why: %s: %s",
                self.id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def describe(self):
        """
        Describe an OpsItem.
        """
        try:
            paginator = self.ssm_client.get_paginator("describe_ops_items")
            ops_items = []
            for page in paginator.paginate(
                OpsItemFilters=[
                    {"Key": "OpsItemId", "Values": [self.id], "Operator":
"Equal"}
                ]
            ):
                ops_items.extend(page["OpsItemSummaries"])

            for item in ops_items:
                print(
```

```
                    f"The item title is {item['Title']} and the status is
{item['Status']}"
                )
            return len(ops_items) > 0
        except ClientError as err:
            logger.error(
                "Couldn't describe ops item %s. Here's why: %s: %s",
                self.id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def update(self, title=None, description=None, status=None):
        """
        Update an OpsItem.

        :param title: The new OpsItem title.
        :param description: The new OpsItem description.
        :param status: The new OpsItem status.
        :return:
        """
        args = dict(OpsItemId=self.id)
        if title is not None:
            args["Title"] = title
        if description is not None:
            args["Description"] = description
        if status is not None:
            args["Status"] = status
        try:
            self.ssm_client.update_ops_item(**args)
        except ClientError as err:
            logger.error(
                "Couldn't update ops item %s. Here's why: %s: %s",
                self.id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

Defina uma classe que englobe as ações de janelas de manutenção.

```
class MaintenanceWindowWrapper:
    """Encapsulates AWS Systems Manager maintenance window actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.window_id = None
        self.name = None

    @classmethod
    def from_client(cls):
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def create(self, name, schedule, duration, cutoff,
 allow_unassociated_targets):
        """
        Create an AWS Systems Manager maintenance window.

        :param name: The name of the maintenance window.
        :param schedule: The schedule of the maintenance window.
        :param duration: The duration of the maintenance window.
        :param cutoff: The cutoff time of the maintenance window.
        :param allow_unassociated_targets: Allow the maintenance window to run on
 managed nodes, even
                                           if you haven't registered those nodes
 as targets.
        """
        try:
            response = self.ssm_client.create_maintenance_window(
                Name=name,
                Schedule=schedule,
                Duration=duration,
                Cutoff=cutoff,
                AllowUnassociatedTargets=allow_unassociated_targets,
            )
```

```
            self.window_id = response["WindowId"]
            self.name = name
            logger.info("Created maintenance window %s.", self.window_id)
        except ParamValidationError as error:
            logger.error(
                "Parameter validation error when trying to create maintenance
window %s. Here's why: %s",
                self.window_id,
                error,
            )
            raise
        except ClientError as err:
            logger.error(
                "Couldn't create maintenance window %s. Here's why: %s: %s",
                name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def delete(self):
        """
        Delete the associated AWS Systems Manager maintenance window.
        """
        if self.window_id is None:
            return

        try:
            self.ssm_client.delete_maintenance_window(WindowId=self.window_id)
            logger.info("Deleted maintenance window %s.", self.window_id)
            print(f"Deleted maintenance window {self.name}")
            self.window_id = None
        except ClientError as err:
            logger.error(
                "Couldn't delete maintenance window %s. Here's why: %s: %s",
                self.window_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def update(
```

```
        self, name, enabled, schedule, duration, cutoff,
allow_unassociated_targets
    ):
        """
        Update an AWS Systems Manager maintenance window.

        :param name: The name of the maintenance window.
        :param enabled: Whether the maintenance window is enabled to run on
managed nodes.
        :param schedule: The schedule of the maintenance window.
        :param duration: The duration of the maintenance window.
        :param cutoff: The cutoff time of the maintenance window.
        :param allow_unassociated_targets: Allow the maintenance window to run on
managed nodes, even
                                           if you haven't registered those nodes
as targets.
        """
        try:
            self.ssm_client.update_maintenance_window(
                WindowId=self.window_id,
                Name=name,
                Enabled=enabled,
                Schedule=schedule,
                Duration=duration,
                Cutoff=cutoff,
                AllowUnassociatedTargets=allow_unassociated_targets,
            )
            self.name = name
            logger.info("Updated maintenance window %s.", self.window_id)
        except ParamValidationError as error:
            logger.error(
                "Parameter validation error when trying to update maintenance
window %s. Here's why: %s",
                self.window_id,
                error,
            )
            raise
        except ClientError as err:
            logger.error(
                "Couldn't update maintenance window %s. Here's why: %s: %s",
                self.name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
```

```
            raise
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência de API do AWS SDK para Python (Boto3).
  - CreateDocument
  - CreateMaintenanceWindow
  - CreateOpsItem
  - DeleteMaintenanceWindow
  - ListCommandInvocations
  - SendCommand
  - UpdateOpsItem

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Ações para o Systems Manager usando SDKs da AWS

Os exemplos de código a seguir demonstram como realizar ações específicas do Systems Manager com AWS SDKs. Cada exemplo inclui um link para o GitHub, em que você pode encontrar instruções para configurar e executar o código.

Os exemplos a seguir incluem apenas as ações mais utilizadas. Para obter uma lista completa, consulte a Referência de APIs do AWS Systems Manager.

Exemplos

- Usar AddTagsToResource com uma CLI
- Usar CancelCommand com uma CLI
- Usar CreateActivation com uma CLI
- Usar CreateAssociation com uma CLI
- Usar CreateAssociationBatch com uma CLI
- Usar CreateDocument com o AWS SDK ou a CLI
- Usar CreateMaintenanceWindow com o AWS SDK ou a CLI

- Usar CreateOpsItem com o AWS SDK ou a CLI
- Usar CreatePatchBaseline com uma CLI
- Usar DeleteActivation com uma CLI
- Usar DeleteAssociation com uma CLI
- Usar DeleteDocument com o AWS SDK ou a CLI
- Usar DeleteMaintenanceWindow com o AWS SDK ou a CLI
- Usar DeleteOpsItem com um SDK da AWS
- Usar DeleteParameter com uma CLI
- Usar DeletePatchBaseline com uma CLI
- Usar DeregisterManagedInstance com uma CLI
- Usar DeregisterPatchBaselineForPatchGroup com uma CLI
- Usar DeregisterTargetFromMaintenanceWindow com uma CLI
- Usar DeregisterTaskFromMaintenanceWindow com uma CLI
- Usar DescribeActivations com uma CLI
- Usar DescribeAssociation com uma CLI
- Usar DescribeAssociationExecutionTargets com uma CLI
- Usar DescribeAssociationExecutions com uma CLI
- Usar DescribeAutomationExecutions com uma CLI
- Usar DescribeAutomationStepExecutions com uma CLI
- Usar DescribeAvailablePatches com uma CLI
- Usar DescribeDocument com uma CLI
- Usar DescribeDocumentPermission com uma CLI
- Usar DescribeEffectiveInstanceAssociations com uma CLI
- Usar DescribeEffectivePatchesForPatchBaseline com uma CLI
- Usar DescribeInstanceAssociationsStatus com uma CLI
- Usar DescribeInstanceInformation com uma CLI
- Usar DescribeInstancePatchStates com uma CLI
- Usar DescribeInstancePatchStatesForPatchGroup com uma CLI
- Usar DescribeInstancePatches com uma CLI

- Usar DescribeMaintenanceWindowExecutionTaskInvocations com uma CLI
- Usar DescribeMaintenanceWindowExecutionTasks com uma CLI
- Usar DescribeMaintenanceWindowExecutions com uma CLI
- Usar DescribeMaintenanceWindowTargets com uma CLI
- Usar DescribeMaintenanceWindowTasks com uma CLI
- Usar DescribeMaintenanceWindows com uma CLI
- Usar DescribeOpsItems com o AWS SDK ou a CLI
- Usar DescribeParameters com o AWS SDK ou a CLI
- Usar DescribePatchBaselines com uma CLI
- Usar DescribePatchGroupState com uma CLI
- Usar DescribePatchGroups com uma CLI
- Usar GetAutomationExecution com uma CLI
- Usar GetCommandInvocation com uma CLI
- Usar GetConnectionStatus com uma CLI
- Usar GetDefaultPatchBaseline com uma CLI
- Usar GetDeployablePatchSnapshotForInstance com uma CLI
- Usar GetDocument com uma CLI
- Usar GetInventory com uma CLI
- Usar GetInventorySchema com uma CLI
- Usar GetMaintenanceWindow com uma CLI
- Usar GetMaintenanceWindowExecution com uma CLI
- Usar GetMaintenanceWindowExecutionTask com uma CLI
- Usar GetParameter com o AWS SDK ou a CLI
- Usar GetParameterHistory com uma CLI
- Usar GetParameters com uma CLI
- Usar GetPatchBaseline com uma CLI
- Usar GetPatchBaselineForPatchGroup com uma CLI
- Usar ListAssociationVersions com uma CLI
- Usar ListAssociations com uma CLI

- Usar ListCommandInvocations com o AWS SDK ou a CLI
- Usar ListCommands com uma CLI
- Usar ListComplianceItems com uma CLI
- Usar ListComplianceSummaries com uma CLI
- Usar ListDocumentVersions com uma CLI
- Usar ListDocuments com uma CLI
- Usar ListInventoryEntries com uma CLI
- Usar ListResourceComplianceSummaries com uma CLI
- Usar ListTagsForResource com uma CLI
- Usar ModifyDocumentPermission com uma CLI
- Usar PutComplianceItems com uma CLI
- Usar PutInventory com uma CLI
- Usar PutParameter com o AWS SDK ou a CLI
- Usar RegisterDefaultPatchBaseline com uma CLI
- Usar RegisterPatchBaselineForPatchGroup com uma CLI
- Usar RegisterTargetWithMaintenanceWindow com uma CLI
- Usar RegisterTaskWithMaintenanceWindow com uma CLI
- Usar RemoveTagsFromResource com uma CLI
- Usar SendCommand com o AWS SDK ou a CLI
- Usar StartAutomationExecution com uma CLI
- Usar StartSession com uma CLI
- Usar StopAutomationExecution com uma CLI
- Usar UpdateAssociation com uma CLI
- Usar UpdateAssociationStatus com uma CLI
- Usar UpdateDocument com uma CLI
- Usar UpdateDocumentDefaultVersion com uma CLI
- Usar UpdateMaintenanceWindow com o AWS SDK ou a CLI
- Usar UpdateManagedInstanceRole com uma CLI
- Usar UpdateOpsItem com o AWS SDK ou a CLI

- Usar UpdatePatchBaseline com uma CLI

Usar **AddTagsToResource** com uma CLI

Os exemplos de código a seguir mostram como usar o `AddTagsToResource`.

CLI

AWS CLI

Exemplo 1: adicionar tags a uma janela de manutenção

O exemplo de `add-tags-to-resource` a seguir adiciona uma tag à janela de manutenção especificada.

```
aws ssm add-tags-to-resource \
    --resource-type "MaintenanceWindow" \
    --resource-id "mw-03eb9db428EXAMPLE" \
    --tags "Key=Stack,Value=Production"
```

Este comando não produz saída.

Exemplo 2: adicionar tags a um parâmetro

O exemplo de `add-tags-to-resource` a seguir adiciona duas tags ao parâmetro especificado.

```
aws ssm add-tags-to-resource \
    --resource-type "Parameter" \
    --resource-id "My-Parameter" \
    --tags '[{"Key":"Region","Value":"East"},{"Key":"Environment",
 "Value":"Production"}]'
```

Este comando não produz saída.

Exemplo 3: adicionar tags a um documento do SSM

O exemplo de `add-tags-to-resource` a seguir adiciona uma tag ao documento especificado.

```
aws ssm add-tags-to-resource \
    --resource-type "Document" \
```

```
    --resource-id "My-Document" \
    --tags "Key=Quarter,Value=Q322"
```

Este comando não produz saída.

Para obter mais informações, consulte Marcar recursos do Systems Manager no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte AddTagsToResource na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo atualiza uma janela de manutenção com novas tags. Não haverá saída se o comando for bem-sucedido. A sintaxe usada nesse exemplo requer o PowerShell versão 3 ou posterior.

```
$option1 = @{Key="Stack";Value=@("Production")}
Add-SSMResourceTag -ResourceId "mw-03eb9db42890fb82d" -ResourceType
 "MaintenanceWindow" -Tag $option1
```

Exemplo 2: com o PowerShell versão 2, é necessário usar New-Object para criar cada tag. Não haverá saída se o comando for bem-sucedido.

```
$tag1 = New-Object Amazon.SimpleSystemsManagement.Model.Tag
$tag1.Key = "Stack"
$tag1.Value = "Production"

Add-SSMResourceTag -ResourceId "mw-03eb9db42890fb82d" -ResourceType
 "MaintenanceWindow" -Tag $tag1
```

- Para obter detalhes da API, consulte AddTagsToResource na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo atualiza uma janela de manutenção com novas tags. Não haverá saída se o comando for bem-sucedido. A sintaxe usada nesse exemplo requer o PowerShell versão 3 ou posterior.

```
$option1 = @{Key="Stack";Value=@("Production")}
Add-SSMResourceTag -ResourceId "mw-03eb9db42890fb82d" -ResourceType
 "MaintenanceWindow" -Tag $option1
```

Exemplo 2: com o PowerShell versão 2, é necessário usar New-Object para criar cada tag. Não haverá saída se o comando for bem-sucedido.

```
$tag1 = New-Object Amazon.SimpleSystemsManagement.Model.Tag
$tag1.Key = "Stack"
$tag1.Value = "Production"

Add-SSMResourceTag -ResourceId "mw-03eb9db42890fb82d" -ResourceType
 "MaintenanceWindow" -Tag $tag1
```

- Para obter detalhes da API, consulte AddTagsToResource na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`CancelCommand`** com uma CLI

Os exemplos de código a seguir mostram como usar o `CancelCommand`.

CLI

AWS CLI

Exemplo 1: cancelar um comando para todas as instâncias

O exemplo de `cancel-command` a seguir tenta cancelar o comando especificado que já está em execução para todas as instâncias.

```
aws ssm cancel-command \
    --command-id "662add3d-5831-4a10-b64a-f2ff3EXAMPLE"
```

Este comando não produz saída.

Exemplo 2: cancelar um comando para instâncias específicas

O exemplo de `cancel-command` a seguir tenta cancelar um comando somente para a instância especificada.

```
aws ssm cancel-command \
    --command-id "662add3d-5831-4a10-b64a-f2ff3EXAMPLE"
    --instance-ids "i-02573cafcfEXAMPLE"
```

Este comando não produz saída.

Para obter mais informações, consulte Marcar parâmetros do Systems Manager no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte CancelCommand na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo tenta cancelar um comando. Não haverá saída se a operação for bem-sucedida.

```
Stop-SSMCommand -CommandId "9ded293e-e792-4440-8e3e-7b8ec5feaa38"
```

- Para obter detalhes da API, consulte CancelCommand na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo tenta cancelar um comando. Não haverá saída se a operação for bem-sucedida.

```
Stop-SSMCommand -CommandId "9ded293e-e792-4440-8e3e-7b8ec5feaa38"
```

- Para obter detalhes da API, consulte CancelCommand na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`CreateActivation`** com uma CLI

Os exemplos de código a seguir mostram como usar o `CreateActivation`.

CLI

AWS CLI

Para criar uma ativação de instância gerenciada

O exemplo de `create-activation` a seguir cria uma ativação de instância gerenciada.

```
aws ssm create-activation \
    --default-instance-name "HybridWebServers" \
    --iam-role "HybridWebServersRole" \
    --registration-limit 5
```

Saída:

```
{
    "ActivationId": "5743558d-563b-4457-8682-d16c3EXAMPLE",
    "ActivationCode": "dRmgnYaFv567vEXAMPLE"
}
```

Para obter mais informações, consulte Etapa 4: criar uma ativação de instância gerenciada para um ambiente híbrido no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte CreateActivation na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo cria uma instância gerenciada.

```
New-SSMActivation -DefaultInstanceName "MyWebServers" -IamRole
 "SSMAutomationRole" -RegistrationLimit 10
```

Saída:

```
ActivationCode       ActivationId
--------------       ------------
KWChhOxBTiwDcKE9BlKC 08e51e79-1e36-446c-8e63-9458569c1363
```

- Para obter detalhes da API, consulte CreateActivation na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo cria uma instância gerenciada.

```
New-SSMActivation -DefaultInstanceName "MyWebServers" -IamRole
 "SSMAutomationRole" -RegistrationLimit 10
```

Saída:

```
ActivationCode       ActivationId
--------------       ------------
KWChhOxBTiwDcKE9BlKC 08e51e79-1e36-446c-8e63-9458569c1363
```

- Para obter detalhes da API, consulte CreateActivation na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`CreateAssociation`** com uma CLI

Os exemplos de código a seguir mostram como usar o `CreateAssociation`.

CLI

AWS CLI

Exemplo 1: associar um documento usando IDs de instância

Esse exemplo associa um documento de configuração a uma instância usando IDs de instância.

```
aws ssm create-association \
```

```
    --instance-id "i-0cb2b964d3e14fd9f" \
    --name "AWS-UpdateSSMAgent"
```

Saída:

```
{
    "AssociationDescription": {
        "Status": {
            "Date": 1487875500.33,
            "Message": "Associated with AWS-UpdateSSMAgent",
            "Name": "Associated"
        },
        "Name": "AWS-UpdateSSMAgent",
        "InstanceId": "i-0cb2b964d3e14fd9f",
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "AssociationId": "b7c3266e-a544-44db-877e-b20d3a108189",
        "DocumentVersion": "$DEFAULT",
        "LastUpdateAssociationDate": 1487875500.33,
        "Date": 1487875500.33,
        "Targets": [
            {
                "Values": [
                    "i-0cb2b964d3e14fd9f"
                ],
                "Key": "InstanceIds"
            }
        ]
    }
}
```

Para obter mais informações, consulte CreateAssociation na AWSReferência da API do Systems Manager.

Exemplo 2: associar um documento usando destinos

Esse exemplo associa um documento de configuração a uma instância usando destinos.

```
aws ssm create-association \
    --name "AWS-UpdateSSMAgent" \
```

```
    --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f"
```

Saída:

```
{
    "AssociationDescription": {
        "Status": {
            "Date": 1487875500.33,
            "Message": "Associated with AWS-UpdateSSMAgent",
            "Name": "Associated"
        },
        "Name": "AWS-UpdateSSMAgent",
        "InstanceId": "i-0cb2b964d3e14fd9f",
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "AssociationId": "b7c3266e-a544-44db-877e-b20d3a108189",
        "DocumentVersion": "$DEFAULT",
        "LastUpdateAssociationDate": 1487875500.33,
        "Date": 1487875500.33,
        "Targets": [
            {
                "Values": [
                    "i-0cb2b964d3e14fd9f"
                ],
                "Key": "InstanceIds"
            }
        ]
    }
}
```

Para obter mais informações, consulte CreateAssociation na AWSReferência da API do Systems Manager.

Exemplo 3: criar uma associação para ser executada somente uma vez

Esse exemplo cria uma nova associação que só é executada uma vez na data e na hora especificadas. As associações criadas com uma data no passado ou no presente (no momento em que são processadas, a data está no passado) são executadas imediatamente.

```
aws ssm create-association \
```

```
    --name "AWS-UpdateSSMAgent" \
    --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f" \
    --schedule-expression "at(2020-05-14T15:55:00)"  \
    --apply-only-at-cron-interval
```

Saída:

```
{
    "AssociationDescription": {
        "Status": {
            "Date": 1487875500.33,
            "Message": "Associated with AWS-UpdateSSMAgent",
            "Name": "Associated"
        },
        "Name": "AWS-UpdateSSMAgent",
        "InstanceId": "i-0cb2b964d3e14fd9f",
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "AssociationId": "b7c3266e-a544-44db-877e-b20d3a108189",
        "DocumentVersion": "$DEFAULT",
        "LastUpdateAssociationDate": 1487875500.33,
        "Date": 1487875500.33,
        "Targets": [
            {
                "Values": [
                    "i-0cb2b964d3e14fd9f"
                ],
                "Key": "InstanceIds"
            }
        ]
    }
}
```

Para obter mais informações, consulte CreateAssociation na Referência da API do AWS Systems Manager ou Referência: expressões cron e rate para Systems Manager no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte CreateAssociation na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo associa um documento de configuração a uma instância usando IDs de instância.

```
New-SSMAssociation -InstanceId "i-0cb2b964d3e14fd9f" -Name "AWS-UpdateSSMAgent"
```

Saída:

```
Name                  : AWS-UpdateSSMAgent
InstanceId            : i-0000293ffd8c57862
Date                  : 2/23/2017 6:55:22 PM
Status.Name           : Associated
Status.Date           : 2/20/2015 8:31:11 AM
Status.Message        : Associated with AWS-UpdateSSMAgent
Status.AdditionalInfo :
```

Exemplo 2: esse exemplo associa um documento de configuração a uma instância usando destinos.

```
$target = @{Key="instanceids";Values=@("i-0cb2b964d3e14fd9f")}
New-SSMAssociation -Name "AWS-UpdateSSMAgent" -Target $target
```

Saída:

```
Name                  : AWS-UpdateSSMAgent
InstanceId            :
Date                  : 3/1/2017 6:22:21 PM
Status.Name           :
Status.Date           :
Status.Message        :
Status.AdditionalInfo :
```

Exemplo 3: esse exemplo associa um documento de configuração a uma instância usando destinos e parâmetros.

```
$target = @{Key="instanceids";Values=@("i-0cb2b964d3e14fd9f")}
$params = @{
```

```
  "action"="configure"
  "mode"="ec2"
  "optionalConfigurationSource"="ssm"
  "optionalConfigurationLocation"=""
  "optionalRestart"="yes"
}
New-SSMAssociation -Name "Configure-CloudWatch" -AssociationName
 "CWConfiguration" -Target $target -Parameter $params
```

Saída:

```
Name                  : Configure-CloudWatch
InstanceId            :
Date                  : 5/17/2018 3:17:44 PM
Status.Name           :
Status.Date           :
Status.Message        :
Status.AdditionalInfo :
```

Exemplo 4: esse exemplo cria uma associação com todas as instâncias na região, com **AWS-GatherSoftwareInventory**. Ele também fornece arquivos personalizados e locais de registro nos parâmetros a serem coletados

```
$params =
 [Collections.Generic.Dictionary[String,Collections.Generic.List[String]]]::new()
$params["windowsRegistry"] ='[{"Path":"HKEY_LOCAL_MACHINE\SOFTWARE\Amazon
\MachineImage","Recursive":false,"ValueNames":["AMIName"]}]'
$params["files"] = '[{"Path":"C:\Program Files","Pattern":
["*.exe"],"Recursive":true}, {"Path":"C:\ProgramData","Pattern":
["*.log"],"Recursive":true}]'
New-SSMAssociation -AssociationName new-in-mum -Name AWS-GatherSoftwareInventory
 -Target @{Key="instanceids";Values="*"} -Parameter $params -region ap-south-1 -
ScheduleExpression "rate(720 minutes)"
```

Saída:

```
Name                  : AWS-GatherSoftwareInventory
InstanceId            :
Date                  : 6/9/2019 8:57:56 AM
Status.Name           :
Status.Date           :
Status.Message        :
```

```
Status.AdditionalInfo :
```

- Para obter detalhes da API, consulte CreateAssociation na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo associa um documento de configuração a uma instância usando IDs de instância.

```
New-SSMAssociation -InstanceId "i-0cb2b964d3e14fd9f" -Name "AWS-UpdateSSMAgent"
```

Saída:

```
Name                 : AWS-UpdateSSMAgent
InstanceId           : i-0000293ffd8c57862
Date                 : 2/23/2017 6:55:22 PM
Status.Name          : Associated
Status.Date          : 2/20/2015 8:31:11 AM
Status.Message       : Associated with AWS-UpdateSSMAgent
Status.AdditionalInfo :
```

Exemplo 2: esse exemplo associa um documento de configuração a uma instância usando destinos.

```
$target = @{Key="instanceids";Values=@("i-0cb2b964d3e14fd9f")}
New-SSMAssociation -Name "AWS-UpdateSSMAgent" -Target $target
```

Saída:

```
Name                 : AWS-UpdateSSMAgent
InstanceId           :
Date                 : 3/1/2017 6:22:21 PM
Status.Name          :
Status.Date          :
Status.Message       :
Status.AdditionalInfo :
```

Exemplo 3: esse exemplo associa um documento de configuração a uma instância usando destinos e parâmetros.

```
$target = @{Key="instanceids";Values=@("i-0cb2b964d3e14fd9f")}
```

```
$params = @{
  "action"="configure"
  "mode"="ec2"
  "optionalConfigurationSource"="ssm"
  "optionalConfigurationLocation"=""
  "optionalRestart"="yes"
}
New-SSMAssociation -Name "Configure-CloudWatch" -AssociationName
 "CWConfiguration" -Target $target -Parameter $params
```

Saída:

```
Name                 : Configure-CloudWatch
InstanceId           :
Date                 : 5/17/2018 3:17:44 PM
Status.Name          :
Status.Date          :
Status.Message       :
Status.AdditionalInfo :
```

Exemplo 4: esse exemplo cria uma associação com todas as instâncias na região, com **AWS-GatherSoftwareInventory**. Ele também fornece arquivos personalizados e locais de registro nos parâmetros a serem coletados

```
$params =
 [Collections.Generic.Dictionary[String,Collections.Generic.List[String]]]::new()
$params["windowsRegistry"] ='[{"Path":"HKEY_LOCAL_MACHINE\SOFTWARE\Amazon
\MachineImage","Recursive":false,"ValueNames":["AMIName"]}]'
$params["files"] = '[{"Path":"C:\Program Files","Pattern":
["*.exe"],"Recursive":true}, {"Path":"C:\ProgramData","Pattern":
["*.log"],"Recursive":true}]'
New-SSMAssociation -AssociationName new-in-mum -Name AWS-GatherSoftwareInventory
 -Target @{Key="instanceids";Values="*"} -Parameter $params -region ap-south-1 -
ScheduleExpression "rate(720 minutes)"
```

Saída:

```
Name                 : AWS-GatherSoftwareInventory
InstanceId           :
Date                 : 6/9/2019 8:57:56 AM
Status.Name          :
Status.Date          :
```

```
Status.Message        :
Status.AdditionalInfo :
```

- Para obter detalhes da API, consulte CreateAssociation na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`CreateAssociationBatch`** com uma CLI

Os exemplos de código a seguir mostram como usar o `CreateAssociationBatch`.

CLI

AWS CLI

Para criar várias associações

Este exemplo associa um documento de configuração a várias instâncias. A saída retorna uma lista de operações bem e malsucedidas, se aplicável.

Comando:

```
aws ssm create-association-batch --entries "Name=AWS-
UpdateSSMAgent,InstanceId=i-1234567890abcdef0" "Name=AWS-
UpdateSSMAgent,InstanceId=i-9876543210abcdef0"
```

Saída:

```
{
  "Successful": [
      {
          "Name": "AWS-UpdateSSMAgent",
          "InstanceId": "i-1234567890abcdef0",
          "AssociationVersion": "1",
          "Date": 1550504725.007,
          "LastUpdateAssociationDate": 1550504725.007,
          "Status": {
              "Date": 1550504725.007,
              "Name": "Associated",
```

```
            "Message": "Associated with AWS-UpdateSSMAgent"
        },
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "DocumentVersion": "$DEFAULT",
        "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
        "Targets": [
            {
                "Key": "InstanceIds",
                "Values": [
                    "i-1234567890abcdef0"
                ]
            }
        ]
    },
    {
        "Name": "AWS-UpdateSSMAgent",
        "InstanceId": "i-9876543210abcdef0",
        "AssociationVersion": "1",
        "Date": 1550504725.057,
        "LastUpdateAssociationDate": 1550504725.057,
        "Status": {
            "Date": 1550504725.057,
            "Name": "Associated",
            "Message": "Associated with AWS-UpdateSSMAgent"
        },
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "DocumentVersion": "$DEFAULT",
        "AssociationId": "9c9f7f20-5154-4fed-a83e-0123456789ab",
        "Targets": [
            {
                "Key": "InstanceIds",
                "Values": [
                    "i-9876543210abcdef0"
                ]
            }
        ]
    }
],
```

```
  "Failed": []
}
```

- Para obter detalhes da API, consulte CreateAssociationBatch na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo associa um documento de configuração a várias instâncias. A saída retorna uma lista de operações bem e malsucedidas, se aplicável.

```
$option1 = @{InstanceId="i-0cb2b964d3e14fd9f";Name=@("AWS-UpdateSSMAgent")}
$option2 = @{InstanceId="i-0000293ffd8c57862";Name=@("AWS-UpdateSSMAgent")}
New-SSMAssociationFromBatch -Entry $option1,$option2
```

Saída:

```
Failed  Successful
------  ----------
{}      {Amazon.SimpleSystemsManagement.Model.FailedCreateAssociation,
 Amazon.SimpleSystemsManagement.Model.FailedCreateAsso...
```

Exemplo 2: esse exemplo mostrará os detalhes completos de uma operação bem-sucedida.

```
$option1 = @{InstanceId="i-0cb2b964d3e14fd9f";Name=@("AWS-UpdateSSMAgent")}
$option2 = @{InstanceId="i-0000293ffd8c57862";Name=@("AWS-UpdateSSMAgent")}
(New-SSMAssociationFromBatch -Entry $option1,$option2).Successful
```

- Para obter detalhes da API, consulte CreateAssociationBatch na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo associa um documento de configuração a várias instâncias. A saída retorna uma lista de operações bem e malsucedidas, se aplicável.

```
$option1 = @{InstanceId="i-0cb2b964d3e14fd9f";Name=@("AWS-UpdateSSMAgent")}
$option2 = @{InstanceId="i-0000293ffd8c57862";Name=@("AWS-UpdateSSMAgent")}
New-SSMAssociationFromBatch -Entry $option1,$option2
```

Saída:

```
Failed  Successful
------  ----------
{}      {Amazon.SimpleSystemsManagement.Model.FailedCreateAssociation,
 Amazon.SimpleSystemsManagement.Model.FailedCreateAsso...
```

Exemplo 2: esse exemplo mostrará os detalhes completos de uma operação bem-sucedida.

```
$option1 = @{InstanceId="i-0cb2b964d3e14fd9f";Name=@("AWS-UpdateSSMAgent")}
$option2 = @{InstanceId="i-0000293ffd8c57862";Name=@("AWS-UpdateSSMAgent")}
(New-SSMAssociationFromBatch -Entry $option1,$option2).Successful
```

- Para obter detalhes da API, consulte CreateAssociationBatch na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`CreateDocument`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `CreateDocument`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Conheça os conceitos básicos

CLI

AWS CLI

Para criar um documento

O exemplo de `create-document` a seguir cria um documento do Systems Manager.

```
aws ssm create-document \
    --content file://exampleDocument.yml \
    --name "Example" \
    --document-type "Automation" \
```

```
    --document-format YAML
```

Saída:

```
{
    "DocumentDescription": {
        "Hash":
 "fc2410281f40779e694a8b95975d0f9f316da8a153daa94e3d9921102EXAMPLE",
        "HashType": "Sha256",
        "Name": "Example",
        "Owner": "29884EXAMPLE",
        "CreatedDate": 1583256349.452,
        "Status": "Creating",
        "DocumentVersion": "1",
        "Description": "Document Example",
        "Parameters": [
            {
                "Name": "AutomationAssumeRole",
                "Type": "String",
                "Description": "(Required) The ARN of the role that allows
 Automation to perform the actions on your behalf. If no role is specified,
 Systems Manager Automation uses your IAM permissions to execute this document.",
                "DefaultValue": ""
            },
            {
                "Name": "InstanceId",
                "Type": "String",
                "Description": "(Required) The ID of the Amazon EC2 instance.",
                "DefaultValue": ""
            }
        ],
        "PlatformTypes": [
            "Windows",
            "Linux"
        ],
        "DocumentType": "Automation",
        "SchemaVersion": "0.3",
        "LatestVersion": "1",
        "DefaultVersion": "1",
        "DocumentFormat": "YAML",
        "Tags": []
    }
}
```

Para obter mais informações, consulte Criar documentos do Systems Manager no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte CreateDocument na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
/**
 * Creates an AWS SSM document asynchronously.
 *
 * @param docName The name of the document to create.
 * <p>
 * This method initiates an asynchronous request to create an SSM document.
 * If the request is successful, it prints the document status.
 * If an exception occurs, it handles the error appropriately.
 */
public void createSSMDoc(String docName) throws SsmException {
    String jsonData = """
    {
    "schemaVersion": "2.2",
    "description": "Run a simple shell command",
    "mainSteps": [
        {
            "action": "aws:runShellScript",
            "name": "runEchoCommand",
            "inputs": {
              "runCommand": [
                "echo 'Hello, world!'"
              ]
            }
          }
        ]
    }
```

```
        """;

        CreateDocumentRequest request = CreateDocumentRequest.builder()
                .content(jsonData)
                .name(docName)
                .documentType(DocumentType.COMMAND)
                .build();

        CompletableFuture<CreateDocumentResponse> future =
getAsyncClient().createDocument(request);
        future.thenAccept(response -> {
            System.out.println("The status of the SSM document is " +
response.documentDescription().status());
        }).exceptionally(ex -> {
            Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
            if (cause instanceof DocumentAlreadyExistsException) {
                throw new CompletionException(cause);
            } else if (cause instanceof SsmException) {
                throw new CompletionException(cause);
            } else {
                throw new RuntimeException(cause);
            }
        }).join();
    }
```

- Para obter detalhes sobre a API, consulte CreateDocument na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
import { CreateDocumentCommand, SSMClient } from "@aws-sdk/client-ssm";
```

```
import { parseArgs } from "node:util";

/**
 * Create an SSM document.
 * @param {{ content: string, name: string, documentType?: DocumentType }}
 */
export const main = async ({ content, name, documentType }) => {
  const client = new SSMClient({});
  try {
    const { documentDescription } = await client.send(
      new CreateDocumentCommand({
        Content: content, // The content for the new SSM document. The content
 must not exceed 64KB.
        Name: name,
        DocumentType: documentType, // Document format type can be JSON, YAML, or
 TEXT. The default format is JSON.
      }),
    );
    console.log("Document created successfully.");
    return { DocumentDescription: documentDescription };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "DocumentAlreadyExists") {
      console.warn(`${caught.message}. Did you provide a new document name?`);
    } else {
      throw caught;
    }
  }
};
```

- Para obter detalhes sobre a API, consulte CreateDocument na Referência da API do AWS SDK para JavaScript.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo cria um documento na sua conta. O documento deve estar em formato JSON. Para obter mais informações sobre como escrever um documento de configuração, consulte Documento de configuração na Referência da API do SSM.

```
New-SSMDocument -Content (Get-Content -Raw "c:\temp\RunShellScript.json") -Name
 "RunShellScript" -DocumentType "Command"
```

Saída:

```
CreatedDate     : 3/1/2017 1:21:33 AM
DefaultVersion  : 1
Description     : Run an updated script
DocumentType    : Command
DocumentVersion : 1
Hash            :
 1d5ce820e999ff051eb4841ed887593daf77120fd76cae0d18a53cc42e4e22c1
HashType        : Sha256
LatestVersion   : 1
Name            : RunShellScript
Owner           : 809632081692
Parameters      : {commands}
PlatformTypes   : {Linux}
SchemaVersion   : 2.0
Sha1            :
Status          : Creating
```

- Para obter detalhes da API, consulte CreateDocument na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo cria um documento na sua conta. O documento deve estar em formato JSON. Para obter mais informações sobre como escrever um documento de configuração, consulte Documento de configuração na Referência da API do SSM.

```
New-SSMDocument -Content (Get-Content -Raw "c:\temp\RunShellScript.json") -Name
 "RunShellScript" -DocumentType "Command"
```

Saída:

```
CreatedDate     : 3/1/2017 1:21:33 AM
DefaultVersion  : 1
Description     : Run an updated script
DocumentType    : Command
DocumentVersion : 1
```

```
Hash            :
 1d5ce820e999ff051eb4841ed887593daf77120fd76cae0d18a53cc42e4e22c1
HashType        : Sha256
LatestVersion   : 1
Name            : RunShellScript
Owner           : 809632081692
Parameters      : {commands}
PlatformTypes   : {Linux}
SchemaVersion   : 2.0
Sha1            :
Status          : Creating
```

- Para obter detalhes da API, consulte CreateDocument na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
class DocumentWrapper:
    """Encapsulates AWS Systems Manager Document actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.name = None

    @classmethod
    def from_client(cls):
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)
```

```
    def create(self, content, name):
        """
        Creates a document.

        :param content: The content of the document.
        :param name: The name of the document.
        """
        try:
            self.ssm_client.create_document(
                Name=name, Content=content, DocumentType="Command"
            )
            self.name = name
        except self.ssm_client.exceptions.DocumentAlreadyExists:
            print(f"Document {name} already exists.")
            self.name = name
        except ClientError as err:
            logger.error(
                "Couldn't create %s. Here's why: %s: %s",
                name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Para obter detalhes da API, consulte CreateDocument na Referência da API AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`CreateMaintenanceWindow`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `CreateMaintenanceWindow`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Conheça os conceitos básicos

CLI

AWS CLI

Exemplo 1: criar uma janela de manutenção

O exemplo de `create-maintenance-window` a seguir cria uma nova janela de manutenção que, a cada cinco minutos, e por até duas horas (conforme necessário), impede que novas tarefas sejam iniciadas dentro de uma hora do final da execução da janela de manutenção, permite destinos não associados (instâncias que você não registrou na janela de manutenção) e indica, por meio do uso de tags personalizadas, que seu criador pretende usá-la em um tutorial.

```
aws ssm create-maintenance-window \
    --name "My-Tutorial-Maintenance-Window" \
    --schedule "rate(5 minutes)" \
    --duration 2 --cutoff 1 \
    --allow-unassociated-targets \
    --tags "Key=Purpose,Value=Tutorial"
```

Saída:

```
{
    "WindowId": "mw-0c50858d01EXAMPLE"
}
```

Exemplo 2: criar uma janela de manutenção que é executada somente uma vez

O exemplo de `create-maintenance-window` a seguir cria uma nova janela de manutenção que só é executada uma vez na data e na hora especificadas.

```
aws ssm create-maintenance-window \
    --name My-One-Time-Maintenance-Window \
    --schedule "at(2020-05-14T15:55:00)" \
    --duration 5 \
    --cutoff 2 \
    --allow-unassociated-targets \
    --tags "Key=Environment,Value=Production"
```

Saída:

```
{
    "WindowId": "mw-01234567890abcdef"
}
```

Para obter mais informações, consulte Janelas de manutenção no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte CreateMaintenanceWindow na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
   /**
    * Creates an SSM maintenance window asynchronously.
    *
    * @param winName The name of the maintenance window.
    * @return The ID of the created or existing maintenance window.
    * <p>
    * This method initiates an asynchronous request to create an SSM maintenance
window.
    * If the request is successful, it prints the maintenance window ID.
    * If an exception occurs, it handles the error appropriately.
    */
   public String createMaintenanceWindow(String winName) throws SsmException,
DocumentAlreadyExistsException {
       CreateMaintenanceWindowRequest request =
CreateMaintenanceWindowRequest.builder()
               .name(winName)
               .description("This is my maintenance window")
               .allowUnassociatedTargets(true)
               .duration(2)
               .cutoff(1)
               .schedule("cron(0 10 ? * MON-FRI *)")
```

```
                .build();

        CompletableFuture<CreateMaintenanceWindowResponse> future =
getAsyncClient().createMaintenanceWindow(request);
        final String[] windowId = {null};
        future.whenComplete((response, ex) -> {
            if (response != null) {
                String maintenanceWindowId = response.windowId();
                System.out.println("The maintenance window id is " +
maintenanceWindowId);
                windowId[0] = maintenanceWindowId;
            } else {
                Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
                if (cause instanceof DocumentAlreadyExistsException) {
                    throw new CompletionException(cause);
                } else if (cause instanceof SsmException) {
                    throw new CompletionException(cause);
                } else {
                    throw new RuntimeException(cause);
                }
            }
        }).join();

        if (windowId[0] == null) {
            MaintenanceWindowFilter filter = MaintenanceWindowFilter.builder()
                    .key("name")
                    .values(winName)
                    .build();

            DescribeMaintenanceWindowsRequest winRequest =
DescribeMaintenanceWindowsRequest.builder()
                    .filters(filter)
                    .build();

            CompletableFuture<DescribeMaintenanceWindowsResponse> describeFuture
= getAsyncClient().describeMaintenanceWindows(winRequest);
            describeFuture.whenComplete((describeResponse, describeEx) -> {
                if (describeResponse != null) {
                    List<MaintenanceWindowIdentity> windows =
describeResponse.windowIdentities();
                    if (!windows.isEmpty()) {
                        windowId[0] = windows.get(0).windowId();
                        System.out.println("Window ID: " + windowId[0]);
```

```
                    } else {
                        System.out.println("Window not found.");
                        windowId[0] = "";
                    }
                } else {
                    Throwable describeCause = (describeEx instanceof
 CompletionException) ? describeEx.getCause() : describeEx;
                    throw new RuntimeException("Error describing maintenance
 windows: " + describeCause.getMessage(), describeCause);
                }
            }).join();
        }

        return windowId[0];
    }
```

- Para obter detalhes da API, consulte CreateMaintenanceWindow na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
import { CreateMaintenanceWindowCommand, SSMClient } from "@aws-sdk/client-ssm";
import { parseArgs } from "node:util";

/**
 * Create an SSM maintenance window.
 * @param {{ name: string, allowUnassociatedTargets: boolean, duration: number,
 cutoff: number, schedule: string, description?: string }}
 */
export const main = async ({
  name,
```

```
  allowUnassociatedTargets, // Allow the maintenance window to run on managed
 nodes, even if you haven't registered those nodes as targets.
  duration, // The duration of the maintenance window in hours.
  cutoff, // The number of hours before the end of the maintenance window that
 Amazon Web Services Systems Manager stops scheduling new tasks for execution.
  schedule, // The schedule of the maintenance window in the form of a cron or
 rate expression.
  description = undefined,
}) => {
  const client = new SSMClient({});

  try {
    const { windowId } = await client.send(
      new CreateMaintenanceWindowCommand({
        Name: name,
        Description: description,
        AllowUnassociatedTargets: allowUnassociatedTargets, // Allow the
 maintenance window to run on managed nodes, even if you haven't registered those
 nodes as targets.
        Duration: duration, // The duration of the maintenance window in hours.
        Cutoff: cutoff, // The number of hours before the end of the maintenance
 window that Amazon Web Services Systems Manager stops scheduling new tasks for
 execution.
        Schedule: schedule, // The schedule of the maintenance window in the form
 of a cron or rate expression.
      }),
    );
    console.log(`Maintenance window created with Id: ${windowId}`);
    return { WindowId: windowId };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "MissingParameter") {
      console.warn(`${caught.message}. Did you provide these values?`);
    } else {
      throw caught;
    }
  }
};
```

- Para obter detalhes da API, consulte CreateMaintenanceWindow na Referência da API do AWS SDK para JavaScript.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo cria uma nova janela de manutenção com o nome especificado que é executada às 16h toda terça-feiras por 4 horas, com um limite de 1 hora, e que permite destinos não associados.

```
New-SSMMaintenanceWindow -Name "MyMaintenanceWindow" -Duration 4 -Cutoff 1 -
AllowUnassociatedTarget $true -Schedule "cron(0 16 ? * TUE *)"
```

Saída:

```
mw-03eb53e1ea7383998
```

- Para obter detalhes da API, consulte CreateMaintenanceWindow na Reference de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo cria uma nova janela de manutenção com o nome especificado que é executada às 16h toda terça-feiras por 4 horas, com um limite de 1 hora, e que permite destinos não associados.

```
New-SSMMaintenanceWindow -Name "MyMaintenanceWindow" -Duration 4 -Cutoff 1 -
AllowUnassociatedTarget $true -Schedule "cron(0 16 ? * TUE *)"
```

Saída:

```
mw-03eb53e1ea7383998
```

- Para obter detalhes da API, consulte CreateMaintenanceWindow na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
class MaintenanceWindowWrapper:
    """Encapsulates AWS Systems Manager maintenance window actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.window_id = None
        self.name = None

    @classmethod
    def from_client(cls):
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def create(self, name, schedule, duration, cutoff,
 allow_unassociated_targets):
        """
        Create an AWS Systems Manager maintenance window.

        :param name: The name of the maintenance window.
        :param schedule: The schedule of the maintenance window.
        :param duration: The duration of the maintenance window.
        :param cutoff: The cutoff time of the maintenance window.
        :param allow_unassociated_targets: Allow the maintenance window to run on
 managed nodes, even
                                           if you haven't registered those nodes
 as targets.
        """
        try:
```

```
            response = self.ssm_client.create_maintenance_window(
                Name=name,
                Schedule=schedule,
                Duration=duration,
                Cutoff=cutoff,
                AllowUnassociatedTargets=allow_unassociated_targets,
            )
            self.window_id = response["WindowId"]
            self.name = name
            logger.info("Created maintenance window %s.", self.window_id)
        except ParamValidationError as error:
            logger.error(
                "Parameter validation error when trying to create maintenance
window %s. Here's why: %s",
                self.window_id,
                error,
            )
            raise
        except ClientError as err:
            logger.error(
                "Couldn't create maintenance window %s. Here's why: %s: %s",
                name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Para obter detalhes da API, consulte CreateMaintenanceWindow na Referência da API AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`CreateOpsItem`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `CreateOpsItem`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Conheça os conceitos básicos

CLI

AWS CLI

Como criar um OpsItems

O exemplo de `create-ops-item` usa a chave /aws/resources no OperationalData para criar um OpsItem com um recurso relacionado do Amazon DynamoDB.

```
aws ssm create-ops-item \
    --title "EC2 instance disk full" \
    --description "Log clean up may have failed which caused the disk to be full"
 \
    --priority 2 \
    --source ec2 \
    --operational-data '{"/aws/resources":{"Value":"[{\"arn
\": \"arn:aws:dynamodb:us-west-2:12345678:table/OpsItems
\"}]","Type":"SearchableString"}}' \
    --notifications Arn="arn:aws:sns:us-west-2:12345678:TestUser"
```

Saída:

```
{
    "OpsItemId": "oi-1a2b3c4d5e6f"
}
```

Para obter mais informações, consulte Criar OpsItems no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte CreateOpsItem na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
   /**
    * Creates an SSM OpsItem asynchronously.
    *
    * @param title The title of the OpsItem.
    * @param source The source of the OpsItem.
    * @param category The category of the OpsItem.
    * @param severity The severity of the OpsItem.
    * @return The ID of the created OpsItem.
    * <p>
    * This method initiates an asynchronous request to create an SSM OpsItem.
    * If the request is successful, it returns the OpsItem ID.
    * If an exception occurs, it handles the error appropriately.
    */
   public String createSSMOpsItem(String title, String source, String category,
String severity) {
       CreateOpsItemRequest opsItemRequest = CreateOpsItemRequest.builder()
               .description("Created by the SSM Java API")
               .title(title)
               .source(source)
               .category(category)
               .severity(severity)
               .build();

       CompletableFuture<CreateOpsItemResponse> future =
getAsyncClient().createOpsItem(opsItemRequest);

       try {
           CreateOpsItemResponse response = future.join();
           return response.opsItemId();
       } catch (CompletionException e) {
           Throwable cause = e.getCause();
           if (cause instanceof SsmException) {
```

```
                throw (SsmException) cause;
            } else {
                throw new RuntimeException(cause);
            }
        }
    }
```

- Para obter detalhes da API, consulte CreateOpsItem na Referência da API AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
import { CreateOpsItemCommand, SSMClient } from "@aws-sdk/client-ssm";
import { parseArgs } from "node:util";

/**
 * Create an SSM OpsItem.
 * @param {{ title: string, source: string, category?: string, severity?:
 string }}
 */
export const main = async ({
  title,
  source,
  category = undefined,
  severity = undefined,
}) => {
  const client = new SSMClient({});
  try {
    const { opsItemArn, opsItemId } = await client.send(
      new CreateOpsItemCommand({
        Title: title,
```

```
        Source: source, // The origin of the OpsItem, such as Amazon EC2 or
 Systems Manager.
        Category: category,
        Severity: severity,
      }),
    );
    console.log(`Ops item created with id: ${opsItemId}`);
    return { OpsItemArn: opsItemArn, OpsItemId: opsItemId };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "MissingParameter") {
      console.warn(`${caught.message}. Did you provide these values?`);
    } else {
      throw caught;
    }
  }
};
```

- Para obter detalhes da API, consulte CreateOpsItem na Referência da API AWS SDK para JavaScript.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
class OpsItemWrapper:
    """Encapsulates AWS Systems Manager OpsItem actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.id = None
```

```
    @classmethod
    def from_client(cls):
        """
        :return: A OpsItemWrapper instance.
        """
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def create(self, title, source, category, severity, description):
        """
        Create an OpsItem

        :param title: The OpsItem title.
        :param source: The OpsItem source.
        :param category: The OpsItem category.
        :param severity: The OpsItem severity.
        :param description: The OpsItem description.

        """
        try:
            response = self.ssm_client.create_ops_item(
                Title=title,
                Source=source,
                Category=category,
                Severity=severity,
                Description=description,
            )
            self.id = response["OpsItemId"]
        except self.ssm_client.exceptions.OpsItemLimitExceededException as err:
            logger.error(
                "Couldn't create ops item because you have exceeded your open
OpsItem limit. "
                "Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        except ClientError as err:
            logger.error(
                "Couldn't create ops item %s. Here's why: %s: %s",
                title,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
```

```
            )
            raise
```

- Para obter detalhes da API, consulte CreateOpsItem na Referência da API AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`CreatePatchBaseline`** com uma CLI

Os exemplos de código a seguir mostram como usar o `CreatePatchBaseline`.

CLI

AWS CLI

Exemplo 1: criar uma lista de referência de patches com aprovação automática

O exemplo de `create-patch-baseline` a seguir cria uma lista de referência de patches para o Windows Server que aprova patches para instâncias de produção sete dias após serem lançados pela Microsoft.

```
aws ssm create-patch-baseline \
    --name "Windows-Production-Baseline-AutoApproval" \
    --operating-system "WINDOWS" \
    --approval-
rules "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Critical,1
{Key=CLASSIFICATION,Values=[SecurityUpdates,Updates,UpdateRollups,CriticalUpdates]}]},App
 \
    --description "Baseline containing all updates approved for Windows Server
 production systems"
```

Saída:

```
{
    "BaselineId": "pb-045f10b4f3EXAMPLE"
}
```

Exemplo 2: criar uma lista de referência de patches com uma data limite para aprovação

O exemplo de `create-patch-baseline` a seguir cria uma lista de referência de patches para o Windows Server que aprova todos os patches para um ambiente de produção lançados até 7 de julho de 2020.

```
aws ssm create-patch-baseline \
    --name "Windows-Production-Baseline-AutoApproval" \
    --operating-system "WINDOWS" \
    --approval-
rules "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Critical,1
{Key=CLASSIFICATION,Values=[SecurityUpdates,Updates,UpdateRollups,CriticalUpdates]}]},App
 \
    --description "Baseline containing all updates approved for Windows Server
 production systems"
```

Saída:

```
{
    "BaselineId": "pb-045f10b4f3EXAMPLE"
}
```

Exemplo 3: criar uma lista de referência de patches com regras de aprovação armazenadas em um arquivo JSON

O exemplo de `create-patch-baseline` a seguir cria uma lista de referência de patches para o Amazon Linux 2017.09 que aprova patches para um ambiente de produção sete dias após seu lançamento, especifica regras de aprovação para a lista de referência de patches e especifica um repositório personalizado para patches.

```
aws ssm create-patch-baseline \
    --cli-input-json file://my-amazon-linux-approval-rules-and-repo.json
```

Conteúdo de `my-amazon-linux-approval-rules-and-repo.json`:

```
{
    "Name": "Amazon-Linux-2017.09-Production-Baseline",
    "Description": "My approval rules patch baseline for Amazon Linux 2017.09
 instances",
    "OperatingSystem": "AMAZON_LINUX",
    "Tags": [
```

```
        {
            "Key": "Environment",
            "Value": "Production"
        }
    ],
    "ApprovalRules": {
        "PatchRules": [
            {
                "ApproveAfterDays": 7,
                "EnableNonSecurity": true,
                "PatchFilterGroup": {
                    "PatchFilters": [
                        {
                            "Key": "SEVERITY",
                            "Values": [
                                "Important",
                                "Critical"
                            ]
                        },
                        {
                            "Key": "CLASSIFICATION",
                            "Values": [
                                "Security",
                                "Bugfix"
                            ]
                        },
                        {
                            "Key": "PRODUCT",
                            "Values": [
                                "AmazonLinux2017.09"
                            ]
                        }
                    ]
                }
            }
        ]
    },
    "Sources": [
        {
            "Name": "My-AL2017.09",
            "Products": [
                "AmazonLinux2017.09"
            ],
```

```
            "Configuration": "[amzn-main] \nname=amzn-main-Base
\nmirrorlist=http://repo./$awsregion./$awsdomain//$releasever/main/
mirror.list //nmirrorlist_expire=300//nmetadata_expire=300 \npriority=10
 \nfailovermethod=priority \nfastestmirror_enabled=0 \ngpgcheck=1
 \ngpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-amazon-ga \nenabled=1 \nretries=3
 \ntimeout=5\nreport_instanceid=yes"
        }
    ]
}
```

Exemplo 4: criar uma lista de referência de patches que especifica patches aprovados e rejeitados

O exemplo de `create-patch-baseline` a seguir especifica explicitamente os patches a serem aprovados e rejeitados como exceção às regras de aprovação padrão.

```
aws ssm create-patch-baseline \
    --name "Amazon-Linux-2017.09-Alpha-Baseline" \
    --description "My custom approve/reject patch baseline for Amazon Linux
 2017.09 instances" \
    --operating-system "AMAZON_LINUX" \
    --approved-patches "CVE-2018-1234567,example-pkg-EE-2018*.amzn1.noarch" \
    --approved-patches-compliance-level "HIGH" \
    --approved-patches-enable-non-security \
    --tags "Key=Environment,Value=Alpha"
```

Para obter mais informações, consulte Criar uma lista de referência de patches personalizada no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte CreatePatchBaseline na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo cria uma lista de referência de patches que aprova patches, sete dias após serem lançados pela Microsoft, para instâncias gerenciadas que executam o Windows Server 2019 em um ambiente de produção.

```
$rule = New-Object Amazon.SimpleSystemsManagement.Model.PatchRule
```

```
$rule.ApproveAfterDays = 7

$ruleFilters = New-Object Amazon.SimpleSystemsManagement.Model.PatchFilterGroup

$patchFilter = New-Object Amazon.SimpleSystemsManagement.Model.PatchFilter
$patchFilter.Key="PRODUCT"
$patchFilter.Values="WindowsServer2019"

$severityFilter = New-Object Amazon.SimpleSystemsManagement.Model.PatchFilter
$severityFilter.Key="MSRC_SEVERITY"
$severityFilter.Values.Add("Critical")
$severityFilter.Values.Add("Important")
$severityFilter.Values.Add("Moderate")

$classificationFilter = New-Object
 Amazon.SimpleSystemsManagement.Model.PatchFilter
$classificationFilter.Key = "CLASSIFICATION"
$classificationFilter.Values.Add( "SecurityUpdates" )
$classificationFilter.Values.Add( "Updates" )
$classificationFilter.Values.Add( "UpdateRollups" )
$classificationFilter.Values.Add( "CriticalUpdates" )

$ruleFilters.PatchFilters.Add($severityFilter)
$ruleFilters.PatchFilters.Add($classificationFilter)
$ruleFilters.PatchFilters.Add($patchFilter)
$rule.PatchFilterGroup = $ruleFilters

New-SSMPatchBaseline -Name "Production-Baseline-Windows2019" -Description
 "Baseline containing all updates approved for production systems" -
ApprovalRules_PatchRule $rule
```

Saída:

```
pb-0z4z6221c4296b23z
```

- Para obter detalhes da API, consulte CreatePatchBaseline na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo cria uma lista de referência de patches que aprova patches, sete dias após serem lançados pela Microsoft, para instâncias gerenciadas que executam o Windows Server 2019 em um ambiente de produção.

```
$rule = New-Object Amazon.SimpleSystemsManagement.Model.PatchRule
$rule.ApproveAfterDays = 7

$ruleFilters = New-Object Amazon.SimpleSystemsManagement.Model.PatchFilterGroup

$patchFilter = New-Object Amazon.SimpleSystemsManagement.Model.PatchFilter
$patchFilter.Key="PRODUCT"
$patchFilter.Values="WindowsServer2019"

$severityFilter = New-Object Amazon.SimpleSystemsManagement.Model.PatchFilter
$severityFilter.Key="MSRC_SEVERITY"
$severityFilter.Values.Add("Critical")
$severityFilter.Values.Add("Important")
$severityFilter.Values.Add("Moderate")

$classificationFilter = New-Object
 Amazon.SimpleSystemsManagement.Model.PatchFilter
$classificationFilter.Key = "CLASSIFICATION"
$classificationFilter.Values.Add( "SecurityUpdates" )
$classificationFilter.Values.Add( "Updates" )
$classificationFilter.Values.Add( "UpdateRollups" )
$classificationFilter.Values.Add( "CriticalUpdates" )

$ruleFilters.PatchFilters.Add($severityFilter)
$ruleFilters.PatchFilters.Add($classificationFilter)
$ruleFilters.PatchFilters.Add($patchFilter)
$rule.PatchFilterGroup = $ruleFilters

New-SSMPatchBaseline -Name "Production-Baseline-Windows2019" -Description
 "Baseline containing all updates approved for production systems" -
ApprovalRules_PatchRule $rule
```

Saída:

```
pb-0z4z6221c4296b23z
```

- Para obter detalhes da API, consulte CreatePatchBaseline na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DeleteActivation`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DeleteActivation`.

CLI

AWS CLI

Para excluir uma ativação de instância gerenciada

O exemplo de `delete-activation` a seguir exclui uma ativação de instância gerenciada.

```
aws ssm delete-activation \
    --activation-id "aa673477-d926-42c1-8757-1358cEXAMPLE"
```

Este comando não produz saída.

Para obter mais informações, consulte Configurar o AWS Systems Manager para ambientes híbridos no Guia do usuário do .AWS Systems Manager.

- Para obter detalhes da API, consulte DeleteActivation na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo exclui uma ativação. Não haverá saída se o comando for bem-sucedido.

```
Remove-SSMActivation -ActivationId "08e51e79-1e36-446c-8e63-9458569c1363"
```

- Para obter detalhes da API, consulte DeleteActivation na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo exclui uma ativação. Não haverá saída se o comando for bem-sucedido.

```
Remove-SSMActivation -ActivationId "08e51e79-1e36-446c-8e63-9458569c1363"
```

- Para obter detalhes da API, consulte DeleteActivation na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DeleteAssociation`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DeleteAssociation`.

CLI

AWS CLI

Exemplo 1: excluir uma associação usando o ID da associação

O exemplo de `delete-association` a seguir exclui a associação para o ID de associação especificado. Não haverá saída se o comando for bem-sucedido.

```
aws ssm delete-association \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Este comando não produz saída.

Para obter mais informações, consulte Editar e criar uma nova versão de uma associação no Guia do usuário do AWS Systems Manager.

Exemplo 2: excluir uma associação

O exemplo de `delete-association` a seguir exclui a associação entre uma instância e um documento. Não haverá saída se o comando for bem-sucedido.

```
aws ssm delete-association \
    --instance-id "i-1234567890abcdef0" \
    --name "AWS-UpdateSSMAgent"
```

Este comando não produz saída.

Para obter mais informações, consulte Trabalhar com associações no Systems Manager no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DeleteAssociation na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo exclui a associação entre uma instância e um documento. Não haverá saída se o comando for bem-sucedido.

```
Remove-SSMAssociation -InstanceId "i-0cb2b964d3e14fd9f" -Name "AWS-
UpdateSSMAgent"
```

- Para obter detalhes da API, consulte DeleteAssociation na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo exclui a associação entre uma instância e um documento. Não haverá saída se o comando for bem-sucedido.

```
Remove-SSMAssociation -InstanceId "i-0cb2b964d3e14fd9f" -Name "AWS-
UpdateSSMAgent"
```

- Para obter detalhes da API, consulte DeleteAssociation na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DeleteDocument`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `DeleteDocument`.

CLI

AWS CLI

Para excluir um documento

O exemplo de `delete-document` a seguir exclui um documento do Systems Manager.

```
aws ssm delete-document \
    --name "Example"
```

Este comando não produz saída.

Para obter mais informações, consulte Criar documentos do Systems Manager no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DeleteDocument na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
    /**
     * Deletes an AWS SSM document asynchronously.
     *
     * @param documentName The name of the document to delete.
     * <p>
     * This method initiates an asynchronous request to delete an SSM document.
     * If an exception occurs, it handles the error appropriately.
     */
    public void deleteDoc(String documentName) {
        DeleteDocumentRequest documentRequest = DeleteDocumentRequest.builder()
                .name(documentName)
                .build();
```

```
        CompletableFuture<Void> future = CompletableFuture.runAsync(() -> {
            getAsyncClient().deleteDocument(documentRequest)
                    .thenAccept(response -> {
                        System.out.println("The SSM document was successfully
deleted.");
                    })
                    .exceptionally(ex -> {
                        throw new CompletionException(ex);
                    }).join();
        }).exceptionally(ex -> {
            Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
            if (cause instanceof SsmException) {
                throw new RuntimeException("SSM error: " + cause.getMessage(),
cause);
            } else {
                throw new RuntimeException("Unexpected error: " +
cause.getMessage(), cause);
            }
        });

        try {
            future.join();
        } catch (CompletionException ex) {
            throw ex.getCause() instanceof RuntimeException ? (RuntimeException)
ex.getCause() : ex;
        }
    }
```

- Para obter detalhes da API, consulte DeleteDocument na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
import { DeleteDocumentCommand, SSMClient } from "@aws-sdk/client-ssm";
import { parseArgs } from "node:util";

/**
 * Delete an SSM document.
 * @param {{ documentName: string }}
 */
export const main = async ({ documentName }) => {
  const client = new SSMClient({});
  try {
    await client.send(new DeleteDocumentCommand({ Name: documentName }));
    console.log(`Document '${documentName}' deleted.`);
    return { Deleted: true };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "MissingParameter") {
      console.warn(`${caught.message}. Did you provide this value?`);
    } else {
      throw caught;
    }
  }
};
```

- Para obter detalhes da API, consulte DeleteDocument na Referência da API do AWS SDK para JavaScript.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo exclui um documento. Não haverá saída se o comando for bem-sucedido.

```
Remove-SSMDocument -Name "RunShellScript"
```

- Para obter detalhes da API, consulte DeleteDocument na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo exclui um documento. Não haverá saída se o comando for bem-sucedido.

```
Remove-SSMDocument -Name "RunShellScript"
```

- Para obter detalhes da API, consulte DeleteDocument na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
class DocumentWrapper:
    """Encapsulates AWS Systems Manager Document actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.name = None

    @classmethod
    def from_client(cls):
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def delete(self):
        """
        Deletes an AWS Systems Manager document.
        """
        if self.name is None:
            return

        try:
            self.ssm_client.delete_document(Name=self.name)
            print(f"Deleted document {self.name}.")
```

```
            self.name = None
        except ClientError as err:
            logger.error(
                "Couldn't delete %s. Here's why: %s: %s",
                self.name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Para obter detalhes da API, consulte DeleteDocument na Referência da API AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DeleteMaintenanceWindow`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `DeleteMaintenanceWindow`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Conheça os conceitos básicos

CLI

AWS CLI

Para excluir uma janela de manutenção

Este exemplo de `delete-maintenance-window` remove a janela de manutenção especificada.

```
aws ssm delete-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9"
```

Saída:

```
{
    "WindowId":"mw-1a2b3c4d5e6f7g8h9"
}
```

Para obter mais informações, consulte Excluir uma janela de manutenção (AWS CLI) no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DeleteMaintenanceWindow na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
   /**
    * Deletes an AWS SSM Maintenance Window asynchronously.
    *
    * @param winId The ID of the Maintenance Window to delete.
    * <p>
    * This method initiates an asynchronous request to delete an SSM Maintenance
Window.
    * If an exception occurs, it handles the error appropriately.
    */
   public void deleteMaintenanceWindow(String winId) {
       DeleteMaintenanceWindowRequest windowRequest =
DeleteMaintenanceWindowRequest.builder()
               .windowId(winId)
               .build();

       CompletableFuture<Void> future = CompletableFuture.runAsync(() -> {
           getAsyncClient().deleteMaintenanceWindow(windowRequest)
                   .thenAccept(response -> {
                       System.out.println("The maintenance window was
successfully deleted.");
                   })
```

```
                    .exceptionally(ex -> {
                        throw new CompletionException(ex);
                    }).join();
        }).exceptionally(ex -> {
            Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
            if (cause instanceof SsmException) {
                throw new RuntimeException("SSM error: " + cause.getMessage(),
cause);
            } else {
                throw new RuntimeException("Unexpected error: " +
cause.getMessage(), cause);
            }
        });

        try {
            future.join();
        } catch (CompletionException ex) {
            throw ex.getCause() instanceof RuntimeException ? (RuntimeException)
ex.getCause() : ex;
        }
    }
```

- Para obter detalhes da API, consulte DeleteMaintenanceWindow na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
import { DeleteMaintenanceWindowCommand, SSMClient } from "@aws-sdk/client-ssm";
import { parseArgs } from "node:util";

/**
```

```
 * Delete an SSM maintenance window.
 * @param {{ windowId: string }}
 */
export const main = async ({ windowId }) => {
  const client = new SSMClient({});
  try {
    await client.send(
      new DeleteMaintenanceWindowCommand({ WindowId: windowId }),
    );
    console.log(`Maintenance window '${windowId}' deleted.`);
    return { Deleted: true };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "MissingParameter") {
      console.warn(`${caught.message}. Did you provide this value?`);
    } else {
      throw caught;
    }
  }
};
```

- Para obter detalhes da API, consulte DeleteMaintenanceWindow na Referência da API do AWS SDK para JavaScript.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo remove uma janela de manutenção.

```
Remove-SSMMaintenanceWindow -WindowId "mw-06d59c1a07c022145"
```

Saída:

```
mw-06d59c1a07c022145
```

- Para obter detalhes da API, consulte DeleteMaintenanceWindow na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo remove uma janela de manutenção.

```
Remove-SSMMaintenanceWindow -WindowId "mw-06d59c1a07c022145"
```

Saída:

```
mw-06d59c1a07c022145
```

- Para obter detalhes da API, consulte DeleteMaintenanceWindow na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
class MaintenanceWindowWrapper:
    """Encapsulates AWS Systems Manager maintenance window actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.window_id = None
        self.name = None

    @classmethod
    def from_client(cls):
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def delete(self):
        """
        Delete the associated AWS Systems Manager maintenance window.
        """
```

```
        if self.window_id is None:
            return

        try:
            self.ssm_client.delete_maintenance_window(WindowId=self.window_id)
            logger.info("Deleted maintenance window %s.", self.window_id)
            print(f"Deleted maintenance window {self.name}")
            self.window_id = None
        except ClientError as err:
            logger.error(
                "Couldn't delete maintenance window %s. Here's why: %s: %s",
                self.window_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Para obter detalhes da API, consulte DeleteMaintenanceWindow na Referência da API AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DeleteOpsItem`** com um SDK da AWS

O código de exemplo a seguir mostra como usar `DeleteOpsItem`.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
class OpsItemWrapper:
```

```
    """Encapsulates AWS Systems Manager OpsItem actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.id = None

    @classmethod
    def from_client(cls):
        """
        :return: A OpsItemWrapper instance.
        """
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def delete(self):
        """
        Delete the OpsItem.
        """
        if self.id is None:
            return
        try:
            self.ssm_client.delete_ops_item(OpsItemId=self.id)
            print(f"Deleted ops item with id {self.id}")
            self.id = None
        except ClientError as err:
            logger.error(
                "Couldn't delete ops item %s. Here's why: %s: %s",
                self.id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Para obter detalhes da API, consulte DeleteOpsItem na Referência da API AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DeleteParameter`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DeleteParameter`.

CLI

AWS CLI

Para excluir um parâmetro

O exemplo de `delete-parameter` a seguir exclui o parâmetro único especificado.

```
aws ssm delete-parameter \
    --name "MyParameter"
```

Este comando não produz saída.

Para obter mais informações, consulte Trabalhar com o Parameter Store no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DeleteParameter na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo exclui um parâmetro. Não haverá saída se o comando for bem-sucedido.

```
Remove-SSMParameter -Name "helloWorld"
```

- Para obter detalhes da API, consulte DeleteParameter na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo exclui um parâmetro. Não haverá saída se o comando for bem-sucedido.

```
Remove-SSMParameter -Name "helloWorld"
```

- Para obter detalhes da API, consulte DeleteParameter na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DeletePatchBaseline`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DeletePatchBaseline`.

CLI

AWS CLI

Para excluir uma lista de referência de patches

O exemplo de `delete-patch-baseline` a seguir exclui a lista de referência de patches especificada.

```
aws ssm delete-patch-baseline \
    --baseline-id "pb-045f10b4f382baeda"
```

Saída:

```
{
    "BaselineId": "pb-045f10b4f382baeda"
}
```

Para obter mais informações, consulte Atualizar ou excluir uma lista de referência de patches (Console) no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DeletePatchBaseline na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo exclui uma lista de referência de patches.

```
Remove-SSMPatchBaseline -BaselineId "pb-045f10b4f382baeda"
```

Saída:

```
pb-045f10b4f382baeda
```

- Para obter detalhes da API, consulte DeletePatchBaseline na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo exclui uma lista de referência de patches.

```
Remove-SSMPatchBaseline -BaselineId "pb-045f10b4f382baeda"
```

Saída:

```
pb-045f10b4f382baeda
```

- Para obter detalhes da API, consulte DeletePatchBaseline na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DeregisterManagedInstance`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DeregisterManagedInstance`.

CLI

AWS CLI

Para cancelar o registro de uma instância gerenciada

O exemplo de `deregister-managed-instance` a seguir cancela o registro da instância gerenciada especificada.

```
aws ssm deregister-managed-instance \
    --instance-id 'mi-08ab247cdfEXAMPLE'
```

Este comando não produz saída.

Para obter mais informações, consulte Cancelar o registro de nós gerenciados em ambientes híbridos e multinuvem no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DeregisterManagedInstance na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo cancela o registro de uma instância gerenciada. Não haverá saída se o comando for bem-sucedido.

```
Unregister-SSMManagedInstance -InstanceId "mi-08ab247cdf1046573"
```

- Para obter detalhes da API, consulte DeregisterManagedInstance na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo cancela o registro de uma instância gerenciada. Não haverá saída se o comando for bem-sucedido.

```
Unregister-SSMManagedInstance -InstanceId "mi-08ab247cdf1046573"
```

- Para obter detalhes da API, consulte DeregisterManagedInstance na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **DeregisterPatchBaselineForPatchGroup** com uma CLI

Os exemplos de código a seguir mostram como usar o `DeregisterPatchBaselineForPatchGroup`.

CLI

AWS CLI

Para cancelar o registro de um grupo de patches de uma lista de referência de patches

O exemplo de `deregister-patch-baseline-for-patch-group` a seguir cancela o registro do grupo de patches especificado da lista de referência de patches especificada.

```
aws ssm deregister-patch-baseline-for-patch-group \
    --patch-group "Production" \
    --baseline-id "pb-0ca44a362fEXAMPLE"
```

Saída:

```
{
  "PatchGroup":"Production",
  "BaselineId":"pb-0ca44a362fEXAMPLE"
}
```

Para obter mais informações, consulte Adicionar um grupo de patches a uma lista de referência de patches no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DeregisterPatchBaselineForPatchGroup na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo cancela o registro de um grupo de patches de uma lista de referência de patches.

```
Unregister-SSMPatchBaselineForPatchGroup -BaselineId "pb-045f10b4f382baeda" -
PatchGroup "Production"
```

Saída:

```
BaselineId           PatchGroup
----------           ----------
pb-045f10b4f382baeda Production
```

- Para obter detalhes da API, consulte DeregisterPatchBaselineForPatchGroup na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo cancela o registro de um grupo de patches de uma lista de referência de patches.

```
Unregister-SSMPatchBaselineForPatchGroup -BaselineId "pb-045f10b4f382baeda" -
PatchGroup "Production"
```

Saída:

```
BaselineId           PatchGroup
----------           ----------
pb-045f10b4f382baeda Production
```

- Para obter detalhes da API, consulte DeregisterPatchBaselineForPatchGroup na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DeregisterTargetFromMaintenanceWindow`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DeregisterTargetFromMaintenanceWindow`.

CLI

AWS CLI

Para remover um destino de uma janela de manutenção

O exemplo de `deregister-target-from-maintenance-window` a seguir remove o destino especificado da janela de manutenção especificada.

```
aws ssm deregister-target-from-maintenance-window \
    --window-id "mw-ab12cd34ef56gh78" \
    --window-target-id "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
```

Saída:

```
{
    "WindowId":"mw-ab12cd34ef56gh78",
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Para obter mais informações, consulte Atualizar uma janela de manutenção (AWS CLI) no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DeregisterTargetFromMaintenanceWindow na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo remove um destino de uma janela de manutenção.

```
Unregister-SSMTargetFromMaintenanceWindow -WindowTargetId
 "6ab5c208-9fc4-4697-84b7-b02a6cc25f7d" -WindowId "mw-06cf17cbefcb4bf4f"
```

Saída:

```
WindowId             WindowTargetId
--------             --------------
mw-06cf17cbefcb4bf4f 6ab5c208-9fc4-4697-84b7-b02a6cc25f7d
```

- Para obter detalhes da API, consulte DeregisterTargetFromMaintenanceWindow na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo remove um destino de uma janela de manutenção.

```
Unregister-SSMTargetFromMaintenanceWindow -WindowTargetId
 "6ab5c208-9fc4-4697-84b7-b02a6cc25f7d" -WindowId "mw-06cf17cbefcb4bf4f"
```

Saída:

```
WindowId             WindowTargetId
--------             --------------
mw-06cf17cbefcb4bf4f 6ab5c208-9fc4-4697-84b7-b02a6cc25f7d
```

- Para obter detalhes da API, consulte DeregisterTargetFromMaintenanceWindow na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DeregisterTaskFromMaintenanceWindow`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DeregisterTaskFromMaintenanceWindow`.

CLI

AWS CLI

Para remover uma tarefa de uma janela de manutenção

O exemplo de `deregister-task-from-maintenance-window` a seguir remove a tarefa especificada da janela de manutenção especificada.

```
aws ssm deregister-task-from-maintenance-window \
    --window-id "mw-ab12cd34ef56gh78" \
    --window-task-id "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c"
```

Saída:

```
{
    "WindowTaskId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c",
    "WindowId":"mw-ab12cd34ef56gh78"
```

```
}
```

Para obter mais informações, consulte Tutoriais de janelas de manutenção do Systems Manager (AWS CLI) no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DeregisterTaskFromMaintenanceWindow na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo remove uma tarefa de uma janela de manutenção.

```
Unregister-SSMTaskFromMaintenanceWindow -WindowTaskId "f34a2c47-ddfd-4c85-
a88d-72366b69af1b" -WindowId "mw-03a342e62c96d31b0"
```

Saída:

```
WindowId             WindowTaskId
--------             ------------
mw-03a342e62c96d31b0 f34a2c47-ddfd-4c85-a88d-72366b69af1b
```

- Para obter detalhes da API, consulte DeregisterTaskFromMaintenanceWindow na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo remove uma tarefa de uma janela de manutenção.

```
Unregister-SSMTaskFromMaintenanceWindow -WindowTaskId "f34a2c47-ddfd-4c85-
a88d-72366b69af1b" -WindowId "mw-03a342e62c96d31b0"
```

Saída:

```
WindowId             WindowTaskId
--------             ------------
mw-03a342e62c96d31b0 f34a2c47-ddfd-4c85-a88d-72366b69af1b
```

- Para obter detalhes da API, consulte DeregisterTaskFromMaintenanceWindow na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeActivations`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeActivations`.

CLI

AWS CLI

Para descrever as ativações

O exemplo de `describe-activations` a seguir lista detalhes sobre as ativações em sua conta da AWS.

```
aws ssm describe-activations
```

Saída:

```
{
    "ActivationList": [
        {
            "ActivationId": "5743558d-563b-4457-8682-d16c3EXAMPLE",
            "Description": "Example1",
            "IamRole": "HybridWebServersRole,
            "RegistrationLimit": 5,
            "RegistrationsCount": 5,
            "ExpirationDate": 1584316800.0,
            "Expired": false,
            "CreatedDate": 1581954699.792
        },
        {
            "ActivationId": "3ee0322b-f62d-40eb-b672-13ebfEXAMPLE",
            "Description": "Example2",
            "IamRole": "HybridDatabaseServersRole",
            "RegistrationLimit": 5,
            "RegistrationsCount": 5,
            "ExpirationDate": 1580515200.0,
            "Expired": true,
            "CreatedDate": 1578064132.002
        },
```

```
    ]
}
```

Para obter mais informações, consulte Etapa 4: criar uma ativação de instância gerenciada para um ambiente híbrido no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribeActivations naReferência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo fornece detalhes sobre as ativações em sua conta.

```
Get-SSMActivation
```

Saída:

```
ActivationId        : 08e51e79-1e36-446c-8e63-9458569c1363
CreatedDate         : 3/1/2017 12:01:51 AM
DefaultInstanceName : MyWebServers
Description         :
ExpirationDate      : 3/2/2017 12:01:51 AM
Expired             : False
IamRole             : AutomationRole
RegistrationLimit   : 10
RegistrationsCount  : 0
```

- Para obter detalhes da API, consulte DescribeActivations na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo fornece detalhes sobre as ativações em sua conta.

```
Get-SSMActivation
```

Saída:

```
ActivationId        : 08e51e79-1e36-446c-8e63-9458569c1363
CreatedDate         : 3/1/2017 12:01:51 AM
```

```
DefaultInstanceName : MyWebServers
Description         :
ExpirationDate      : 3/2/2017 12:01:51 AM
Expired             : False
IamRole             : AutomationRole
RegistrationLimit   : 10
RegistrationsCount  : 0
```

- Para obter detalhes da API, consulte DescribeActivations na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeAssociation`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeAssociation`.

CLI

AWS CLI

Exemplo 1: obter detalhes de uma associação

O exemplo de `describe-association` a seguir descreve a associação para o ID de associação especificado.

```
aws ssm describe-association \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Saída:

```
{
    "AssociationDescription": {
        "Name": "AWS-GatherSoftwareInventory",
        "AssociationVersion": "1",
        "Date": 1534864780.995,
        "LastUpdateAssociationDate": 1543235759.81,
        "Overview": {
            "Status": "Success",
            "AssociationStatusAggregatedCount": {
```

```
                "Success": 2
            }
        },
        "DocumentVersion": "$DEFAULT",
        "Parameters": {
            "applications": [
                "Enabled"
            ],
            "awsComponents": [
                "Enabled"
            ],
            "customInventory": [
                "Enabled"
            ],
            "files": [
                ""
            ],
            "instanceDetailedInformation": [
                "Enabled"
            ],
            "networkConfig": [
                "Enabled"
            ],
            "services": [
                "Enabled"
            ],
            "windowsRegistry": [
                ""
            ],
            "windowsRoles": [
                "Enabled"
            ],
            "windowsUpdates": [
                "Enabled"
            ]
        },
        "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
        "Targets": [
            {
                "Key": "InstanceIds",
                "Values": [
                    "*"
                ]
            }
```

```
        ],
        "ScheduleExpression": "rate(24 hours)",
        "LastExecutionDate": 1550501886.0,
        "LastSuccessfulExecutionDate": 1550501886.0,
        "AssociationName": "Inventory-Association"
    }
}
```

Para obter mais informações, consulte Editar e criar uma nova versão de uma associação no Guia do usuário do AWS Systems Manager.

Exemplo 2: obter detalhes de uma associação para uma instância e um documento específicos

O exemplo de `describe-association` a seguir descreve a associação entre uma instância e um documento.

```
aws ssm describe-association \
    --instance-id "i-1234567890abcdef0" \
    --name "AWS-UpdateSSMAgent"
```

Saída:

```
{
    "AssociationDescription": {
        "Status": {
            "Date": 1487876122.564,
            "Message": "Associated with AWS-UpdateSSMAgent",
            "Name": "Associated"
        },
        "Name": "AWS-UpdateSSMAgent",
        "InstanceId": "i-1234567890abcdef0",
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Associated",
            "AssociationStatusAggregatedCount": {
                "Pending": 1
            }
        },
        "AssociationId": "d8617c07-2079-4c18-9847-1234567890ab",
        "DocumentVersion": "$DEFAULT",
        "LastUpdateAssociationDate": 1487876122.564,
```

```
        "Date": 1487876122.564,
        "Targets": [
            {
                "Values": [
                    "i-1234567890abcdef0"
                ],
                "Key": "InstanceIds"
            }
        ]
    }
}
```

Para obter mais informações, consulte Editar e criar uma nova versão de uma associação no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribeAssociation na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo descreve a associação entre uma instância e um documento.

```
Get-SSMAssociation -InstanceId "i-0000293ffd8c57862" -Name "AWS-UpdateSSMAgent"
```

Saída:

```
Name                  : AWS-UpdateSSMAgent
InstanceId            : i-0000293ffd8c57862
Date                  : 2/23/2017 6:55:22 PM
Status.Name           : Pending
Status.Date           : 2/20/2015 8:31:11 AM
Status.Message        : temp_status_change
Status.AdditionalInfo : Additional-Config-Needed
```

- Para obter detalhes da API, consulte DescribeAssociation na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo descreve a associação entre uma instância e um documento.

```
Get-SSMAssociation -InstanceId "i-0000293ffd8c57862" -Name "AWS-UpdateSSMAgent"
```

Saída:

```
Name                  : AWS-UpdateSSMAgent
InstanceId            : i-0000293ffd8c57862
Date                  : 2/23/2017 6:55:22 PM
Status.Name           : Pending
Status.Date           : 2/20/2015 8:31:11 AM
Status.Message        : temp_status_change
Status.AdditionalInfo : Additional-Config-Needed
```

- Para obter detalhes da API, consulte DescribeAssociation na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeAssociationExecutionTargets`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeAssociationExecutionTargets`.

CLI

AWS CLI

Para obter detalhes da execução de uma associação

O exemplo de `describe-association-execution-targets` a seguir descreve a execução da associação especificada.

```
aws ssm describe-association-execution-targets \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --execution-id "7abb6378-a4a5-4f10-8312-0123456789ab"
```

Saída:

```
{
    "AssociationExecutionTargets": [
```

```
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",
            "ResourceId": "i-1234567890abcdef0",
            "ResourceType": "ManagedInstance",
            "Status": "Success",
            "DetailedStatus": "Success",
            "LastExecutionDate": 1550505538.497,
            "OutputSource": {
                "OutputSourceId": "97fff367-fc5a-4299-aed8-0123456789ab",
                "OutputSourceType": "RunCommand"
            }
        }
    ]
}
```

Para obter mais informações, consulte Visualizar históricos de associações no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribeAssociationExecutionTargets na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo exibe o ID do recurso e seu status de execução que fazem parte dos destinos de execução da associação

```
Get-SSMAssociationExecutionTarget -AssociationId 123a45a0-
c678-9012-3456-78901234db5e -ExecutionId 123a45a0-c678-9012-3456-78901234db5e |
 Select-Object ResourceId, Status
```

Saída:

```
ResourceId          Status
----------          ------
i-0b1b2a3456f7a890b  Success
i-01c12a45d6fc7a89f  Success
i-0a1caf234f56d7dc8  Success
i-012a3fd45af6dbcfe  Failed
```

```
i-0ddc1df23c4a5fb67  Success
```

Exemplo 2: esse comando verifica a execução específica de uma automação específica desde ontem, onde documento de comandos está associado. Além disso, ele verifica se a execução da associação falhou e, em caso afirmativo, exibie os detalhes da invocação do comando para a execução junto com o ID da instância

```
$AssociationExecution= Get-SSMAssociationExecutionTarget -
AssociationId 1c234567-890f-1aca-a234-5a678d901cb0 -ExecutionId
 12345ca12-3456-2345-2b45-23456789012 |
    Where-Object {$_.LastExecutionDate -gt (Get-Date -Hour 00 -Minute
 00).AddDays(-1)}

foreach ($execution in $AssociationExecution) {
    if($execution.Status -ne 'Success'){
        Write-Output "There was an issue executing the association
 $($execution.AssociationId) on $($execution.ResourceId)"
        Get-SSMCommandInvocation -CommandId
 $execution.OutputSource.OutputSourceId -Detail:$true | Select-Object -
ExpandProperty CommandPlugins
    }
}
```

Saída:

```
There was an issue executing the association 1c234567-890f-1aca-a234-5a678d901cb0
 on i-0a1caf234f56d7dc8


Name                   : aws:runPowerShellScript
Output                 :
                         ----------ERROR-------
                         failed to run commands: exit status 1
OutputS3BucketName     :
OutputS3KeyPrefix      :
OutputS3Region         : eu-west-1
ResponseCode           : 1
ResponseFinishDateTime : 5/29/2019 11:04:49 AM
ResponseStartDateTime  : 5/29/2019 11:04:49 AM
StandardErrorUrl       :
StandardOutputUrl      :
Status                 : Failed
```

```
StatusDetails          : Failed
```

- Para obter detalhes da API, consulte DescribeAssociationExecutionTargets na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo exibe o ID do recurso e seu status de execução que fazem parte dos destinos de execução da associação

```
Get-SSMAssociationExecutionTarget -AssociationId 123a45a0-
c678-9012-3456-78901234db5e -ExecutionId 123a45a0-c678-9012-3456-78901234db5e |
 Select-Object ResourceId, Status
```

Saída:

```
ResourceId           Status
----------           ------
i-0b1b2a3456f7a890b  Success
i-01c12a45d6fc7a89f  Success
i-0a1caf234f56d7dc8  Success
i-012a3fd45af6dbcfe  Failed
i-0ddc1df23c4a5fb67  Success
```

Exemplo 2: esse comando verifica a execução específica de uma automação específica desde ontem, onde documento de comandos está associado. Além disso, ele verifica se a execução da associação falhou e, em caso afirmativo, exibe os detalhes da invocação do comando para a execução junto com o ID da instância

```
$AssociationExecution= Get-SSMAssociationExecutionTarget -
AssociationId 1c234567-890f-1aca-a234-5a678d901cb0 -ExecutionId
 12345ca12-3456-2345-2b45-23456789012 |
    Where-Object {$_.LastExecutionDate -gt (Get-Date -Hour 00 -Minute
 00).AddDays(-1)}

foreach ($execution in $AssociationExecution) {
    if($execution.Status -ne 'Success'){
        Write-Output "There was an issue executing the association
 $($execution.AssociationId) on $($execution.ResourceId)"
        Get-SSMCommandInvocation -CommandId
 $execution.OutputSource.OutputSourceId -Detail:$true | Select-Object -
ExpandProperty CommandPlugins
```

```
    }
}
```

Saída:

```
There was an issue executing the association 1c234567-890f-1aca-a234-5a678d901cb0
 on i-0a1caf234f56d7dc8


Name                   : aws:runPowerShellScript
Output                 :
                         ----------ERROR-------
                         failed to run commands: exit status 1
OutputS3BucketName     :
OutputS3KeyPrefix      :
OutputS3Region         : eu-west-1
ResponseCode           : 1
ResponseFinishDateTime : 5/29/2019 11:04:49 AM
ResponseStartDateTime  : 5/29/2019 11:04:49 AM
StandardErrorUrl       :
StandardOutputUrl      :
Status                 : Failed
StatusDetails          : Failed
```

- Para obter detalhes da API, consulte DescribeAssociationExecutionTargets na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeAssociationExecutions`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeAssociationExecutions`.

CLI

AWS CLI

Exemplo 1: obter detalhes de todas as execuções de uma associação

O exemplo de `describe-association-executions` a seguir descreve todas as execuções da associação especificada.

```
aws ssm describe-association-executions \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Saída:

```
{
    "AssociationExecutions": [
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "ExecutionId": "474925ef-1249-45a2-b93d-0123456789ab",
            "Status": "Success",
            "DetailedStatus": "Success",
            "CreatedTime": 1550505827.119,
            "ResourceCountByStatus": "{Success=1}"
        },
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",
            "Status": "Success",
            "DetailedStatus": "Success",
            "CreatedTime": 1550505536.843,
            "ResourceCountByStatus": "{Success=1}"
        },
        ...
    ]
}
```

Para obter mais informações, consulte Visualizar históricos de associações no Guia do usuário do AWS Systems Manager.

Exemplo 2: obter detalhes de todas as execuções de uma associação após uma data e uma hora específicas

O exemplo de `describe-association-executions` a seguir descreve todas as execuções de uma associação após a data e a hora especificadas.

```
aws ssm describe-association-executions \
```

```
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --filters "Key=CreatedTime,Value=2019-02-18T16:00:00Z,Type=GREATER_THAN"
```

Saída:

```
{
    "AssociationExecutions": [
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "ExecutionId": "474925ef-1249-45a2-b93d-0123456789ab",
            "Status": "Success",
            "DetailedStatus": "Success",
            "CreatedTime": 1550505827.119,
            "ResourceCountByStatus": "{Success=1}"
        },
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",
            "Status": "Success",
            "DetailedStatus": "Success",
            "CreatedTime": 1550505536.843,
            "ResourceCountByStatus": "{Success=1}"
        },
        ...
    ]
}
```

Para obter mais informações, consulte Visualizar históricos de associações no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribeAssociationExecutions na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo retorna as execuções para o ID de associação fornecido

```
Get-SSMAssociationExecution -AssociationId 123a45a0-c678-9012-3456-78901234db5e
```

Saída:

```
AssociationId         : 123a45a0-c678-9012-3456-78901234db5e
AssociationVersion    : 2
CreatedTime           : 3/2/2019 8:53:29 AM
DetailedStatus        :
ExecutionId           : 123a45a0-c678-9012-3456-78901234db5e
LastExecutionDate     : 1/1/0001 12:00:00 AM
ResourceCountByStatus : {Success=4}
Status                : Success
```

- Para obter detalhes da API, consulte DescribeAssociationExecutions na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo retorna as execuções para o ID de associação fornecido

```
Get-SSMAssociationExecution -AssociationId 123a45a0-c678-9012-3456-78901234db5e
```

Saída:

```
AssociationId         : 123a45a0-c678-9012-3456-78901234db5e
AssociationVersion    : 2
CreatedTime           : 3/2/2019 8:53:29 AM
DetailedStatus        :
ExecutionId           : 123a45a0-c678-9012-3456-78901234db5e
LastExecutionDate     : 1/1/0001 12:00:00 AM
ResourceCountByStatus : {Success=4}
Status                : Success
```

- Para obter detalhes da API, consulte DescribeAssociationExecutions na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeAutomationExecutions`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeAutomationExecutions`.

CLI

AWS CLI

Para descrever uma execução do Automation

O exemplo de `describe-automation-executions` a seguir exibe detalhes sobre uma execução do Automation.

```
aws ssm describe-automation-executions \
    --filters Key=ExecutionId,Values=73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

Saída:

```
{
    "AutomationExecutionMetadataList": [
        {
            "AutomationExecutionId": "73c8eef8-f4ee-4a05-820c-e354fEXAMPLE",
            "DocumentName": "AWS-StartEC2Instance",
            "DocumentVersion": "1",
            "AutomationExecutionStatus": "Success",
            "ExecutionStartTime": 1583737233.748,
            "ExecutionEndTime": 1583737234.719,
            "ExecutedBy": "arn:aws:sts::29884EXAMPLE:assumed-role/
mw_service_role/OrchestrationService",
            "LogFile": "",
            "Outputs": {},
            "Mode": "Auto",
            "Targets": [],
            "ResolvedTargets": {
                "ParameterValues": [],
                "Truncated": false
            },
            "AutomationType": "Local"
        }
    ]
}
```

Para obter mais informações, consulte Executar um fluxo de trabalho simples do Automation no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribeAutomationExecutions na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo descreve todas as execuções do Automation ativas e encerradas associadas à sua conta.

```
Get-SSMAutomationExecutionList
```

Saída:

```
AutomationExecutionId     : 4105a4fc-f944-11e6-9d32-8fb2db27a909
AutomationExecutionStatus : Failed
DocumentName              : AWS-UpdateLinuxAmi
DocumentVersion           : 1
ExecutedBy                : admin
ExecutionEndTime          : 2/22/2017 9:17:08 PM
ExecutionStartTime        : 2/22/2017 9:17:02 PM
LogFile                   :
Outputs                   : {[createImage.ImageId,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
```

Exemplo 2: esse exemplo exibe ExecutionID, documento, data e hora de início/término da execução para execuções com AutomationExecutionStatus diferente de "Êxito"

```
Get-SSMAutomationExecutionList | Where-Object AutomationExecutionStatus
 -ne "Success" | Select-Object AutomationExecutionId, DocumentName,
 AutomationExecutionStatus, ExecutionStartTime, ExecutionEndTime | Format-Table -
AutoSize
```

Saída:

```
AutomationExecutionId                DocumentName
 AutomationExecutionStatus ExecutionStartTime   ExecutionEndTime
--------------------                -----------
 ------------------------ ------------------   ----------------
e1d2bad3-4567-8901-ae23-456c7c8901be AWS-UpdateWindowsAmi
 Cancelled                 4/16/2019 5:37:04 AM 4/16/2019 5:47:29 AM
61234567-a7f8-90e1-2b34-567b8bf9012c Fixed-UpdateAmi
 Cancelled                 4/16/2019 5:33:04 AM 4/16/2019 5:40:15 AM
```

```
91234d56-7e89-0ac1-2aee-34ea5d6a7c89 AWS-UpdateWindowsAmi
 Failed                  4/16/2019 5:22:46 AM 4/16/2019 5:27:29 AM
```

- Para obter detalhes da API, consulte DescribeAutomationExecutions na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo descreve todas as execuções do Automation ativas e encerradas associadas à sua conta.

```
Get-SSMAutomationExecutionList
```

Saída:

```
AutomationExecutionId     : 4105a4fc-f944-11e6-9d32-8fb2db27a909
AutomationExecutionStatus : Failed
DocumentName              : AWS-UpdateLinuxAmi
DocumentVersion           : 1
ExecutedBy                : admin
ExecutionEndTime          : 2/22/2017 9:17:08 PM
ExecutionStartTime        : 2/22/2017 9:17:02 PM
LogFile                   :
Outputs                   : {[createImage.ImageId,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
```

Exemplo 2: esse exemplo exibe ExecutionID, documento, data e hora de início/término da execução para execuções com AutomationExecutionStatus diferente de "Êxito"

```
Get-SSMAutomationExecutionList | Where-Object AutomationExecutionStatus
 -ne "Success" | Select-Object AutomationExecutionId, DocumentName,
 AutomationExecutionStatus, ExecutionStartTime, ExecutionEndTime | Format-Table -
AutoSize
```

Saída:

```
AutomationExecutionId               DocumentName
 AutomationExecutionStatus ExecutionStartTime   ExecutionEndTime
--------------------               ------------
 ------------------------- ------------------   ----------------
e1d2bad3-4567-8901-ae23-456c7c8901be AWS-UpdateWindowsAmi
 Cancelled                 4/16/2019 5:37:04 AM 4/16/2019 5:47:29 AM
```

```
61234567-a7f8-90e1-2b34-567b8bf9012c Fixed-UpdateAmi
 Cancelled                 4/16/2019 5:33:04 AM 4/16/2019 5:40:15 AM
91234d56-7e89-0ac1-2aee-34ea5d6a7c89 AWS-UpdateWindowsAmi
 Failed                    4/16/2019 5:22:46 AM 4/16/2019 5:27:29 AM
```

- Para obter detalhes da API, consulte DescribeAutomationExecutions na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeAutomationStepExecutions`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeAutomationStepExecutions`.

CLI

AWS CLI

Exemplo 1: descrever todas as etapas de uma execução de automação

O exemplo de `describe-automation-step-executions` a seguir exibe detalhes sobre as etapas de uma execução do Automation.

```
aws ssm describe-automation-step-executions \
    --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

Saída:

```
{
    "StepExecutions": [
        {
            "StepName": "startInstances",
            "Action": "aws:changeInstanceState",
            "ExecutionStartTime": 1583737234.134,
            "ExecutionEndTime": 1583737234.672,
            "StepStatus": "Success",
            "Inputs": {
                "DesiredState": "\"running\"",
                "InstanceIds": "[\"i-0cb99161f6EXAMPLE\"]"
            },
```

```
            "Outputs": {
                "InstanceStates": [
                    "running"
                ]
            },
            "StepExecutionId": "95e70479-cf20-4d80-8018-7e4e2EXAMPLE",
            "OverriddenParameters": {}
        }
    ]
}
```

Exemplo 2: descrever uma etapa específica de uma execução do Automation

O exemplo de `describe-automation-step-executions` a seguir exibe detalhes sobre uma etapa específica de uma execução do Automation.

```
aws ssm describe-automation-step-executions \
    --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE \
    --filters Key=StepExecutionId,Values=95e70479-cf20-4d80-8018-7e4e2EXAMPLE
```

Para obter mais informações, consulte Executar um fluxo de trabalho do Automation passo a passo (Linha de comando) no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribeAutomationStepExecutions na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo exibe informações sobre todas as execuções de etapas ativas e encerradas em um fluxo de trabalho do Automation.

```
Get-SSMAutomationStepExecution -AutomationExecutionId e1d2bad3-4567-8901-
ae23-456c7c8901be | Select-Object StepName, Action, StepStatus
```

Saída:

```
StepName                Action                  StepStatus
--------                ------                  ----------
LaunchInstance          aws:runInstances        Success
```

```
OSCompatibilityCheck      aws:runCommand          Success
RunPreUpdateScript        aws:runCommand          Success
UpdateEC2Config           aws:runCommand          Cancelled
UpdateSSMAgent            aws:runCommand          Pending
UpdateAWSPVDriver         aws:runCommand          Pending
UpdateAWSEnaNetworkDriver aws:runCommand          Pending
UpdateAWSNVMe             aws:runCommand          Pending
InstallWindowsUpdates     aws:runCommand          Pending
RunPostUpdateScript       aws:runCommand          Pending
RunSysprepGeneralize      aws:runCommand          Pending
StopInstance              aws:changeInstanceState Pending
CreateImage               aws:createImage         Pending
TerminateInstance         aws:changeInstanceState Pending
```

- Para obter detalhes da API, consulte DescribeAutomationStepExecutions na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo exibe informações sobre todas as execuções de etapas ativas e encerradas em um fluxo de trabalho do Automation.

```
Get-SSMAutomationStepExecution -AutomationExecutionId e1d2bad3-4567-8901-
ae23-456c7c8901be | Select-Object StepName, Action, StepStatus
```

Saída:

```
StepName                  Action                  StepStatus
--------                  ------                  ----------
LaunchInstance            aws:runInstances        Success
OSCompatibilityCheck      aws:runCommand          Success
RunPreUpdateScript        aws:runCommand          Success
UpdateEC2Config           aws:runCommand          Cancelled
UpdateSSMAgent            aws:runCommand          Pending
UpdateAWSPVDriver         aws:runCommand          Pending
UpdateAWSEnaNetworkDriver aws:runCommand          Pending
UpdateAWSNVMe             aws:runCommand          Pending
InstallWindowsUpdates     aws:runCommand          Pending
RunPostUpdateScript       aws:runCommand          Pending
RunSysprepGeneralize      aws:runCommand          Pending
StopInstance              aws:changeInstanceState Pending
CreateImage               aws:createImage         Pending
TerminateInstance         aws:changeInstanceState Pending
```

- Para obter detalhes da API, consulte DescribeAutomationStepExecutions na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeAvailablePatches`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeAvailablePatches`.

CLI

AWS CLI

Para obter os patches disponíveis

O exemplo de `describe-available-patches` a seguir recupera detalhes sobre todos os patches disponíveis para o Windows Server 2019 que apresentam gravidade MSRC crítica.

```
aws ssm describe-available-patches \
    --
filters "Key=PRODUCT,Values=WindowsServer2019" "Key=MSRC_SEVERITY,Values=Critical"
```

Saída:

```
{
    "Patches": [
        {
            "Id": "fe6bd8c2-3752-4c8b-ab3e-1a7ed08767ba",
            "ReleaseDate": 1544047205.0,
            "Title": "2018-11 Update for Windows Server 2019 for x64-based
 Systems (KB4470788)",
            "Description": "Install this update to resolve issues in Windows.
 For a complete listing of the issues that are included in this update, see the
 associated Microsoft Knowledge Base article for more information. After you
 install this item, you may have to restart your computer.",
            "ContentUrl": "https://support.microsoft.com/en-us/kb/4470788",
            "Vendor": "Microsoft",
            "ProductFamily": "Windows",
            "Product": "WindowsServer2019",
```

```
            "Classification": "SecurityUpdates",
            "MsrcSeverity": "Critical",
            "KbNumber": "KB4470788",
            "MsrcNumber": "",
            "Language": "All"
        },
        {
            "Id": "c96115e1-5587-4115-b851-22baa46a3f11",
            "ReleaseDate": 1549994410.0,
            "Title": "2019-02 Security Update for Adobe Flash Player for Windows
 Server 2019 for x64-based Systems (KB4487038)",
            "Description": "A security issue has been identified in a Microsoft
 software product that could affect your system. You can help protect your system
 by installing this update from Microsoft. For a complete listing of the issues
 that are included in this update, see the associated Microsoft Knowledge Base
 article. After you install this update, you may have to restart your system.",
            "ContentUrl": "https://support.microsoft.com/en-us/kb/4487038",
            "Vendor": "Microsoft",
            "ProductFamily": "Windows",
            "Product": "WindowsServer2019",
            "Classification": "SecurityUpdates",
            "MsrcSeverity": "Critical",
            "KbNumber": "KB4487038",
            "MsrcNumber": "",
            "Language": "All"
        },
        ...
    ]
}
```

Para obter detalhes de um patch específico

O exemplo de `describe-available-patches` a seguir recupera detalhes do patch especificado.

```
aws ssm describe-available-patches \
    --filters "Key=PATCH_ID,Values=KB4480979"
```

Saída:

```
{
    "Patches": [
        {
```

```
            "Id": "680861e3-fb75-432e-818e-d72e5f2be719",
            "ReleaseDate": 1546970408.0,
            "Title": "2019-01 Security Update for Adobe Flash Player for Windows
 Server 2016 for x64-based Systems (KB4480979)",
            "Description": "A security issue has been identified in a Microsoft
 software product that could affect your system. You can help protect your system
 by installing this update from Microsoft. For a complete listing of the issues
 that are included in this update, see the associated Microsoft Knowledge Base
 article. After you install this update, you may have to restart your system.",
            "ContentUrl": "https://support.microsoft.com/en-us/kb/4480979",
            "Vendor": "Microsoft",
            "ProductFamily": "Windows",
            "Product": "WindowsServer2016",
            "Classification": "SecurityUpdates",
            "MsrcSeverity": "Critical",
            "KbNumber": "KB4480979",
            "MsrcNumber": "",
            "Language": "All"
        }
    ]
}
```

Para obter mais informações, consulte Como as operações do Patch Manager funcionam no Guia do Usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribeAvailablePatches na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo obtém todos os patches disponíveis para o Windows Server 2012 que apresentam gravidade MSRC crítica A sintaxe usada nesse exemplo requer o PowerShell versão 3 ou posterior.

```
$filter1 = @{Key="PRODUCT";Values=@("WindowsServer2012")}
$filter2 = @{Key="MSRC_SEVERITY";Values=@("Critical")}

Get-SSMAvailablePatch -Filter $filter1,$filter2
```

Saída:

```
Classification : SecurityUpdates
ContentUrl     : https://support.microsoft.com/en-us/kb/2727528
Description    : A security issue has been identified that could allow an
 unauthenticated remote attacker to compromise your system and gain control
                 over it. You can help protect your system by installing this
 update from Microsoft. After you install this update, you may have to
                 restart your system.
Id             : 1eb507be-2040-4eeb-803d-abc55700b715
KbNumber       : KB2727528
Language       : All
MsrcNumber     : MS12-072
MsrcSeverity   : Critical
Product        : WindowsServer2012
ProductFamily  : Windows
ReleaseDate    : 11/13/2012 6:00:00 PM
Title          : Security Update for Windows Server 2012 (KB2727528)
Vendor         : Microsoft
...
```

Exemplo 2: com o PowerShell versão 2, é necessário usar New-Object para criar cada filtro.

```
$filter1 = New-Object
 Amazon.SimpleSystemsManagement.Model.PatchOrchestratorFilter
$filter1.Key = "PRODUCT"
$filter1.Values = "WindowsServer2012"
$filter2 = New-Object
 Amazon.SimpleSystemsManagement.Model.PatchOrchestratorFilter
$filter2.Key = "MSRC_SEVERITY"
$filter2.Values = "Critical"

Get-SSMAvailablePatch -Filter $filter1,$filter2
```

Exemplo 3: esse exemplo busca todas as atualizações lançadas nos últimos 20 dias que são aplicáveis a produtos correspondentes ao Windows Server 2019

```
Get-SSMAvailablePatch | Where-Object ReleaseDate -ge (Get-Date).AddDays(-20)
 | Where-Object Product -eq "WindowsServer2019" | Select-Object ReleaseDate,
 Product, Title
```

Saída:

```
ReleaseDate         Product           Title
```

```
-----------         -------           -----
4/9/2019 5:00:12 PM WindowsServer2019 2019-04 Security Update for Adobe Flash
 Player for Windows Server 2019 for x64-based Systems (KB4493478)
4/9/2019 5:00:06 PM WindowsServer2019 2019-04 Cumulative Update for Windows
 Server 2019 for x64-based Systems (KB4493509)
4/2/2019 5:00:06 PM WindowsServer2019 2019-03 Servicing Stack Update for Windows
 Server 2019 for x64-based Systems (KB4493510)
```

- Para obter detalhes da API, consulte DescribeAvailablePatches na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo obtém todos os patches disponíveis para o Windows Server 2012 que apresentam gravidade MSRC crítica A sintaxe usada nesse exemplo requer o PowerShell versão 3 ou posterior.

```
$filter1 = @{Key="PRODUCT";Values=@("WindowsServer2012")}
$filter2 = @{Key="MSRC_SEVERITY";Values=@("Critical")}

Get-SSMAvailablePatch -Filter $filter1,$filter2
```

Saída:

```
Classification : SecurityUpdates
ContentUrl     : https://support.microsoft.com/en-us/kb/2727528
Description    : A security issue has been identified that could allow an
 unauthenticated remote attacker to compromise your system and gain control
                 over it. You can help protect your system by installing this
 update from Microsoft. After you install this update, you may have to
                 restart your system.
Id             : 1eb507be-2040-4eeb-803d-abc55700b715
KbNumber       : KB2727528
Language       : All
MsrcNumber     : MS12-072
MsrcSeverity   : Critical
Product        : WindowsServer2012
ProductFamily  : Windows
ReleaseDate    : 11/13/2012 6:00:00 PM
Title          : Security Update for Windows Server 2012 (KB2727528)
Vendor         : Microsoft
...
```

Exemplo 2: com o PowerShell versão 2, é necessário usar New-Object para criar cada filtro.

```
$filter1 = New-Object
 Amazon.SimpleSystemsManagement.Model.PatchOrchestratorFilter
$filter1.Key = "PRODUCT"
$filter1.Values = "WindowsServer2012"
$filter2 = New-Object
 Amazon.SimpleSystemsManagement.Model.PatchOrchestratorFilter
$filter2.Key = "MSRC_SEVERITY"
$filter2.Values = "Critical"

Get-SSMAvailablePatch -Filter $filter1,$filter2
```

Exemplo 3: esse exemplo busca todas as atualizações lançadas nos últimos 20 dias que são aplicáveis a produtos correspondentes ao Windows Server 2019

```
Get-SSMAvailablePatch | Where-Object ReleaseDate -ge (Get-Date).AddDays(-20)
 | Where-Object Product -eq "WindowsServer2019" | Select-Object ReleaseDate,
 Product, Title
```

Saída:

```
ReleaseDate         Product           Title
-----------         -------           -----
4/9/2019 5:00:12 PM WindowsServer2019 2019-04 Security Update for Adobe Flash
 Player for Windows Server 2019 for x64-based Systems (KB4493478)
4/9/2019 5:00:06 PM WindowsServer2019 2019-04 Cumulative Update for Windows
 Server 2019 for x64-based Systems (KB4493509)
4/2/2019 5:00:06 PM WindowsServer2019 2019-03 Servicing Stack Update for Windows
 Server 2019 for x64-based Systems (KB4493510)
```

- Para obter detalhes da API, consulte DescribeAvailablePatches na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeDocument`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeDocument`.

CLI

AWS CLI

Para exibir detalhes de um documento

O exemplo de `describe-document` a seguir exibe detalhes sobre um documento do Systems Manager em sua conta da AWS.

```
aws ssm describe-document \
    --name "Example"
```

Saída:

```
{
    "Document": {
        "Hash":
 "fc2410281f40779e694a8b95975d0f9f316da8a153daa94e3d9921102EXAMPLE",
        "HashType": "Sha256",
        "Name": "Example",
        "Owner": "29884EXAMPLE",
        "CreatedDate": 1583257938.266,
        "Status": "Active",
        "DocumentVersion": "1",
        "Description": "Document Example",
        "Parameters": [
            {
                "Name": "AutomationAssumeRole",
                "Type": "String",
                "Description": "(Required) The ARN of the role that allows
 Automation to perform the actions on your behalf. If no role is specified,
 Systems Manager Automation uses your IAM permissions to execute this document.",
                "DefaultValue": ""
            },
            {
                "Name": "InstanceId",
                "Type": "String",
                "Description": "(Required) The ID of the Amazon EC2 instance.",
                "DefaultValue": ""
            }
        ],
        "PlatformTypes": [
            "Windows",
```

```
            "Linux"
        ],
        "DocumentType": "Automation",
        "SchemaVersion": "0.3",
        "LatestVersion": "1",
        "DefaultVersion": "1",
        "DocumentFormat": "YAML",
        "Tags": []
    }
}
```

Para obter mais informações, consulte Criar documentos do Systems Manager no Guia do usuário do AWS Systems Manager.

- Para ver detalhes da API, consulte DescribeDocument na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo retorna informações sobre um documento.

```
Get-SSMDocumentDescription -Name "RunShellScript"
```

Saída:

```
CreatedDate     : 2/24/2017 5:25:13 AM
DefaultVersion  : 1
Description     : Run an updated script
DocumentType    : Command
DocumentVersion : 1
Hash            :
 f775e5df4904c6fa46686c4722fae9de1950dace25cd9608ff8d622046b68d9b
HashType        : Sha256
LatestVersion   : 1
Name            : RunShellScript
Owner           : 123456789012
Parameters      : {commands}
PlatformTypes   : {Linux}
SchemaVersion   : 2.0
Sha1            :
```

```
Status          : Active
```

- Para obter detalhes da API, consulte DescribeDocument na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo retorna informações sobre um documento.

```
Get-SSMDocumentDescription -Name "RunShellScript"
```

Saída:

```
CreatedDate     : 2/24/2017 5:25:13 AM
DefaultVersion  : 1
Description     : Run an updated script
DocumentType    : Command
DocumentVersion : 1
Hash            :
 f775e5df4904c6fa46686c4722fae9de1950dace25cd9608ff8d622046b68d9b
HashType        : Sha256
LatestVersion   : 1
Name            : RunShellScript
Owner           : 123456789012
Parameters      : {commands}
PlatformTypes   : {Linux}
SchemaVersion   : 2.0
Sha1            :
Status          : Active
```

- Para obter detalhes da API, consulte DescribeDocument na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeDocumentPermission`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeDocumentPermission`.

CLI

AWS CLI

Para descrever permissões do documento

O exemplo de `describe-document-permission` a seguir exibe detalhes de permissão sobre um documento do Systems Manager que é compartilhado publicamente.

```
aws ssm describe-document-permission \
    --name "Example" \
    --permission-type "Share"
```

Saída:

```
{
    "AccountIds": [
        "all"
    ],
    "AccountSharingInfoList": [
        {
            "AccountId": "all",
            "SharedDocumentVersion": "$DEFAULT"
        }
    ]
}
```

Para obter mais informações, consulte Compartilhar um documento do Systems Manager no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribeDocumentPermission na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo lista todas as versões de um documento.

```
Get-SSMDocumentVersionList -Name "RunShellScript"
```

Saída:

```
CreatedDate          DocumentVersion IsDefaultVersion Name
-----------          --------------- ---------------- ----
2/24/2017 5:25:13 AM 1               True             RunShellScript
```

- Para obter detalhes da API, consulte DescribeDocumentPermission na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo lista todas as versões de um documento.

```
Get-SSMDocumentVersionList -Name "RunShellScript"
```

Saída:

```
CreatedDate          DocumentVersion IsDefaultVersion Name
-----------          --------------- ---------------- ----
2/24/2017 5:25:13 AM 1               True             RunShellScript
```

- Para obter detalhes da API, consulte DescribeDocumentPermission na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeEffectiveInstanceAssociations`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeEffectiveInstanceAssociations`.

CLI

AWS CLI

Para obter detalhes das associações efetivas de uma instância

O exemplo de `describe-effective-instance-associations` a seguir recupera detalhes sobre as associações efetivas de uma instância.

Comando:

```
aws ssm describe-effective-instance-associations --instance-
id "i-1234567890abcdef0"
```

Saída:

```
{
    "Associations": [
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "InstanceId": "i-1234567890abcdef0",
            "Content": "{\n    \"schemaVersion\": \"1.2\",\n    \"description\":
 \"Update the Amazon SSM Agent to the latest version or specified version.\",\n
  \"parameters\": {\n        \"version\": {\n            \"default\": \"\",\n
       \"description\": \"(Optional) A specific version of the Amazon SSM Agent
 to install. If not specified, the agent will be updated to the latest version.
\",\n            \"type\": \"String\"\n        },\n        \"allowDowngrade
\": {\n            \"default\": \"false\",\n            \"description\":
 \"(Optional) Allow the Amazon SSM Agent service to be downgraded to an earlier
 version. If set to false, the service can be upgraded to newer versions only
 (default). If set to true, specify the earlier version.\",\n            \"type
\": \"String\",\n            \"allowedValues\": [\n                \"true\",\n
               \"false\"\n            ]\n        }\n    },\n    \"runtimeConfig
\": {\n        \"aws:updateSsmAgent\": {\n            \"properties\": [\n
         {\n                \"agentName\": \"amazon-ssm-agent\",\n
  \"source\": \"https://s3.{Region}.amazonaws.com/amazon-ssm-{Region}/ssm-agent-
manifest.json\",\n                \"allowDowngrade\": \"{{ allowDowngrade }}\",\n
                \"targetVersion\": \"{{ version }}\"\n                }\n
    ]\n        }\n    }\n}\n",
            "AssociationVersion": "1"
        }
    ]
}
```

- Para obter detalhes da API, consulte DescribeEffectiveInstanceAssociations na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo descreve as associações efetivas de uma instância.

```
Get-SSMEffectiveInstanceAssociationList -InstanceId "i-0000293ffd8c57862" -
MaxResult 5
```

Saída:

```
AssociationId                        Content
-------------                        -------
d8617c07-2079-4c18-9847-1655fc2698b0 {...
```

Exemplo 2: esse exemplo exibe o conteúdo das associações efetivas de uma instância.

```
(Get-SSMEffectiveInstanceAssociationList -InstanceId "i-0000293ffd8c57862" -
MaxResult 5).Content
```

Saída:

```
{
    "schemaVersion": "1.2",
    "description": "Update the Amazon SSM Agent to the latest version or
 specified version.",
    "parameters": {
        "version": {
            "default": "",
            "description": "(Optional) A specific version of the Amazon SSM Agent
 to install. If not specified, the agen
t will be updated to the latest version.",
            "type": "String"
        },
        "allowDowngrade": {
            "default": "false",
            "description": "(Optional) Allow the Amazon SSM Agent service to be
 downgraded to an earlier version. If set
 to false, the service can be upgraded to newer versions only (default). If set
 to true, specify the earlier version.",
            "type": "String",
            "allowedValues": [
                "true",
                "false"
            ]
        }
    },
```

```
    "runtimeConfig": {
        "aws:updateSsmAgent": {
            "properties": [
                {
                "agentName": "amazon-ssm-agent",
                "source": "https://s3.{Region}.amazonaws.com/amazon-ssm-{Region}/
ssm-agent-manifest.json",
                "allowDowngrade": "{{ allowDowngrade }}",
                "targetVersion": "{{ version }}"
                }
            ]
        }
    }
}
```

- Para obter detalhes da API, consulte DescribeEffectiveInstanceAssociations na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo descreve as associações efetivas de uma instância.

```
Get-SSMEffectiveInstanceAssociationList -InstanceId "i-0000293ffd8c57862" -
MaxResult 5
```

Saída:

```
AssociationId                        Content
-------------                        -------
d8617c07-2079-4c18-9847-1655fc2698b0 {...
```

Exemplo 2: esse exemplo exibe o conteúdo das associações efetivas de uma instância.

```
(Get-SSMEffectiveInstanceAssociationList -InstanceId "i-0000293ffd8c57862" -
MaxResult 5).Content
```

Saída:

```
{
    "schemaVersion": "1.2",
    "description": "Update the Amazon SSM Agent to the latest version or
 specified version.",
```

```
    "parameters": {
        "version": {
            "default": "",
            "description": "(Optional) A specific version of the Amazon SSM Agent
 to install. If not specified, the agen
t will be updated to the latest version.",
            "type": "String"
        },
        "allowDowngrade": {
            "default": "false",
            "description": "(Optional) Allow the Amazon SSM Agent service to be
 downgraded to an earlier version. If set
 to false, the service can be upgraded to newer versions only (default). If set
 to true, specify the earlier version.",
            "type": "String",
            "allowedValues": [
                "true",
                "false"
            ]
        }
    },
    "runtimeConfig": {
        "aws:updateSsmAgent": {
            "properties": [
                {
                "agentName": "amazon-ssm-agent",
                "source": "https://s3.{Region}.amazonaws.com/amazon-ssm-{Region}/
ssm-agent-manifest.json",
                "allowDowngrade": "{{ allowDowngrade }}",
                "targetVersion": "{{ version }}"
                }
            ]
        }
    }
}
```

- Para obter detalhes da API, consulte DescribeEffectiveInstanceAssociations na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeEffectivePatchesForPatchBaseline`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeEffectivePatchesForPatchBaseline`.

CLI

AWS CLI

Exemplo 1: obter todos os patches definidos por uma lista de referência de patches

O exemplo de `describe-effective-patches-for-patch-baseline` a seguir retorna os patches definidos por uma lista de referência de patches personalizada na conta da AWS atual. Observe que, para uma lista de referência personalizada, somente a ID é necessária para `--baseline-id`.

```
aws ssm describe-effective-patches-for-patch-baseline \
    --baseline-id "pb-08b654cf9b9681f04"
```

Saída:

```
{
    "EffectivePatches": [
        {
            "Patch": {
                "Id": "fe6bd8c2-3752-4c8b-ab3e-1a7ed08767ba",
                "ReleaseDate": 1544047205.0,
                "Title": "2018-11 Update for Windows Server 2019 for x64-based
 Systems (KB4470788)",
                "Description": "Install this update to resolve issues in Windows.
 For a complete listing of the issues that are included in this update, see the
 associated Microsoft Knowledge Base article for more information. After you
 install this item, you may have to restart your computer.",
                "ContentUrl": "https://support.microsoft.com/en-us/kb/4470788",
                "Vendor": "Microsoft",
                "ProductFamily": "Windows",
                "Product": "WindowsServer2019",
                "Classification": "SecurityUpdates",
                "MsrcSeverity": "Critical",
                "KbNumber": "KB4470788",
                "MsrcNumber": "",
                "Language": "All"
```

```
            },
            "PatchStatus": {
                "DeploymentStatus": "APPROVED",
                "ComplianceLevel": "CRITICAL",
                "ApprovalDate": 1544047205.0
            }
        },
        {
            "Patch": {
                "Id": "915a6b1a-f556-4d83-8f50-b2e75a9a7e58",
                "ReleaseDate": 1549994400.0,
                "Title": "2019-02 Cumulative Update for .NET Framework 3.5 and
 4.7.2 for Windows Server 2019 for x64 (KB4483452)",
                "Description": "A security issue has been identified in a
 Microsoft software product that could affect your system. You can help protect
 your system by installing this update from Microsoft. For a complete listing
 of the issues that are included in this update, see the associated Microsoft
 Knowledge Base article. After you install this update, you may have to restart
 your system.",
                "ContentUrl": "https://support.microsoft.com/en-us/kb/4483452",
                "Vendor": "Microsoft",
                "ProductFamily": "Windows",
                "Product": "WindowsServer2019",
                "Classification": "SecurityUpdates",
                "MsrcSeverity": "Important",
                "KbNumber": "KB4483452",
                "MsrcNumber": "",
                "Language": "All"
            },
            "PatchStatus": {
                "DeploymentStatus": "APPROVED",
                "ComplianceLevel": "CRITICAL",
                "ApprovalDate": 1549994400.0
            }
        },
        ...
    ],
    "NextToken": "--token string truncated--"
}
```

Exemplo 2: obter todos os patches definidos por uma lista de referência de patches gerenciada pela AWS

O exemplo de `describe-effective-patches-for-patch-baseline` a seguir retorna os patches definidos por uma lista de referência de patches gerenciada pela AWS. Observe que, para uma lista de referência gerenciada pela AWS, o ARN completo da lista de referência é necessário para `--baseline-id`

```
aws ssm describe-effective-patches-for-patch-baseline \
    --baseline-id "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-020d361a05defe4ed"
```

Consulte a amostra de saída no exemplo 1.

Para obter mais informações, consulte Como os patches de segurança são selecionados no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribeEffectivePatchesForPatchBaseline na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo mostra todas as listas de referência de patches, com uma lista de resultados máxima de 1.

```
Get-SSMEffectivePatchesForPatchBaseline -BaselineId "pb-0a2f1059b670ebd31" -
MaxResult 1
```

Saída:

```
Patch                                       PatchStatus
-----                                       -----------
Amazon.SimpleSystemsManagement.Model.Patch
 Amazon.SimpleSystemsManagement.Model.PatchStatus
```

Exemplo 2: esse exemplo mostra o status do patch para todas as listas de referência de patches, com uma lista de resultados máxima de 1.

```
(Get-SSMEffectivePatchesForPatchBaseline -BaselineId "pb-0a2f1059b670ebd31" -
MaxResult 1).PatchStatus
```

Saída:

```
ApprovalDate          DeploymentStatus
------------          ----------------
12/21/2010 6:00:00 PM APPROVED
```

- Para obter detalhes da API, consulte DescribeEffectivePatchesForPatchBaseline na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo mostra todas as listas de referência de patches, com uma lista de resultados máxima de 1.

```
Get-SSMEffectivePatchesForPatchBaseline -BaselineId "pb-0a2f1059b670ebd31" -
MaxResult 1
```

Saída:

```
Patch                                     PatchStatus
-----                                     -----------
Amazon.SimpleSystemsManagement.Model.Patch
 Amazon.SimpleSystemsManagement.Model.PatchStatus
```

Exemplo 2: esse exemplo mostra o status do patch para todas as listas de referência de patches, com uma lista de resultados máxima de 1.

```
(Get-SSMEffectivePatchesForPatchBaseline -BaselineId "pb-0a2f1059b670ebd31" -
MaxResult 1).PatchStatus
```

Saída:

```
ApprovalDate          DeploymentStatus
------------          ----------------
12/21/2010 6:00:00 PM APPROVED
```

- Para obter detalhes da API, consulte DescribeEffectivePatchesForPatchBaseline na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeInstanceAssociationsStatus`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeInstanceAssociationsStatus`.

CLI

AWS CLI

Para descrever o status das associações de uma instância

Este exemplo mostra detalhes das associações de uma instância.

Comando:

```
aws ssm describe-instance-associations-status --instance-id "i-1234567890abcdef0"
```

Saída:

```
{
  "InstanceAssociationStatusInfos": [
      {
          "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
          "Name": "AWS-GatherSoftwareInventory",
          "DocumentVersion": "1",
          "AssociationVersion": "1",
          "InstanceId": "i-1234567890abcdef0",
          "ExecutionDate": 1550501886.0,
          "Status": "Success",
          "ExecutionSummary": "1 out of 1 plugin processed, 1 success, 0 failed,
 0 timedout, 0 skipped. ",
          "AssociationName": "Inventory-Association"
      },
      {
          "AssociationId": "5c5a31f6-6dae-46f9-944c-0123456789ab",
          "Name": "AWS-UpdateSSMAgent",
          "DocumentVersion": "1",
          "AssociationVersion": "1",
```

```
            "InstanceId": "i-1234567890abcdef0",
            "ExecutionDate": 1550505828.548,
            "Status": "Success",
            "DetailedStatus": "Success",
            "AssociationName": "UpdateSSMAgent"
        }
    ]
}
```

- Para obter detalhes da API, consulte DescribeInstanceAssociationsStatus na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo mostra detalhes das associações de uma instância.

```
Get-SSMInstanceAssociationsStatus -InstanceId "i-0000293ffd8c57862"
```

Saída:

```
AssociationId    : d8617c07-2079-4c18-9847-1655fc2698b0
DetailedStatus   : Pending
DocumentVersion  : 1
ErrorCode        :
ExecutionDate    : 2/20/2015 8:31:11 AM
ExecutionSummary : temp_status_change
InstanceId       : i-0000293ffd8c57862
Name             : AWS-UpdateSSMAgent
OutputUrl        :
Status           : Pending
```

Exemplo 2: esse exemplo verifica o status da associação da instância para o ID da instância fornecido e exibe o status de execução dessas associações

```
Get-SSMInstanceAssociationsStatus -InstanceId i-012e3cb4df567e8aa | ForEach-
Object {Get-SSMAssociationExecution -AssociationId .AssociationId}
```

Saída:

```
AssociationId         : 512a34a5-c678-1234-1234-12345678db9e
AssociationVersion    : 2
CreatedTime           : 3/2/2019 8:53:29 AM
DetailedStatus        :
ExecutionId           : 512a34a5-c678-1234-1234-12345678db9e
LastExecutionDate     : 1/1/0001 12:00:00 AM
ResourceCountByStatus : {Success=9}
Status                : Success
```

- Para obter detalhes da API, consulte DescribeInstanceAssociationsStatus na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo mostra detalhes das associações de uma instância.

```
Get-SSMInstanceAssociationsStatus -InstanceId "i-0000293ffd8c57862"
```

Saída:

```
AssociationId    : d8617c07-2079-4c18-9847-1655fc2698b0
DetailedStatus   : Pending
DocumentVersion  : 1
ErrorCode        :
ExecutionDate    : 2/20/2015 8:31:11 AM
ExecutionSummary : temp_status_change
InstanceId       : i-0000293ffd8c57862
Name             : AWS-UpdateSSMAgent
OutputUrl        :
Status           : Pending
```

Exemplo 2: esse exemplo verifica o status da associação da instância para o ID da instância fornecido e exibe o status de execução dessas associações

```
Get-SSMInstanceAssociationsStatus -InstanceId i-012e3cb4df567e8aa | ForEach-
Object {Get-SSMAssociationExecution -AssociationId .AssociationId}
```

Saída:

```
AssociationId         : 512a34a5-c678-1234-1234-12345678db9e
```

```
AssociationVersion    : 2
CreatedTime           : 3/2/2019 8:53:29 AM
DetailedStatus        :
ExecutionId           : 512a34a5-c678-1234-1234-12345678db9e
LastExecutionDate     : 1/1/0001 12:00:00 AM
ResourceCountByStatus : {Success=9}
Status                : Success
```

- Para obter detalhes da API, consulte DescribeInstanceAssociationsStatus na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeInstanceInformation`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeInstanceInformation`.

CLI

AWS CLI

Exemplo 1: descrever as informações da instância gerenciada

O exemplo de `describe-instance-information` a seguir recupera detalhes de cada uma das suas instâncias gerenciadas.

```
aws ssm describe-instance-information
```

Exemplo 2: descrever informações sobre uma instância gerenciada específica

O exemplo de `describe-instance-information` a seguir mostra os detalhes da instância gerenciada `i-028ea792daEXAMPLE`.

```
aws ssm describe-instance-information \
    --filters "Key=InstanceIds,Values=i-028ea792daEXAMPLE"
```

Exemplo 3: descrever informações sobre instâncias gerenciadas com uma chave de tag específica

O exemplo de `describe-instance-information` a seguir mostra detalhes de instâncias gerenciadas que têm a chave de tag `DEV`.

```
aws ssm describe-instance-information \
    --filters "Key=tag-key,Values=DEV"
```

Saída:

```
{
    "InstanceInformationList": [
        {
            "InstanceId": "i-028ea792daEXAMPLE",
            "PingStatus": "Online",
            "LastPingDateTime": 1582221233.421,
            "AgentVersion": "2.3.842.0",
            "IsLatestVersion": true,
            "PlatformType": "Linux",
            "PlatformName": "SLES",
            "PlatformVersion": "15.1",
            "ResourceType": "EC2Instance",
            "IPAddress": "192.0.2.0",
            "ComputerName": "ip-198.51.100.0.us-east-2.compute.internal",
            "AssociationStatus": "Success",
            "LastAssociationExecutionDate": 1582220806.0,
            "LastSuccessfulAssociationExecutionDate": 1582220806.0,
            "AssociationOverview": {
                "DetailedStatus": "Success",
                "InstanceAssociationStatusAggregatedCount": {
                    "Success": 2
                }
            }
        }
    ]
}
```

Para obter mais informações, consulte Instâncias gerenciadas no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribeInstanceInformation na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo mostra detalhes de cada uma de suas instâncias.

```
Get-SSMInstanceInformation
```

Saída:

```
ActivationId                          :
AgentVersion                          : 2.0.672.0
AssociationOverview                   :
 Amazon.SimpleSystemsManagement.Model.InstanceAggregatedAssociationOverview
AssociationStatus                     : Success
ComputerName                          : ip-172-31-44-222.us-
west-2.compute.internal
IamRole                               :
InstanceId                            : i-0cb2b964d3e14fd9f
IPAddress                             : 172.31.44.222
IsLatestVersion                       : True
LastAssociationExecutionDate          : 2/24/2017 3:18:09 AM
LastPingDateTime                      : 2/24/2017 3:35:03 AM
LastSuccessfulAssociationExecutionDate : 2/24/2017 3:18:09 AM
Name                                  :
PingStatus                            : ConnectionLost
PlatformName                          : Amazon Linux AMI
PlatformType                          : Linux
PlatformVersion                       : 2016.09
RegistrationDate                      : 1/1/0001 12:00:00 AM
ResourceType                          : EC2Instance
```

Exemplo 2: esse exemplo mostra como usar o parâmetro -Filter para filtrar os resultados para mostrar somente as instâncias do AWS Systems Manager na região **`us-east-1`** com **`AgentVersion`** igual a **`2.2.800.0`**. É possível encontrar uma lista de valores da chave -Filter válidos no tópico de referência da API InstanceInformation (https://docs.aws.amazon.com/systems-manager/latest/APIReference/API_InstanceInformation.html#systemsmanager-Type-InstanceInformation-ActivationId).

```
$Filters = @{
    Key="AgentVersion"
```

```
    Values="2.2.800.0"
}
Get-SSMInstanceInformation -Region us-east-1 -Filter $Filters
```

Saída:

```
ActivationId                          :
AgentVersion                          : 2.2.800.0
AssociationOverview                   :
 Amazon.SimpleSystemsManagement.Model.InstanceAggregatedAssociationOverview
AssociationStatus                     : Success
ComputerName                          : EXAMPLE-EXAMPLE.WORKGROUP
IamRole                               :
InstanceId                            : i-EXAMPLEb0792d98ce
IPAddress                             : 10.0.0.01
IsLatestVersion                       : False
LastAssociationExecutionDate          : 8/16/2018 12:02:50 AM
LastPingDateTime                      : 8/16/2018 7:40:27 PM
LastSuccessfulAssociationExecutionDate : 8/16/2018 12:02:50 AM
Name                                  :
PingStatus                            : Online
PlatformName                          : Microsoft Windows Server 2016 Datacenter
PlatformType                          : Windows
PlatformVersion                       : 10.0.14393
RegistrationDate                      : 1/1/0001 12:00:00 AM
ResourceType                          : EC2Instance

ActivationId                          :
AgentVersion                          : 2.2.800.0
AssociationOverview                   :
 Amazon.SimpleSystemsManagement.Model.InstanceAggregatedAssociationOverview
AssociationStatus                     : Success
ComputerName                          : EXAMPLE-EXAMPLE.WORKGROUP
IamRole                               :
InstanceId                            : i-EXAMPLEac7501d023
IPAddress                             : 10.0.0.02
IsLatestVersion                       : False
LastAssociationExecutionDate          : 8/16/2018 12:00:20 AM
LastPingDateTime                      : 8/16/2018 7:40:35 PM
LastSuccessfulAssociationExecutionDate : 8/16/2018 12:00:20 AM
Name                                  :
PingStatus                            : Online
PlatformName                          : Microsoft Windows Server 2016 Datacenter
```

```
PlatformType                        : Windows
PlatformVersion                     : 10.0.14393
RegistrationDate                    : 1/1/0001 12:00:00 AM
ResourceType                        : EC2Instance
```

Exemplo 3: esse exemplo mostra como usar o parâmetro -InstanceInformationFilterList para filtrar os resultados para mostrar somente as instâncias do AWS Systems Manager na região **`us-east-1`** com **`PlatformTypes`** igual a **`Windows`** ou **`Linux`**. É possível encontrar uma lista de valores da chave -InstanceInformationFilterList válidos no tópico de referência da API InstanceInformationFilter (https://docs.aws.amazon.com/systems-manager/latest/APIReference/API_InstanceInformationFilter.html).

```
$Filters = @{
   Key="PlatformTypes"
   ValueSet=("Windows","Linux")
}
Get-SSMInstanceInformation -Region us-east-1 -InstanceInformationFilterList
 $Filters
```

Saída:

```
ActivationId                        :
AgentVersion                        : 2.2.800.0
AssociationOverview                 :
 Amazon.SimpleSystemsManagement.Model.InstanceAggregatedAssociationOverview
AssociationStatus                   : Success
ComputerName                        : EXAMPLE-EXAMPLE.WORKGROUP
IamRole                             :
InstanceId                          : i-EXAMPLEb0792d98ce
IPAddress                           : 10.0.0.27
IsLatestVersion                     : False
LastAssociationExecutionDate        : 8/16/2018 12:02:50 AM
LastPingDateTime                    : 8/16/2018 7:40:27 PM
LastSuccessfulAssociationExecutionDate : 8/16/2018 12:02:50 AM
Name                                :
PingStatus                          : Online
PlatformName                        : Ubuntu Server 18.04 LTS
PlatformType                        : Linux
PlatformVersion                     : 18.04
RegistrationDate                    : 1/1/0001 12:00:00 AM
ResourceType                        : EC2Instance
```

```
ActivationId                          :
AgentVersion                          : 2.2.800.0
AssociationOverview                   :
 Amazon.SimpleSystemsManagement.Model.InstanceAggregatedAssociationOverview
AssociationStatus                     : Success
ComputerName                          : EXAMPLE-EXAMPLE.WORKGROUP
IamRole                               :
InstanceId                            : i-EXAMPLEac7501d023
IPAddress                             : 10.0.0.100
IsLatestVersion                       : False
LastAssociationExecutionDate          : 8/16/2018 12:00:20 AM
LastPingDateTime                      : 8/16/2018 7:40:35 PM
LastSuccessfulAssociationExecutionDate : 8/16/2018 12:00:20 AM
Name                                  :
PingStatus                            : Online
PlatformName                          : Microsoft Windows Server 2016 Datacenter
PlatformType                          : Windows
PlatformVersion                       : 10.0.14393
RegistrationDate                      : 1/1/0001 12:00:00 AM
ResourceType                          : EC2Instance
```

Exemplo 4: esse exemplo lista instâncias gerenciadas pelo SSM e exporta InstanceID, PingStatus, LastPingDateTime e PlatformName para um arquivo csv.

```
Get-SSMInstanceInformation | Select-Object InstanceId, PingStatus,
 LastPingDateTime, PlatformName | Export-Csv Instance-details.csv -
NoTypeInformation
```

- Para obter detalhes da API, consulte DescribeInstanceInformation na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo mostra detalhes de cada uma de suas instâncias.

```
Get-SSMInstanceInformation
```

Saída:

```
ActivationId                          :
AgentVersion                          : 2.0.672.0
AssociationOverview                   :
 Amazon.SimpleSystemsManagement.Model.InstanceAggregatedAssociationOverview
```

```
AssociationStatus                     : Success
ComputerName                          : ip-172-31-44-222.us-
west-2.compute.internal
IamRole                               :
InstanceId                            : i-0cb2b964d3e14fd9f
IPAddress                             : 172.31.44.222
IsLatestVersion                       : True
LastAssociationExecutionDate          : 2/24/2017 3:18:09 AM
LastPingDateTime                      : 2/24/2017 3:35:03 AM
LastSuccessfulAssociationExecutionDate : 2/24/2017 3:18:09 AM
Name                                  :
PingStatus                            : ConnectionLost
PlatformName                          : Amazon Linux AMI
PlatformType                          : Linux
PlatformVersion                       : 2016.09
RegistrationDate                      : 1/1/0001 12:00:00 AM
ResourceType                          : EC2Instance
```

Exemplo 2: esse exemplo mostra como usar o parâmetro -Filter para filtrar os resultados para mostrar somente as instâncias do AWS Systems Manager na região **`us-east-1`** com **`AgentVersion`** igual a **`2.2.800.0`**. É possível encontrar uma lista de valores da chave -Filter válidos no tópico de referência da API InstanceInformation (https://docs.aws.amazon.com/systems-manager/latest/APIReference/API_InstanceInformation.html#systemsmanager-Type-InstanceInformation-ActivationId).

```
$Filters = @{
    Key="AgentVersion"
    Values="2.2.800.0"
}
Get-SSMInstanceInformation -Region us-east-1 -Filter $Filters
```

Saída:

```
ActivationId                          :
AgentVersion                          : 2.2.800.0
AssociationOverview                   :
 Amazon.SimpleSystemsManagement.Model.InstanceAggregatedAssociationOverview
AssociationStatus                     : Success
ComputerName                          : EXAMPLE-EXAMPLE.WORKGROUP
IamRole                               :
InstanceId                            : i-EXAMPLEb0792d98ce
IPAddress                             : 10.0.0.01
```

```
IsLatestVersion                          : False
LastAssociationExecutionDate             : 8/16/2018 12:02:50 AM
LastPingDateTime                         : 8/16/2018 7:40:27 PM
LastSuccessfulAssociationExecutionDate : 8/16/2018 12:02:50 AM
Name                                     :
PingStatus                               : Online
PlatformName                             : Microsoft Windows Server 2016 Datacenter
PlatformType                             : Windows
PlatformVersion                          : 10.0.14393
RegistrationDate                         : 1/1/0001 12:00:00 AM
ResourceType                             : EC2Instance

ActivationId                             :
AgentVersion                             : 2.2.800.0
AssociationOverview                      :
 Amazon.SimpleSystemsManagement.Model.InstanceAggregatedAssociationOverview
AssociationStatus                        : Success
ComputerName                             : EXAMPLE-EXAMPLE.WORKGROUP
IamRole                                  :
InstanceId                               : i-EXAMPLEac7501d023
IPAddress                                : 10.0.0.02
IsLatestVersion                          : False
LastAssociationExecutionDate             : 8/16/2018 12:00:20 AM
LastPingDateTime                         : 8/16/2018 7:40:35 PM
LastSuccessfulAssociationExecutionDate : 8/16/2018 12:00:20 AM
Name                                     :
PingStatus                               : Online
PlatformName                             : Microsoft Windows Server 2016 Datacenter
PlatformType                             : Windows
PlatformVersion                          : 10.0.14393
RegistrationDate                         : 1/1/0001 12:00:00 AM
ResourceType                             : EC2Instance
```

Exemplo 3: esse exemplo mostra como usar o parâmetro -InstanceInformationFilterList para filtrar os resultados para mostrar somente as instâncias do AWS Systems Manager na região **`us-east-1`** com **`PlatformTypes`** igual a **`Windows`** ou **`Linux`**. É possível encontrar uma lista de valores da chave -InstanceInformationFilterList válidos no tópico de referência da API InstanceInformationFilter (https://docs.aws.amazon.com/systems-manager/latest/APIReference/API_InstanceInformationFilter.html).

```
$Filters = @{
   Key="PlatformTypes"
```

```
   ValueSet=("Windows","Linux")
}
Get-SSMInstanceInformation -Region us-east-1 -InstanceInformationFilterList
 $Filters
```

Saída:

```
ActivationId                          :
AgentVersion                          : 2.2.800.0
AssociationOverview                   :
 Amazon.SimpleSystemsManagement.Model.InstanceAggregatedAssociationOverview
AssociationStatus                     : Success
ComputerName                          : EXAMPLE-EXAMPLE.WORKGROUP
IamRole                               :
InstanceId                            : i-EXAMPLEb0792d98ce
IPAddress                             : 10.0.0.27
IsLatestVersion                       : False
LastAssociationExecutionDate          : 8/16/2018 12:02:50 AM
LastPingDateTime                      : 8/16/2018 7:40:27 PM
LastSuccessfulAssociationExecutionDate : 8/16/2018 12:02:50 AM
Name                                  :
PingStatus                            : Online
PlatformName                          : Ubuntu Server 18.04 LTS
PlatformType                          : Linux
PlatformVersion                       : 18.04
RegistrationDate                      : 1/1/0001 12:00:00 AM
ResourceType                          : EC2Instance

ActivationId                          :
AgentVersion                          : 2.2.800.0
AssociationOverview                   :
 Amazon.SimpleSystemsManagement.Model.InstanceAggregatedAssociationOverview
AssociationStatus                     : Success
ComputerName                          : EXAMPLE-EXAMPLE.WORKGROUP
IamRole                               :
InstanceId                            : i-EXAMPLEac7501d023
IPAddress                             : 10.0.0.100
IsLatestVersion                       : False
LastAssociationExecutionDate          : 8/16/2018 12:00:20 AM
LastPingDateTime                      : 8/16/2018 7:40:35 PM
LastSuccessfulAssociationExecutionDate : 8/16/2018 12:00:20 AM
Name                                  :
PingStatus                            : Online
```

```
PlatformName                    : Microsoft Windows Server 2016 Datacenter
PlatformType                    : Windows
PlatformVersion                 : 10.0.14393
RegistrationDate                : 1/1/0001 12:00:00 AM
ResourceType                    : EC2Instance
```

Exemplo 4: esse exemplo lista instâncias gerenciadas pelo SSM e exporta InstanceID, PingStatus, LastPingDateTime e PlatformName para um arquivo csv.

```
Get-SSMInstanceInformation | Select-Object InstanceId, PingStatus,
 LastPingDateTime, PlatformName | Export-Csv Instance-details.csv -
NoTypeInformation
```

- Para obter detalhes da API, consulte DescribeInstanceInformation na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeInstancePatchStates`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeInstancePatchStates`.

CLI

AWS CLI

Para obter os estados resumidos de patches para instâncias

Este exemplo de `describe-instance-patch-states` obtém os estados resumidos de patches para uma instância.

```
aws ssm describe-instance-patch-states \
    --instance-ids "i-1234567890abcdef0"
```

Saída:

```
{
    "InstancePatchStates": [
```

```
        {
            "InstanceId": "i-1234567890abcdef0",
            "PatchGroup": "my-patch-group",
            "BaselineId": "pb-0713accee01234567",
            "SnapshotId": "521c3536-930c-4aa9-950e-01234567abcd",
            "CriticalNonCompliantCount": 2,
            "SecurityNonCompliantCount": 2,
            "OtherNonCompliantCount": 1,
            "InstalledCount": 123,
            "InstalledOtherCount": 334,
            "InstalledPendingRebootCount": 0,
            "InstalledRejectedCount": 0,
            "MissingCount": 1,
            "FailedCount": 2,
            "UnreportedNotApplicableCount": 11,
            "NotApplicableCount": 2063,
            "OperationStartTime": "2021-05-03T11:00:56-07:00",
            "OperationEndTime": "2021-05-03T11:01:09-07:00",
            "Operation": "Scan",
            "LastNoRebootInstallOperationTime": "2020-06-14T12:17:41-07:00",
            "RebootOption": "RebootIfNeeded"
        }
    ]
}
```

Para obter mais informações, consulte Sobre a conformidade de patches no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribeInstancePatchStates na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo obtém os estados resumidos de patches para uma instância.

```
Get-SSMInstancePatchState -InstanceId "i-08ee91c0b17045407"
```

Exemplo 2: esse exemplo obtém os estados resumidos de patches para duas instâncias.

```
Get-SSMInstancePatchState -InstanceId "i-08ee91c0b17045407","i-09a618aec652973a9"
```

- Para obter detalhes da API, consulte DescribeInstancePatchStates na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo obtém os estados resumidos de patches para uma instância.

```
Get-SSMInstancePatchState -InstanceId "i-08ee91c0b17045407"
```

Exemplo 2: esse exemplo obtém os estados resumidos de patches para duas instâncias.

```
Get-SSMInstancePatchState -InstanceId "i-08ee91c0b17045407","i-09a618aec652973a9"
```

- Para obter detalhes da API, consulte DescribeInstancePatchStates na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeInstancePatchStatesForPatchGroup`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeInstancePatchStatesForPatchGroup`.

CLI

AWS CLI

Exemplo 1: obter os estados da instância de um grupo de patches

O exemplo de `describe-instance-patch-states-for-patch-group` a seguir recupera detalhes sobre os estados resumidos de patches por instância para o grupo de patches especificado.

```
aws ssm describe-instance-patch-states-for-patch-group \
    --patch-group "Production"
```

Saída:

```
{
```

```
    "InstancePatchStates": [
        {
            "InstanceId": "i-02573cafcfEXAMPLE",
            "PatchGroup": "Production",
            "BaselineId": "pb-0c10e65780EXAMPLE",
            "SnapshotId": "a3f5ff34-9bc4-4d2c-a665-4d1c1EXAMPLE",
            "OwnerInformation": "",
            "InstalledCount": 32,
            "InstalledOtherCount": 1,
            "InstalledPendingRebootCount": 0,
            "InstalledRejectedCount": 0,
            "MissingCount": 2,
            "FailedCount": 0,
            "UnreportedNotApplicableCount": 2671,
            "NotApplicableCount": 400,
            "OperationStartTime": "2021-08-04T11:03:50.590000-07:00",
            "OperationEndTime": "2021-08-04T11:04:21.555000-07:00",
            "Operation": "Scan",
            "RebootOption": "NoReboot",
            "CriticalNonCompliantCount": 0,
            "SecurityNonCompliantCount": 1,
            "OtherNonCompliantCount": 0
        },
        {
            "InstanceId": "i-0471e04240EXAMPLE",
            "PatchGroup": "Production",
            "BaselineId": "pb-09ca3fb51fEXAMPLE",
            "SnapshotId": "05d8ffb0-1bbe-4812-ba2d-d9b7bEXAMPLE",
            "OwnerInformation": "",
            "InstalledCount": 32,
            "InstalledOtherCount": 1,
            "InstalledPendingRebootCount": 0,
            "InstalledRejectedCount": 0,
            "MissingCount": 2,
            "FailedCount": 0,
            "UnreportedNotApplicableCount": 2671,
            "NotApplicableCount": 400,
            "OperationStartTime": "2021-08-04T22:06:20.340000-07:00",
            "OperationEndTime": "2021-08-04T22:07:11.220000-07:00",
            "Operation": "Scan",
            "RebootOption": "NoReboot",
            "CriticalNonCompliantCount": 0,
            "SecurityNonCompliantCount": 1,
            "OtherNonCompliantCount": 0
```

```
        }
    ]
}
```

Exemplo 2: obter os estados da instância de um grupo de patches com mais de cinco patches ausentes

O exemplo de `describe-instance-patch-states-for-patch-group` a seguir recupera detalhes sobre os estados resumidos de patches para o grupo de patches especificado por instâncias com mais de cinco patches ausentes.

```
aws ssm describe-instance-patch-states-for-patch-group \
    --filters Key=MissingCount,Type=GreaterThan,Values=5 \
    --patch-group "Production"
```

Saída:

```
{
    "InstancePatchStates": [
        {
            "InstanceId": "i-02573cafcfEXAMPLE",
            "PatchGroup": "Production",
            "BaselineId": "pb-0c10e65780EXAMPLE",
            "SnapshotId": "a3f5ff34-9bc4-4d2c-a665-4d1c1EXAMPLE",
            "OwnerInformation": "",
            "InstalledCount": 46,
            "InstalledOtherCount": 4,
            "InstalledPendingRebootCount": 1,
            "InstalledRejectedCount": 1,
            "MissingCount": 7,
            "FailedCount": 0,
            "UnreportedNotApplicableCount": 232,
            "NotApplicableCount": 654,
            "OperationStartTime": "2021-08-04T11:03:50.590000-07:00",
            "OperationEndTime": "2021-08-04T11:04:21.555000-07:00",
            "Operation": "Scan",
            "RebootOption": "NoReboot",
            "CriticalNonCompliantCount": 0,
            "SecurityNonCompliantCount": 1,
            "OtherNonCompliantCount": 1
        }
    ]
```

```
}
```

Exemplo 3: obter os estados da instância de um grupo de patches com menos de dez instâncias que exigem uma reinicialização

O exemplo de `describe-instance-patch-states-for-patch-group` a seguir recupera detalhes sobre os estados resumidos de patches para o grupo de patches especificado por instâncias com menos de dez instâncias que exigem uma reinicialização.

```
aws ssm describe-instance-patch-states-for-patch-group \
    --filters Key=InstalledPendingRebootCount,Type=LessThan,Values=10 \
    --patch-group "Production"
```

Saída:

```
{
    "InstancePatchStates": [
        {
            "InstanceId": "i-02573cafcfEXAMPLE",
            "BaselineId": "pb-0c10e65780EXAMPLE",
            "SnapshotId": "a3f5ff34-9bc4-4d2c-a665-4d1c1EXAMPLE",
            "PatchGroup": "Production",
            "OwnerInformation": "",
            "InstalledCount": 32,
            "InstalledOtherCount": 1,
            "InstalledPendingRebootCount": 4,
            "InstalledRejectedCount": 0,
            "MissingCount": 2,
            "FailedCount": 0,
            "UnreportedNotApplicableCount": 846,
            "NotApplicableCount": 212,
            "OperationStartTime": "2021-08-046T11:03:50.590000-07:00",
            "OperationEndTime": "2021-08-06T11:04:21.555000-07:00",
            "Operation": "Scan",
            "RebootOption": "NoReboot",
            "CriticalNonCompliantCount": 0,
            "SecurityNonCompliantCount": 1,
            "OtherNonCompliantCount": 0
        }
    ]
}
```

Para obter mais informações, consulte Noções básicas sobre valores de destado de conformidade de patches no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribeInstancePatchStatesForPatchGroup na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo obtém os estados de resumo de patches por instância de um grupo de patches.

```
Get-SSMInstancePatchStatesForPatchGroup -PatchGroup "Production"
```

- Para obter detalhes da API, consulte DescribeInstancePatchStatesForPatchGroup na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo obtém os estados de resumo de patches por instância de um grupo de patches.

```
Get-SSMInstancePatchStatesForPatchGroup -PatchGroup "Production"
```

- Para obter detalhes da API, consulte DescribeInstancePatchStatesForPatchGroup na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeInstancePatches`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeInstancePatches`.

CLI

AWS CLI

Exemplo 1: obter os detalhes do estado do patch para uma instância

O exemplo de `describe-instance-patches` a seguir recupera detalhes sobre os patches para a instância especificada.

```
aws ssm describe-instance-patches \
    --instance-id "i-1234567890abcdef0"
```

Saída:

```
{
    "Patches": [
        {
            "Title": "2019-01 Security Update for Adobe Flash Player for Windows
 Server 2016 for x64-based Systems (KB4480979)",
            "KBId": "KB4480979",
            "Classification": "SecurityUpdates",
            "Severity": "Critical",
            "State": "Installed",
            "InstalledTime": "2019-01-09T00:00:00+00:00"
        },
        {
            "Title": "",
            "KBId": "KB4481031",
            "Classification": "",
            "Severity": "",
            "State": "InstalledOther",
            "InstalledTime": "2019-02-08T00:00:00+00:00"
        },
        ...
    ],
    "NextToken": "--token string truncated--"
}
```

Exemplo 2: obter uma lista de patches no estado Ausente para uma instância

O exemplo de `describe-instance-patches` a seguir recupera informações sobre patches que estão no estado Ausente para a instância especificada.

```
aws ssm describe-instance-patches \
    --instance-id "i-1234567890abcdef0" \
    --filters Key=State,Values=Missing
```

Saída:

```
{
    "Patches": [
        {
            "Title": "Windows Malicious Software Removal Tool x64 - February 2019
 (KB890830)",
            "KBId": "KB890830",
            "Classification": "UpdateRollups",
            "Severity": "Unspecified",
            "State": "Missing",
            "InstalledTime": "1970-01-01T00:00:00+00:00"
        },
        ...
    ],
    "NextToken": "--token string truncated--"
}
```

Para obter mais informações, consulte Sobre estados de conformidade de patches no Guia do usuário do AWS Systems Manager.

Exemplo 3: obter uma lista de patches instalados desde um horário de instalação especificado para uma instância

O exemplo de `describe-instance-patches` a seguir recupera informações sobre patches instalados desde um horário especificado para a instância especificada combinando o uso de `--filters` e `--query`.

```
aws ssm describe-instance-patches \
    --instance-id "i-1234567890abcdef0" \
    --filters Key=State,Values=Installed \
    --query "Patches[?InstalledTime >= `2023-01-01T16:00:00`]"
```

Saída:

```
{
    "Patches": [
        {
            "Title": "2023-03 Cumulative Update for Windows Server 2019 (1809)
 for x64-based Systems (KB5023702)",
            "KBId": "KB5023702",
            "Classification": "SecurityUpdates",
            "Severity": "Critical",
```

```
            "State": "Installed",
            "InstalledTime": "2023-03-16T11:00:00+00:00"
        },
        ...
    ],
    "NextToken": "--token string truncated--"
}
```

- Para obter detalhes da API, consulte DescribeInstancePatches na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo obtém os detalhes de conformidade do patch para uma instância.

```
Get-SSMInstancePatch -InstanceId "i-08ee91c0b17045407"
```

- Para obter detalhes da API, consulte DescribeInstancePatches na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo obtém os detalhes de conformidade do patch para uma instância.

```
Get-SSMInstancePatch -InstanceId "i-08ee91c0b17045407"
```

- Para obter detalhes da API, consulte DescribeInstancePatches na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeMaintenanceWindowExecutionTaskInvocations`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeMaintenanceWindowExecutionTaskInvocations`.

CLI

AWS CLI

Para fazer com que as invocações da tarefa específica sejam realizadas para a execução de tarefa de uma janela de manutenção

O exemplo de `describe-maintenance-window-execution-task-invocations` a seguir lista as invocações para a tarefa especificada executada como parte da execução da janela de manutenção especificada.

```
aws ssm describe-maintenance-window-execution-task-invocations \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2a638355" \
    --task-id "ac0c6ae1-daa3-4a89-832e-d384503b6586"
```

Saída:

```
{
    "WindowExecutionTaskInvocationIdentities": [
        {
            "Status": "SUCCESS",
            "Parameters": "{\"documentName\":\"AWS-RunShellScript\",
\"instanceIds\":[\"i-0000293ffd8c57862\"],\"parameters\":{\"commands\":[\"df\"]},
\"maxConcurrency\":\"1\",\"maxErrors\":\"1\"}",
            "InvocationId": "e274b6e1-fe56-4e32-bd2a-8073c6381d8b",
            "StartTime": 1487692834.723,
            "EndTime": 1487692834.871,
            "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2a638355",
            "TaskExecutionId": "ac0c6ae1-daa3-4a89-832e-d384503b6586"
        }
    ]
}
```

Para obter mais informações, consulte Visualizar informações sobre tarefas e execuções de tarefas (AWS CLI) no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribeMaintenanceWindowExecutionTaskInvocations na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo lista as invocações de uma tarefa executada como parte da execução de uma janela de manutenção.

```
Get-SSMMaintenanceWindowExecutionTaskInvocationList -TaskId "ac0c6ae1-
daa3-4a89-832e-d384503b6586" -WindowExecutionId "518d5565-5969-4cca-8f0e-
da3b2a638355"
```

Saída:

```
EndTime           : 2/21/2017 4:00:34 PM
ExecutionId       :
InvocationId      : e274b6e1-fe56-4e32-bd2a-8073c6381d8b
OwnerInformation  :
Parameters        : {"documentName":"AWS-RunShellScript","instanceIds":
["i-0000293ffd8c57862"],"parameters":{"commands":["df"]},"maxConcurrency":"1",
                    "maxErrors":"1"}
StartTime         : 2/21/2017 4:00:34 PM
Status            : FAILED
StatusDetails     : The instance IDs list contains an invalid entry.
TaskExecutionId   : ac0c6ae1-daa3-4a89-832e-d384503b6586
WindowExecutionId : 518d5565-5969-4cca-8f0e-da3b2a638355
WindowTargetId    :
```

- Para obter detalhes da API, consulte DescribeMaintenanceWindowExecutionTaskInvocations na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo lista as invocações de uma tarefa executada como parte da execução de uma janela de manutenção.

```
Get-SSMMaintenanceWindowExecutionTaskInvocationList -TaskId "ac0c6ae1-
daa3-4a89-832e-d384503b6586" -WindowExecutionId "518d5565-5969-4cca-8f0e-
da3b2a638355"
```

Saída:

```
EndTime           : 2/21/2017 4:00:34 PM
ExecutionId       :
InvocationId      : e274b6e1-fe56-4e32-bd2a-8073c6381d8b
OwnerInformation  :
Parameters        : {"documentName":"AWS-RunShellScript","instanceIds":
["i-0000293ffd8c57862"],"parameters":{"commands":["df"]},"maxConcurrency":"1",
                    "maxErrors":"1"}
StartTime         : 2/21/2017 4:00:34 PM
Status            : FAILED
StatusDetails     : The instance IDs list contains an invalid entry.
TaskExecutionId   : ac0c6ae1-daa3-4a89-832e-d384503b6586
WindowExecutionId : 518d5565-5969-4cca-8f0e-da3b2a638355
WindowTargetId    :
```

- Para obter detalhes da API, consulte DescribeMaintenanceWindowExecutionTaskInvocations na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeMaintenanceWindowExecutionTasks`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeMaintenanceWindowExecutionTasks`.

CLI

AWS CLI

Para listar todas as tarefas associadas à execução de uma janela de manutenção

O exemplo de `ssm describe-maintenance-window-execution-tasks` a seguir lista as tarefas associadas à execução da janela de manutenção especificada.

```
aws ssm describe-maintenance-window-execution-tasks \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE"
```

Saída:

```
{
    "WindowExecutionTaskIdentities": [
        {
            "Status": "SUCCESS",
            "TaskArn": "AWS-RunShellScript",
            "StartTime": 1487692834.684,
            "TaskType": "RUN_COMMAND",
            "EndTime": 1487692835.005,
            "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2EXAMPLE",
            "TaskExecutionId": "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"
        }
    ]
}
```

Para obter mais informações, consulte Visualizar informações sobre tarefas e execuções de tarefas (AWS CLI) no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribeMaintenanceWindowExecutionTasks na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo lista todas as tarefas associadas à execução de uma janela de manutenção.

```
Get-SSMMaintenanceWindowExecutionTaskList -WindowExecutionId
 "518d5565-5969-4cca-8f0e-da3b2a638355"
```

Saída:

```
EndTime           : 2/21/2017 4:00:35 PM
StartTime         : 2/21/2017 4:00:34 PM
Status            : SUCCESS
TaskArn           : AWS-RunShellScript
TaskExecutionId   : ac0c6ae1-daa3-4a89-832e-d384503b6586
TaskType          : RUN_COMMAND
WindowExecutionId : 518d5565-5969-4cca-8f0e-da3b2a638355
```

- Para obter detalhes da API, consulte DescribeMaintenanceWindowExecutionTasks na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo lista todas as tarefas associadas à execução de uma janela de manutenção.

```
Get-SSMMaintenanceWindowExecutionTaskList -WindowExecutionId
 "518d5565-5969-4cca-8f0e-da3b2a638355"
```

Saída:

```
EndTime           : 2/21/2017 4:00:35 PM
StartTime         : 2/21/2017 4:00:34 PM
Status            : SUCCESS
TaskArn           : AWS-RunShellScript
TaskExecutionId   : ac0c6ae1-daa3-4a89-832e-d384503b6586
TaskType          : RUN_COMMAND
WindowExecutionId : 518d5565-5969-4cca-8f0e-da3b2a638355
```

- Para obter detalhes da API, consulte DescribeMaintenanceWindowExecutionTasks na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeMaintenanceWindowExecutions`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeMaintenanceWindowExecutions`.

CLI

AWS CLI

Exemplo 1: listar todas as execuções para uma janela de manutenção

O exemplo de `describe-maintenance-window-executions` a seguir lista todas as execuções da janela de manutenção especificada.

```
aws ssm describe-maintenance-window-executions \
    --window-id "mw-ab12cd34eEXAMPLE"
```

Saída:

```
{
    "WindowExecutions": [
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowExecutionId": "6027b513-64fe-4cf0-be7d-1191aEXAMPLE",
            "Status": "IN_PROGRESS",
            "StartTime": "2021-08-04T11:00:00.000000-07:00"

        },
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowExecutionId": "ff75b750-4834-4377-8f61-b3cadEXAMPLE",
            "Status": "SUCCESS",
            "StartTime": "2021-08-03T11:00:00.000000-07:00",
            "EndTime": "2021-08-03T11:37:21.450000-07:00"
        },
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowExecutionId": "9fac7dd9-ff21-42a5-96ad-bbc4bEXAMPLE",
            "Status": "FAILED",
            "StatusDetails": "One or more tasks in the orchestration failed.",
            "StartTime": "2021-08-02T11:00:00.000000-07:00",
            "EndTime": "2021-08-02T11:22:36.190000-07:00"
        }
    ]
}
```

Exemplo 2: listar todas as execuções para uma janela de manutenção antes de uma data especificada

O exemplo de `describe-maintenance-window-executions` a seguir lista todas as execuções da janela de manutenção especificada antes da data especificada.

```
aws ssm describe-maintenance-window-executions \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=ExecutedBefore,Values=2021-08-03T00:00:00Z"
```

Saída:

```
{
    "WindowExecutions": [
        {
        "WindowId": "mw-ab12cd34eEXAMPLE",
        "WindowExecutionId": "9fac7dd9-ff21-42a5-96ad-bbc4bEXAMPLE",
        "Status": "FAILED",
        "StatusDetails": "One or more tasks in the orchestration failed.",
        "StartTime": "2021-08-02T11:00:00.000000-07:00",
        "EndTime": "2021-08-02T11:22:36.190000-07:00"
    }
    ]
}
```

Exemplo 3: listar todas as execuções para uma janela de manutenção após uma data especificada

O exemplo de `describe-maintenance-window-executions` a seguir lista todas as execuções da janela de manutenção especificada após a data especificada.

```
aws ssm describe-maintenance-window-executions \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=ExecutedAfter,Values=2021-08-04T00:00:00Z"
```

Saída:

```
{
    "WindowExecutions": [
        {
        "WindowId": "mw-ab12cd34eEXAMPLE",
        "WindowExecutionId": "6027b513-64fe-4cf0-be7d-1191aEXAMPLE",
        "Status": "IN_PROGRESS",
        "StartTime": "2021-08-04T11:00:00.000000-07:00"
        }
    ]
}
```

Para obter mais informações, consulte Visualizar informações sobre tarefas e execuções de tarefas (AWS CLI) no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribeMaintenanceWindowExecutions na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo lista todas as execuções para uma janela de manutenção.

```
Get-SSMMaintenanceWindowExecutionList -WindowId "mw-03eb9db42890fb82d"
```

Saída:

```
EndTime           : 2/20/2017 6:30:17 PM
StartTime         : 2/20/2017 6:30:16 PM
Status            : FAILED
StatusDetails     : One or more tasks in the orchestration failed.
WindowExecutionId : 6f3215cf-4101-4fa0-9b7b-9523269599c7
WindowId          : mw-03eb9db42890fb82d
```

Exemplo 2: esse exemplo lista todas as execuções para uma janela de manutenção antes de uma data especificada.

```
$option1 = @{Key="ExecutedBefore";Values=@("2016-11-04T05:00:00Z")}
Get-SSMMaintenanceWindowExecutionList -WindowId "mw-03eb9db42890fb82d" -Filter
 $option1
```

Exemplo 3: esse exemplo lista todas as execuções para uma janela de manutenção após uma data especificada.

```
$option1 = @{Key="ExecutedAfter";Values=@("2016-11-04T05:00:00Z")}
Get-SSMMaintenanceWindowExecutionList -WindowId "mw-03eb9db42890fb82d" -Filter
 $option1
```

- Para obter detalhes da API, consulte DescribeMaintenanceWindowExecutions na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo lista todas as execuções para uma janela de manutenção.

```
Get-SSMMaintenanceWindowExecutionList -WindowId "mw-03eb9db42890fb82d"
```

Saída:

```
EndTime           : 2/20/2017 6:30:17 PM
StartTime         : 2/20/2017 6:30:16 PM
Status            : FAILED
StatusDetails     : One or more tasks in the orchestration failed.
WindowExecutionId : 6f3215cf-4101-4fa0-9b7b-9523269599c7
WindowId          : mw-03eb9db42890fb82d
```

Exemplo 2: esse exemplo lista todas as execuções para uma janela de manutenção antes de uma data especificada.

```
$option1 = @{Key="ExecutedBefore";Values=@("2016-11-04T05:00:00Z")}
Get-SSMMaintenanceWindowExecutionList -WindowId "mw-03eb9db42890fb82d" -Filter
 $option1
```

Exemplo 3: esse exemplo lista todas as execuções para uma janela de manutenção após uma data especificada.

```
$option1 = @{Key="ExecutedAfter";Values=@("2016-11-04T05:00:00Z")}
Get-SSMMaintenanceWindowExecutionList -WindowId "mw-03eb9db42890fb82d" -Filter
 $option1
```

- Para obter detalhes da API, consulte DescribeMaintenanceWindowExecutions na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeMaintenanceWindowTargets`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeMaintenanceWindowTargets`.

CLI

AWS CLI

Exemplo 1: listar todos os destinos para uma janela de manutenção

O exemplo de `describe-maintenance-window-targets` a seguir lista todos os destinos para uma janela de manutenção.

```
aws ssm describe-maintenance-window-targets \
    --window-id "mw-06cf17cbefEXAMPLE"
```

Saída:

```
{
    "Targets": [
        {
            "ResourceType": "INSTANCE",
            "OwnerInformation": "Single instance",
            "WindowId": "mw-06cf17cbefEXAMPLE",
            "Targets": [
                {
                    "Values": [
                        "i-0000293ffdEXAMPLE"
                    ],
                    "Key": "InstanceIds"
                }
            ],
            "WindowTargetId": "350d44e6-28cc-44e2-951f-4b2c9EXAMPLE"
        },
        {
            "ResourceType": "INSTANCE",
            "OwnerInformation": "Two instances in a list",
            "WindowId": "mw-06cf17cbefEXAMPLE",
            "Targets": [
                {
                    "Values": [
                        "i-0000293ffdEXAMPLE",
                        "i-0cb2b964d3EXAMPLE"
                    ],
                    "Key": "InstanceIds"
                }
            ],
```

```
            "WindowTargetId": "e078a987-2866-47be-bedd-d9cf4EXAMPLE"
        }
    ]
}
```

Exemplo 2: listar todos os destinos para uma janela de manutenção que correspondem ao valor das informações de um proprietário específico

Esse exemplo de `describe-maintenance-window-targets` lista todos os destinos de uma janela de manutenção com um valor específico.

```
aws ssm describe-maintenance-window-targets \
    --window-id "mw-0ecb1226ddEXAMPLE" \
    --filters "Key=OwnerInformation,Values=CostCenter1"
```

Saída:

```
{
    "Targets": [
        {
            "WindowId": "mw-0ecb1226ddEXAMPLE",
            "WindowTargetId": "da89dcc3-7f9c-481d-ba2b-edcb7d0057f9",
            "ResourceType": "INSTANCE",
            "Targets": [
                {
                    "Key": "tag:Environment",
                    "Values": [
                        "Prod"
                    ]
                }
            ],
            "OwnerInformation": "CostCenter1",
            "Name": "ProdTarget1"
        }
    ]
}
```

Para obter mais informações, consulte Visualizar informações sobre janelas de manutenção (AWS CLI) no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribeMaintenanceWindowTargets na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo lista todos os destinos para uma janela de manutenção.

```
Get-SSMMaintenanceWindowTarget -WindowId "mw-06cf17cbefcb4bf4f"
```

Saída:

```
OwnerInformation : Single instance
ResourceType     : INSTANCE
Targets          : {InstanceIds}
WindowId         : mw-06cf17cbefcb4bf4f
WindowTargetId   : 350d44e6-28cc-44e2-951f-4b2c985838f6

OwnerInformation : Two instances in a list
ResourceType     : INSTANCE
Targets          : {InstanceIds}
WindowId         : mw-06cf17cbefcb4bf4f
WindowTargetId   : e078a987-2866-47be-bedd-d9cf49177d3a
```

- Para obter detalhes da API, consulte DescribeMaintenanceWindowTargets na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo lista todos os destinos para uma janela de manutenção.

```
Get-SSMMaintenanceWindowTarget -WindowId "mw-06cf17cbefcb4bf4f"
```

Saída:

```
OwnerInformation : Single instance
ResourceType     : INSTANCE
Targets          : {InstanceIds}
WindowId         : mw-06cf17cbefcb4bf4f
WindowTargetId   : 350d44e6-28cc-44e2-951f-4b2c985838f6

OwnerInformation : Two instances in a list
ResourceType     : INSTANCE
Targets          : {InstanceIds}
WindowId         : mw-06cf17cbefcb4bf4f
```

```
WindowTargetId   : e078a987-2866-47be-bedd-d9cf49177d3a
```

- Para obter detalhes da API, consulte DescribeMaintenanceWindowTargets na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeMaintenanceWindowTasks`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeMaintenanceWindowTasks`.

CLI

AWS CLI

Exemplo 1: listar todas as tarefas para uma janela de manutenção

O exemplo de `describe-maintenance-window-tasks` a seguir lista todas as tarefas para a janela de manutenção especificada.

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-06cf17cbefEXAMPLE"
```

Saída:

```
{
    "Tasks": [
        {
            "WindowId": "mw-06cf17cbefEXAMPLE",
            "WindowTaskId": "018b31c3-2d77-4b9e-bd48-c91edEXAMPLE",
            "TaskArn": "AWS-RestartEC2Instance",
            "TaskParameters": {},
            "Type": "AUTOMATION",
            "Description": "Restarting EC2 Instance for maintenance",
            "MaxConcurrency": "1",
            "MaxErrors": "1",
            "Name": "My-Automation-Example-Task",
            "Priority": 0,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
```

```
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ]
        },
        {
            "WindowId": "mw-06cf17cbefEXAMPLE",
            "WindowTaskId": "1943dee0-0a17-4978-9bf4-3cc2fEXAMPLE",
            "TaskArn": "AWS-DisableS3BucketPublicReadWrite",
            "TaskParameters": {},
            "Type": "AUTOMATION",
            "Description": "Automation task to disable read/write access on
 public S3 buckets",
            "MaxConcurrency": "10",
            "MaxErrors": "5",
            "Name": "My-Disable-S3-Public-Read-Write-Access-Automation-Task",
            "Priority": 0,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ]
        }
    ]
}
```

Exemplo 2: listar todas as tarefas para uma janela de manutenção que invoca o documento de comandos AWS-RunPowerShellScript

O exemplo de `describe-maintenance-window-tasks` a seguir lista todas as tarefas para a janela de manutenção especificada que invoca o documento de comandos do `AWS-RunPowerShellScript`.

```
aws ssm describe-maintenance-window-tasks \
```

```
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=TaskArn,Values=AWS-RunPowerShellScript"
```

Saída:

```
{
    "Tasks": [
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowTaskId": "0d36e6b4-3a4f-411e-adcb-3558eEXAMPLE",
            "TaskArn": "AWS-RunPowerShellScript",
            "Type": "RUN_COMMAND",
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ],
            "TaskParameters": {},
            "Priority": 1,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "MaxConcurrency": "1",
            "MaxErrors": "1",
            "Name": "MyTask"
        }
    ]
}
```

Exemplo 3: listar todas as tarefas de uma janela de manutenção que têm a prioridade igual a 3

O exemplo de `describe-maintenance-window-tasks` a seguir lista todas as tarefas para a janela de manutenção especificada que tem `Priority` igual a `3`.

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=Priority,Values=3"
```

Saída:

```
{
    "Tasks": [
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowTaskId": "0d36e6b4-3a4f-411e-adcb-3558eEXAMPLE",
            "TaskArn": "AWS-RunPowerShellScript",
            "Type": "RUN_COMMAND",
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ],
            "TaskParameters": {},
            "Priority": 3,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "MaxConcurrency": "1",
            "MaxErrors": "1",
            "Name": "MyRunCommandTask"
        },
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowTaskId": "ee45feff-ad65-4a6c-b478-5cab8EXAMPLE",
            "TaskArn": "AWS-RestartEC2Instance",
            "Type": "AUTOMATION",
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ],
            "TaskParameters": {},
            "Priority": 3,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "MaxConcurrency": "10",
            "MaxErrors": "5",
            "Name": "My-Automation-Task",
```

```
            "Description": "A description for my Automation task"
        }
    ]
}
```

Exemplo 4: listar todas as tarefas de uma janela de manutenção que têm a prioridade igual a 1 e usam o Run Command

Esse exemplo de `describe-maintenance-window-tasks` lista todas as tarefas para a janela de manutenção especificada que tem `Priority` igual a 1 e usam `Run Command`.

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=Priority,Values=1" "Key=TaskType,Values=RUN_COMMAND"
```

Saída:

```
{
    "Tasks": [
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowTaskId": "0d36e6b4-3a4f-411e-adcb-3558eEXAMPLE",
            "TaskArn": "AWS-RunPowerShellScript",
            "Type": "RUN_COMMAND",
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ],
            "TaskParameters": {},
            "Priority": 1,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "MaxConcurrency": "1",
            "MaxErrors": "1",
            "Name": "MyRunCommandTask"
        }
    ]
}
```

Para obter mais informações, consulte Visualizar informações sobre janelas de manutenção (AWS CLI) no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribeMaintenanceWindowTasks na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo lista todas as tarefas para uma janela de manutenção.

```
Get-SSMMaintenanceWindowTaskList -WindowId "mw-06cf17cbefcb4bf4f"
```

Saída:

```
LoggingInfo    :
MaxConcurrency : 1
MaxErrors      : 1
Priority       : 10
ServiceRoleArn : arn:aws:iam::123456789012:role/MaintenanceWindowsRole
Targets        : {InstanceIds}
TaskArn        : AWS-RunShellScript
TaskParameters : {[commands,
 Amazon.SimpleSystemsManagement.Model.MaintenanceWindowTaskParameterValueExpression]}
Type           : RUN_COMMAND
WindowId       : mw-06cf17cbefcb4bf4f
WindowTaskId   : a23e338d-ff30-4398-8aa3-09cd052ebf17
```

- Para obter detalhes da API, consulte DescribeMaintenanceWindowTasks na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo lista todas as tarefas para uma janela de manutenção.

```
Get-SSMMaintenanceWindowTaskList -WindowId "mw-06cf17cbefcb4bf4f"
```

Saída:

```
LoggingInfo    :
```

```
MaxConcurrency : 1
MaxErrors      : 1
Priority       : 10
ServiceRoleArn : arn:aws:iam::123456789012:role/MaintenanceWindowsRole
Targets        : {InstanceIds}
TaskArn        : AWS-RunShellScript
TaskParameters : {[commands,
 Amazon.SimpleSystemsManagement.Model.MaintenanceWindowTaskParameterValueExpression]}
Type           : RUN_COMMAND
WindowId       : mw-06cf17cbefcb4bf4f
WindowTaskId   : a23e338d-ff30-4398-8aa3-09cd052ebf17
```

- Para obter detalhes da API, consulte DescribeMaintenanceWindowTasks na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeMaintenanceWindows`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeMaintenanceWindows`.

CLI

AWS CLI

Exemplo 1: listar todas as janelas de manutenção

O exemplo de `describe-maintenance-windows` a seguir lista todas as janelas de manutenção em sua conta da AWS na região atual.

```
aws ssm describe-maintenance-windows
```

Saída:

```
{
    "WindowIdentities": [
        {
            "WindowId": "mw-0ecb1226ddEXAMPLE",
```

```
            "Name": "MyMaintenanceWindow-1",
            "Enabled": true,
            "Duration": 2,
            "Cutoff": 1,
            "Schedule": "rate(180 minutes)",
            "NextExecutionTime": "2020-02-12T23:19:20.596Z"
        },
        {
            "WindowId": "mw-03eb9db428EXAMPLE",
            "Name": "MyMaintenanceWindow-2",
            "Enabled": true,
            "Duration": 3,
            "Cutoff": 1,
            "Schedule": "rate(7 days)",
            "NextExecutionTime": "2020-02-17T23:22:00.956Z"
        },
    ]
}
```

Exemplo 2: listar todas as janelas de manutenção habilitadas

O exemplo de `describe-maintenance-windows` a seguir lista todas as janelas de manutenção habilitadas.

```
aws ssm describe-maintenance-windows \
    --filters "Key=Enabled,Values=true"
```

Exemplo 3: listar janelas de manutenção que correspondem a um nome específico

Esse exemplo de `describe-maintenance-windows` lista todas as janelas de manutenção com o nome especificado.

```
aws ssm describe-maintenance-windows \
    --filters "Key=Name,Values=MyMaintenanceWindow"
```

Para obter mais informações, consulte Visualizar informações sobre janelas de manutenção (AWS CLI) no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribeMaintenanceWindows na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo lista todas as janelas de manutenção em sua conta.

```
Get-SSMMaintenanceWindowList
```

Saída:

```
Cutoff   : 1
Duration : 4
Enabled  : True
Name     : My-First-Maintenance-Window
WindowId : mw-06d59c1a07c022145
```

- Para obter detalhes da API, consulte DescribeMaintenanceWindows na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo lista todas as janelas de manutenção em sua conta.

```
Get-SSMMaintenanceWindowList
```

Saída:

```
Cutoff   : 1
Duration : 4
Enabled  : True
Name     : My-First-Maintenance-Window
WindowId : mw-06d59c1a07c022145
```

- Para obter detalhes da API, consulte DescribeMaintenanceWindows na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeOpsItems`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `DescribeOpsItems`.

CLI

AWS CLI

Para listar um conjunto de OpsItems

O exemplo de `describe-ops-items` a seguir exibe uma lista de todos os OpsItems abertos na conta da AWS.

```
aws ssm describe-ops-items \
    --ops-item-filters "Key=Status,Values=Open,Operator=Equal"
```

Saída:

```
{
    "OpsItemSummaries": [
        {
            "CreatedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-
Role/fbf77cbe264a33509569f23e4EXAMPLE",
            "CreatedTime": "2020-03-14T17:02:46.375000-07:00",
            "LastModifiedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-
CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
            "LastModifiedTime": "2020-03-14T17:02:46.375000-07:00",
            "Source": "SSM",
            "Status": "Open",
            "OpsItemId": "oi-7cfc5EXAMPLE",
            "Title": "SSM Maintenance Window execution failed",
            "OperationalData": {
                "/aws/dedup": {
                    "Value": "{\"dedupString\":\"SSMOpsItems-SSM-maintenance-
window-execution-failed\"}",
                    "Type": "SearchableString"
                },
                "/aws/resources": {
                    "Value": "[{\"arn\":\"arn:aws:ssm:us-
east-2:111222333444:maintenancewindow/mw-034093d322EXAMPLE\"}]",
                    "Type": "SearchableString"
                }
```

```
            },
            "Category": "Availability",
            "Severity": "3"
        },
        {
            "CreatedBy": "arn:aws:sts::1112223233444:assumed-role/OpsItem-CWE-
Role/fbf77cbe264a33509569f23e4EXAMPLE",
            "CreatedTime": "2020-02-26T11:43:15.426000-08:00",
            "LastModifiedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-
CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
            "LastModifiedTime": "2020-02-26T11:43:15.426000-08:00",
            "Source": "EC2",
            "Status": "Open",
            "OpsItemId": "oi-6f966EXAMPLE",
            "Title": "EC2 instance stopped",
            "OperationalData": {
                "/aws/automations": {
                    "Value": "[ { \"automationType\": \"AWS:SSM:Automation\",
 \"automationId\": \"AWS-RestartEC2Instance\" } ]",
                    "Type": "SearchableString"
                },
                "/aws/dedup": {
                    "Value": "{\"dedupString\":\"SSMOpsItems-EC2-instance-stopped
\"}",
                    "Type": "SearchableString"
                },
                "/aws/resources": {
                    "Value": "[{\"arn\":\"arn:aws:ec2:us-
east-2:111222333444:instance/i-0beccfbc02EXAMPLE\"}]",
                    "Type": "SearchableString"
                }
            },
            "Category": "Availability",
            "Severity": "3"
        }
    ]
}
```

Para obter mais informações, consulte Gerenciamento de OpsItems no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribeOpsItems na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
    /**
     * Describes AWS SSM OpsItems asynchronously.
     *
     * @param key The key to filter OpsItems by (e.g., OPS_ITEM_ID).
     *
     * This method initiates an asynchronous request to describe SSM OpsItems.
     * If the request is successful, it prints the title and status of each
OpsItem.
     * If an exception occurs, it handles the error appropriately.
     */
    public void describeOpsItems(String key) {
        OpsItemFilter filter = OpsItemFilter.builder()
                .key(OpsItemFilterKey.OPS_ITEM_ID)
                .values(key)
                .operator(OpsItemFilterOperator.EQUAL)
                .build();

        DescribeOpsItemsRequest itemsRequest = DescribeOpsItemsRequest.builder()
                .maxResults(10)
                .opsItemFilters(filter)
                .build();

        CompletableFuture<Void> future = CompletableFuture.runAsync(() -> {
            getAsyncClient().describeOpsItems(itemsRequest)
                    .thenAccept(itemsResponse -> {
                        List<OpsItemSummary> items =
itemsResponse.opsItemSummaries();
                        for (OpsItemSummary item : items) {
                            System.out.println("The item title is " +
item.title() + " and the status is " + item.status().toString());
                        }
                    })
```

```
                    .exceptionally(ex -> {
                        throw new CompletionException(ex);
                    }).join();
        }).exceptionally(ex -> {
            Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
            if (cause instanceof SsmException) {
                throw new RuntimeException("SSM error: " + cause.getMessage(),
cause);
            } else {
                throw new RuntimeException("Unexpected error: " +
cause.getMessage(), cause);
            }
        });

        try {
            future.join();
        } catch (CompletionException ex) {
            throw ex.getCause() instanceof RuntimeException ? (RuntimeException)
ex.getCause() : ex;
        }
    }
```

- Para obter detalhes da API, consulte DescribeOpsItems na Referência da API AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
import {
  OpsItemFilterOperator,
  OpsItemFilterKey,
  paginateDescribeOpsItems,
```

```
  SSMClient,
} from "@aws-sdk/client-ssm";
import { parseArgs } from "node:util";

/**
 * Describe SSM OpsItems.
 * @param {{ opsItemId: string }}
 */
export const main = async ({ opsItemId }) => {
  const client = new SSMClient({});
  try {
    const describeOpsItemsPaginated = [];
    for await (const page of paginateDescribeOpsItems(
      { client },
      {
        OpsItemFilters: {
          Key: OpsItemFilterKey.OPSITEM_ID,
          Operator: OpsItemFilterOperator.EQUAL,
          Values: opsItemId,
        },
      },
    )) {
      describeOpsItemsPaginated.push(...page.OpsItemSummaries);
    }
    console.log("Here are the ops items:");
    console.log(describeOpsItemsPaginated);
    return { OpsItemSummaries: describeOpsItemsPaginated };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "MissingParameter") {
      console.warn(`${caught.message}. Did you provide this value?`);
    }
    throw caught;
  }
};
```

- Para obter detalhes da API, consulte DescribeOpsItems na Referência da API AWS SDK para JavaScript.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
class OpsItemWrapper:
    """Encapsulates AWS Systems Manager OpsItem actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.id = None

    @classmethod
    def from_client(cls):
        """
        :return: A OpsItemWrapper instance.
        """
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def describe(self):
        """
        Describe an OpsItem.
        """
        try:
            paginator = self.ssm_client.get_paginator("describe_ops_items")
            ops_items = []
            for page in paginator.paginate(
                OpsItemFilters=[
                    {"Key": "OpsItemId", "Values": [self.id], "Operator":
 "Equal"}
                ]
            ):
```

```
                ops_items.extend(page["OpsItemSummaries"])

            for item in ops_items:
                print(
                    f"The item title is {item['Title']} and the status is
{item['Status']}"
                )
            return len(ops_items) > 0
        except ClientError as err:
            logger.error(
                "Couldn't describe ops item %s. Here's why: %s: %s",
                self.id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Para obter detalhes da API, consulte DescribeOpsItems na Referência da API AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeParameters`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `DescribeParameters`.

CLI

AWS CLI

Exemplo 1: listar todos os parâmetros

O exemplo de `describe-parameters` a seguir lista todos os parâmetros na conta e região atuais da AWS.

```
aws ssm describe-parameters
```

Saída:

```
{
    "Parameters": [
        {
            "Name": "MySecureStringParameter",
            "Type": "SecureString",
            "KeyId": "alias/aws/ssm",
            "LastModifiedDate": 1582155479.205,
            "LastModifiedUser": "arn:aws:sts::111222333444:assumed-role/Admin/
Richard-Roe-Managed",
            "Description": "This is a SecureString parameter",
            "Version": 2,
            "Tier": "Advanced",
            "Policies": [
                {
                    "PolicyText": "{\"Type\":\"Expiration\",\"Version\":\"1.0\",
\"Attributes\":{\"Timestamp\":\"2020-07-07T22:30:00Z\"}}",
                    "PolicyType": "Expiration",
                    "PolicyStatus": "Pending"
                },
                {
                    "PolicyText": "{\"Type\":\"ExpirationNotification\",\"Version
\":\"1.0\",\"Attributes\":{\"Before\":\"12\",\"Unit\":\"Hours\"}}",
                    "PolicyType": "ExpirationNotification",
                    "PolicyStatus": "Pending"
                }
            ]
        },
        {
            "Name": "MyStringListParameter",
            "Type": "StringList",
            "LastModifiedDate": 1582154764.222,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
            "Description": "This is a StringList parameter",
            "Version": 1,
            "Tier": "Standard",
            "Policies": []
        },
        {
            "Name": "MyStringParameter",
            "Type": "String",
            "LastModifiedDate": 1582154711.976,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Alejandro-
Rosalez",
```

```
            "Description": "This is a String parameter",
            "Version": 1,
            "Tier": "Standard",
            "Policies": []
        },
        {
            "Name": "latestAmi",
            "Type": "String",
            "LastModifiedDate": 1580862415.521,
            "LastModifiedUser": "arn:aws:sts::111222333444:assumed-role/lambda-
ssm-role/Automation-UpdateSSM-Param",
            "Version": 3,
            "Tier": "Standard",
            "Policies": []
        }
    ]
}
```

Exemplo 2: como listar todos os parâmetros que correspondem a metadados específicos

Esse exemplo de `describe-parameters` lista todos os parâmetros que correspondem a um filtro.

aws ssm describe-parameters --filters "Key=Type,Values=StringList"

Saída:

```
{
    "Parameters": [
        {
            "Name": "MyStringListParameter",
            "Type": "StringList",
            "LastModifiedDate": 1582154764.222,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
            "Description": "This is a StringList parameter",
            "Version": 1,
            "Tier": "Standard",
            "Policies": []
        }
    ]
}
```

Para obter mais informações, consulte Pesquisando parâmetros do Systems Manager, no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribeParameters na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.ssm.SsmClient;
import software.amazon.awssdk.services.ssm.model.GetParameterRequest;
import software.amazon.awssdk.services.ssm.model.GetParameterResponse;
import software.amazon.awssdk.services.ssm.model.SsmException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class GetParameter {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <paraName>

                Where:
                    paraName - The name of the parameter.
                """;
```

```
        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String paraName = args[0];
        Region region = Region.US_EAST_1;
        SsmClient ssmClient = SsmClient.builder()
                .region(region)
                .build();

        getParaValue(ssmClient, paraName);
        ssmClient.close();
    }

    public static void getParaValue(SsmClient ssmClient, String paraName) {
        try {
            GetParameterRequest parameterRequest = GetParameterRequest.builder()
                    .name(paraName)
                    .build();

            GetParameterResponse parameterResponse =
 ssmClient.getParameter(parameterRequest);
            System.out.println("The parameter value is " +
 parameterResponse.parameter().value());

        } catch (SsmException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte DescribeParameters na Referência da API AWS SDK for Java 2.x.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo lista todos os parâmetros.

```
Get-SSMParameterList
```

Saída:

```
Description      :
KeyId            :
LastModifiedDate : 3/3/2017 6:58:23 PM
LastModifiedUser : arn:aws:iam::123456789012:user/admin
Name             : Welcome
Type             : String
```

- Para obter detalhes da API, consulte DescribeParameters na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo lista todos os parâmetros.

```
Get-SSMParameterList
```

Saída:

```
Description      :
KeyId            :
LastModifiedDate : 3/3/2017 6:58:23 PM
LastModifiedUser : arn:aws:iam::123456789012:user/admin
Name             : Welcome
Type             : String
```

- Para obter detalhes da API, consulte DescribeParameters na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Rust

SDK para Rust

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
async fn show_parameters(client: &Client) -> Result<(), Error> {
    let resp = client.describe_parameters().send().await?;

    for param in resp.parameters() {
        println!("  {}", param.name().unwrap_or_default());
    }

    Ok(())
}
```

- Para obter detalhes da API, consulte DescribeParameters na Referência da API AWS SDK para Rust.

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribePatchBaselines`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribePatchBaselines`.

CLI

AWS CLI

Exemplo 1: Para listar todas as linhas de base do patch

O exemplo de `describe-patch-baselines` a seguir recupera os detalhes de todas as listas de referências de patches da sua conta na região atual.

```
aws ssm describe-patch-baselines
```

Saída:

```
{
    "BaselineIdentities": [
        {
            "BaselineName": "AWS-SuseDefaultPatchBaseline",
            "DefaultBaseline": true,
```

```
            "BaselineDescription": "Default Patch Baseline for Suse Provided by
 AWS.",
            "BaselineId": "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-0123fdb36e334a3b2",
            "OperatingSystem": "SUSE"
        },
        {
            "BaselineName": "AWS-DefaultPatchBaseline",
            "DefaultBaseline": false,
            "BaselineDescription": "Default Patch Baseline Provided by AWS.",
            "BaselineId": "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-020d361a05defe4ed",
            "OperatingSystem": "WINDOWS"
        },
        ...
        {
            "BaselineName": "MyWindowsPatchBaseline",
            "DefaultBaseline": true,
            "BaselineDescription": "My patch baseline for EC2 instances for
 Windows Server",
            "BaselineId": "pb-0ad00e0dd7EXAMPLE",
            "OperatingSystem": "WINDOWS"
        }
    ]
}
```

Exemplo 2: listar todas as listas de referência de patches fornecidas pela AWS

O exemplo de `describe-patch-baselines` a seguir lista todas as listas de referência de patches fornecidas pela AWS.

```
aws ssm describe-patch-baselines \
    --filters "Key=OWNER,Values=[AWS]"
```

Exemplo 3: listar todas as listas de referência de patches pertencentes a você

O exemplo de `describe-patch-baselines` a seguir lista todas as listas de referências de patches criadas em sua conta na região atual.

```
aws ssm describe-patch-baselines \
    --filters "Key=OWNER,Values=[Self]"
```

Para obter mais informações, consulte Sobre listas de referência de patches predefinidas e personalizadas no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribePatchBaselines na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo mostra todas as listas de referência de patches.

```
Get-SSMPatchBaseline
```

Saída:

```
BaselineDescription                                                BaselineId
                                                  BaselineName
------------------                                                ----------
                                                  ------------
Default Patch Baseline Provided by AWS.                           arn:aws:ssm:us-
west-2:123456789012:patchbaseline/pb-04fb4ae6142167966 AWS-DefaultP...
Baseline containing all updates approved for production systems
 pb-045f10b4f382baeda
 Production-B...
Baseline containing all updates approved for production systems
 pb-0a2f1059b670ebd31
 Production-B...
```

Exemplo 2: esse exemplo mostra todas as listas de referência de patches fornecidas pela AWS. A sintaxe usada nesse exemplo requer o PowerShell versão 3 ou posterior.

```
$filter1 = @{Key="OWNER";Values=@("AWS")}
```

Saída:

```
Get-SSMPatchBaseline -Filter $filter1
```

Exemplo 3: esse exemplo mostra todas as listas de referência de patches pertencentes a você. A sintaxe usada nesse exemplo requer o PowerShell versão 3 ou posterior.

```
$filter1 = @{Key="OWNER";Values=@("Self")}
```

Saída:

```
Get-SSMPatchBaseline -Filter $filter1
```

Exemplo 4: com o PowerShell versão 2, é necessário usar New-Object para criar cada tag.

```
$filter1 = New-Object
 Amazon.SimpleSystemsManagement.Model.PatchOrchestratorFilter
$filter1.Key = "OWNER"
$filter1.Values = "AWS"

Get-SSMPatchBaseline -Filter $filter1
```

Saída:

```
BaselineDescription                     BaselineId
                            BaselineName             DefaultBaselin

                                                     e
-------------------                     ----------
                            ------------             --------------
Default Patch Baseline Provided by AWS. arn:aws:ssm:us-
west-2:123456789012:patchbaseline/pb-04fb4ae6142167966 AWS-DefaultPatchBaseline
 True
```

- Para obter detalhes da API, consulte DescribePatchBaselines na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo mostra todas as listas de referência de patches.

```
Get-SSMPatchBaseline
```

Saída:

```
BaselineDescription                                                BaselineId
                                                  BaselineName
```

```
------------------                                                  ----------
                                                          ------------
Default Patch Baseline Provided by AWS.                             arn:aws:ssm:us-
west-2:123456789012:patchbaseline/pb-04fb4ae6142167966 AWS-DefaultP...
Baseline containing all updates approved for production systems
 pb-045f10b4f382baeda
 Production-B...
Baseline containing all updates approved for production systems
 pb-0a2f1059b670ebd31
 Production-B...
```

Exemplo 2: esse exemplo mostra todas as listas de referência de patches fornecidas pela AWS. A sintaxe usada nesse exemplo requer o PowerShell versão 3 ou posterior.

```
$filter1 = @{Key="OWNER";Values=@("AWS")}
```

Saída:

```
Get-SSMPatchBaseline -Filter $filter1
```

Exemplo 3: esse exemplo mostra todas as listas de referência de patches pertencentes a você. A sintaxe usada nesse exemplo requer o PowerShell versão 3 ou posterior.

```
$filter1 = @{Key="OWNER";Values=@("Self")}
```

Saída:

```
Get-SSMPatchBaseline -Filter $filter1
```

Exemplo 4: com o PowerShell versão 2, é necessário usar New-Object para criar cada tag.

```
$filter1 = New-Object
 Amazon.SimpleSystemsManagement.Model.PatchOrchestratorFilter
$filter1.Key = "OWNER"
$filter1.Values = "AWS"

Get-SSMPatchBaseline -Filter $filter1
```

Saída:

```
BaselineDescription                     BaselineId
                            BaselineName            DefaultBaselin

                                                    e
------------------                      ----------
                            ------------            --------------
Default Patch Baseline Provided by AWS. arn:aws:ssm:us-
west-2:123456789012:patchbaseline/pb-04fb4ae6142167966 AWS-DefaultPatchBaseline
 True
```

- Para obter detalhes da API, consulte DescribePatchBaselines na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribePatchGroupState`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribePatchGroupState`.

CLI

AWS CLI

Para obter o estado de um grupo de patches

O exemplo de `describe-patch-group-state` a seguir recupera o resumo de conformidade de patches de alto nível para um grupo de patches.

```
aws ssm describe-patch-group-state \
    --patch-group "Production"
```

Saída:

```
{
    "Instances": 21,
    "InstancesWithCriticalNonCompliantPatches": 1,
    "InstancesWithFailedPatches": 2,
    "InstancesWithInstalledOtherPatches": 3,
    "InstancesWithInstalledPatches": 21,
```

```
    "InstancesWithInstalledPendingRebootPatches": 2,
    "InstancesWithInstalledRejectedPatches": 1,
    "InstancesWithMissingPatches": 3,
    "InstancesWithNotApplicablePatches": 4,
    "InstancesWithOtherNonCompliantPatches": 1,
    "InstancesWithSecurityNonCompliantPatches": 1,
    "InstancesWithUnreportedNotApplicablePatches": 2
}
```

Para obter mais informações, consulte Sobre grupos de patches <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-patchgroups.html>__ e Noções básicas sobre valores de estado de conformidade de patches no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribePatchGroupState na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo obtém o resumo da conformidade de patches de alto nível para um grupo de patches.

```
Get-SSMPatchGroupState -PatchGroup "Production"
```

Saída:

```
Instances                       : 4
InstancesWithFailedPatches      : 1
InstancesWithInstalledOtherPatches : 4
InstancesWithInstalledPatches   : 3
InstancesWithMissingPatches     : 0
InstancesWithNotApplicablePatches  : 0
```

- Para obter detalhes da API, consulte DescribePatchGroupState na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo obtém o resumo da conformidade de patches de alto nível para um grupo de patches.

```
Get-SSMPatchGroupState -PatchGroup "Production"
```

Saída:

```
Instances                          : 4
InstancesWithFailedPatches         : 1
InstancesWithInstalledOtherPatches : 4
InstancesWithInstalledPatches      : 3
InstancesWithMissingPatches        : 0
InstancesWithNotApplicablePatches  : 0
```

- Para obter detalhes da API, consulte DescribePatchGroupState na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribePatchGroups`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribePatchGroups`.

CLI

AWS CLI

Para exibir registros de grupos de patches

O exemplo de `describe-patch-groups` a seguir lista os registros de grupos de patches.

```
aws ssm describe-patch-groups
```

Saída:

```
{
    "Mappings": [
        {
            "PatchGroup": "Production",
            "BaselineIdentity": {
```

```
                "BaselineId": "pb-0123456789abcdef0",
                "BaselineName": "ProdPatching",
                "OperatingSystem": "WINDOWS",
                "BaselineDescription": "Patches for Production",
                "DefaultBaseline": false
            }
        },
        {
            "PatchGroup": "Development",
            "BaselineIdentity": {
                "BaselineId": "pb-0713accee01234567",
                "BaselineName": "DevPatching",
                "OperatingSystem": "WINDOWS",
                "BaselineDescription": "Patches for Development",
                "DefaultBaseline": true
            }
        },
        ...
    ]
}
```

Para obter mais informações, consulte Criar um grupo de patches <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html>__ e Adicionar um grupo de patches a uma lista de referência de patches no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte DescribePatchGroups na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo lista os registros do grupo de patches.

```
Get-SSMPatchGroup
```

Saída:

```
BaselineIdentity                                                PatchGroup
----------------                                                ----------
```

```
Amazon.SimpleSystemsManagement.Model.PatchBaselineIdentity Production
```

- Para obter detalhes da API, consulte DescribePatchGroups na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo lista os registros do grupo de patches.

```
Get-SSMPatchGroup
```

Saída:

```
BaselineIdentity                                                PatchGroup
----------------                                                ----------
Amazon.SimpleSystemsManagement.Model.PatchBaselineIdentity Production
```

- Para obter detalhes da API, consulte DescribePatchGroups na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`GetAutomationExecution`** com uma CLI

Os exemplos de código a seguir mostram como usar o `GetAutomationExecution`.

CLI

AWS CLI

Para exibir detalhes sobre uma execução do Automation

O exemplo de `get-automation-execution` a seguir exibe informações detalhadas sobre uma execução do Automation.

```
aws ssm get-automation-execution \
    --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

Saída:

```
{
    "AutomationExecution": {
        "AutomationExecutionId": "73c8eef8-f4ee-4a05-820c-e354fEXAMPLE",
        "DocumentName": "AWS-StartEC2Instance",
        "DocumentVersion": "1",
        "ExecutionStartTime": 1583737233.748,
        "ExecutionEndTime": 1583737234.719,
        "AutomationExecutionStatus": "Success",
        "StepExecutions": [
            {
                "StepName": "startInstances",
                "Action": "aws:changeInstanceState",
                "ExecutionStartTime": 1583737234.134,
                "ExecutionEndTime": 1583737234.672,
                "StepStatus": "Success",
                "Inputs": {
                    "DesiredState": "\"running\"",
                    "InstanceIds": "[\"i-0cb99161f6EXAMPLE\"]"
                },
                "Outputs": {
                    "InstanceStates": [
                        "running"
                    ]
                },
                "StepExecutionId": "95e70479-cf20-4d80-8018-7e4e2EXAMPLE",
                "OverriddenParameters": {}
            }
        ],
        "StepExecutionsTruncated": false,
        "Parameters": {
            "AutomationAssumeRole": [
                ""
            ],
            "InstanceId": [
                "i-0cb99161f6EXAMPLE"
            ]
        },
        "Outputs": {},
        "Mode": "Auto",
        "ExecutedBy": "arn:aws:sts::29884EXAMPLE:assumed-role/mw_service_role/
OrchestrationService",
        "Targets": [],
```

```
        "ResolvedTargets": {
            "ParameterValues": [],
            "Truncated": false
        }
    }
}
```

Para obter mais informações, consulte Passo a passo: corrigir uma AMI do Linux (AWS CLI) no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte GetAutomationExecution na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo exibe os detalhes de uma execução do Automation.

```
Get-SSMAutomationExecution -AutomationExecutionId "4105a4fc-
f944-11e6-9d32-8fb2db27a909"
```

Saída:

```
AutomationExecutionId     : 4105a4fc-f944-11e6-9d32-8fb2db27a909
AutomationExecutionStatus : Failed
DocumentName              : AWS-UpdateLinuxAmi
DocumentVersion           : 1
ExecutionEndTime          : 2/22/2017 9:17:08 PM
ExecutionStartTime        : 2/22/2017 9:17:02 PM
FailureMessage            : Step launchInstance failed maximum allowed times. You
 are not authorized to perform this operation. Encoded
                            authorization failure message:
 B_V2QyyN7NhSZQYpmVzpEc4oSnj2GLTNYnXUHsTbqJkNMoDgubmbtthLmZyaiUYekORIrA42-
fv1x-04q5Fjff6glh
                            Yb6TI5b0GQeeNrpwNvpDzmO-
PSR1swlAbg9fdM9BcNjyrznspUkWpuKu9EC1Ou6v3OXU1KC9nZ7mPlWMFZNkSioQqpwWEvMw-
GZktsQzm67qOhUhBNOLWYhbS
                            pkfiqzY-5nw3S0obx30fhd3EJa5O_-
GjV_a0nFXQJa70ik40bFOrEh3MtCSbrQT6--DvFy_FQ8TKvkIXadyVskeJI84XOF5WmA60f1pi5GI08i-
nRfZS6oDeU
```

```
 gELBjjoFKD8s3L2aI0B6umWVxnQOjqhQRxwJ53b54sZJ2PW3v_mtg9-q0CK0ezS3xfh_y0ilaUGOAZG-
xjQFuvU_JZedWpla3xi-MZsmblAifBI
                            (Service: AmazonEC2; Status Code: 403; Error Code:
 UnauthorizedOperation; Request ID:
                            6a002f94-ba37-43fd-99e6-39517715fce5)
Outputs                   : {[createImage.ImageId,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
Parameters                : {[AutomationAssumeRole,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]], [InstanceIamRole,

 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]], [SourceAmiId,

 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
StepExecutions            : {launchInstance, updateOSSoftware, stopInstance,
 createImage...}
```

Exemplo 2: esse exemplo lista os detalhes da etapa para o ID de execução do Automation fornecido

```
Get-SSMAutomationExecution -AutomationExecutionId e1d2bad3-4567-8901-
ae23-456c7c8901be | Select-Object -ExpandProperty StepExecutions | Select-Object
 StepName, Action, StepStatus, ValidNextSteps
```

Saída:

```
StepName                  Action                  StepStatus ValidNextSteps
--------                  ------                  ---------- --------------
LaunchInstance            aws:runInstances        Success
 {OSCompatibilityCheck}
OSCompatibilityCheck      aws:runCommand          Success    {RunPreUpdateScript}
RunPreUpdateScript        aws:runCommand          Success    {UpdateEC2Config}
UpdateEC2Config           aws:runCommand          Cancelled  {}
UpdateSSMAgent            aws:runCommand          Pending    {}
UpdateAWSPVDriver         aws:runCommand          Pending    {}
UpdateAWSEnaNetworkDriver aws:runCommand          Pending    {}
UpdateAWSNVMe             aws:runCommand          Pending    {}
InstallWindowsUpdates     aws:runCommand          Pending    {}
RunPostUpdateScript       aws:runCommand          Pending    {}
RunSysprepGeneralize      aws:runCommand          Pending    {}
StopInstance              aws:changeInstanceState Pending    {}
CreateImage               aws:createImage         Pending    {}
```

```
TerminateInstance         aws:changeInstanceState Pending    {}
```

- Para obter detalhes da API, consulte GetAutomationExecution na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo exibe os detalhes de uma execução do Automation.

```
Get-SSMAutomationExecution -AutomationExecutionId "4105a4fc-
f944-11e6-9d32-8fb2db27a909"
```

Saída:

```
AutomationExecutionId     : 4105a4fc-f944-11e6-9d32-8fb2db27a909
AutomationExecutionStatus : Failed
DocumentName              : AWS-UpdateLinuxAmi
DocumentVersion           : 1
ExecutionEndTime          : 2/22/2017 9:17:08 PM
ExecutionStartTime        : 2/22/2017 9:17:02 PM
FailureMessage            : Step launchInstance failed maximum allowed times. You
 are not authorized to perform this operation. Encoded
                            authorization failure message:
 B_V2QyyN7NhSZQYpmVzpEc4oSnj2GLTNYnXUHsTbqJkNMoDgubmbtthLmZyaiUYekORIrA42-
fv1x-04q5Fjff6glh
                            Yb6TI5b0GQeeNrpwNvpDzmO-
PSR1swlAbg9fdM9BcNjyrznspUkWpuKu9EC1Ou6v3OXU1KC9nZ7mPlWMFZNkSioQqpwWEvMw-
GZktsQzm67qOhUhBNOLWYhbS
                            pkfiqzY-5nw3S0obx30fhd3EJa5O_-
GjV_a0nFXQJa70ik40bFOrEh3MtCSbrQT6--DvFy_FQ8TKvkIXadyVskeJI84XOF5WmA60f1pi5GI08i-
nRfZS6oDeU

 gELBjjoFKD8s3L2aI0B6umWVxnQOjqhQRxwJ53b54sZJ2PW3v_mtg9-q0CK0ezS3xfh_y0ilaUGOAZG-
xjQFuvU_JZedWpla3xi-MZsmblAifBI
                            (Service: AmazonEC2; Status Code: 403; Error Code:
 UnauthorizedOperation; Request ID:
                            6a002f94-ba37-43fd-99e6-39517715fce5)
Outputs                   : {[createImage.ImageId,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
Parameters                : {[AutomationAssumeRole,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]], [InstanceIamRole,

 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]], [SourceAmiId,
```

```
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
StepExecutions            : {launchInstance, updateOSSoftware, stopInstance,
 createImage...}
```

Exemplo 2: esse exemplo lista os detalhes da etapa para o ID de execução do Automation fornecido

```
Get-SSMAutomationExecution -AutomationExecutionId e1d2bad3-4567-8901-
ae23-456c7c8901be | Select-Object -ExpandProperty StepExecutions | Select-Object
 StepName, Action, StepStatus, ValidNextSteps
```

Saída:

```
StepName                  Action                  StepStatus ValidNextSteps
--------                  ------                  ---------- --------------
LaunchInstance            aws:runInstances        Success
 {OSCompatibilityCheck}
OSCompatibilityCheck      aws:runCommand          Success    {RunPreUpdateScript}
RunPreUpdateScript        aws:runCommand          Success    {UpdateEC2Config}
UpdateEC2Config           aws:runCommand          Cancelled  {}
UpdateSSMAgent            aws:runCommand          Pending    {}
UpdateAWSPVDriver         aws:runCommand          Pending    {}
UpdateAWSEnaNetworkDriver aws:runCommand          Pending    {}
UpdateAWSNVMe             aws:runCommand          Pending    {}
InstallWindowsUpdates     aws:runCommand          Pending    {}
RunPostUpdateScript       aws:runCommand          Pending    {}
RunSysprepGeneralize      aws:runCommand          Pending    {}
StopInstance              aws:changeInstanceState Pending    {}
CreateImage               aws:createImage         Pending    {}
TerminateInstance         aws:changeInstanceState Pending    {}
```

- Para obter detalhes da API, consulte GetAutomationExecution na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`GetCommandInvocation`** com uma CLI

Os exemplos de código a seguir mostram como usar o `GetCommandInvocation`.

CLI

AWS CLI

Para exibir os detalhes de uma invocação de comando

O exemplo de `get-command-invocation` a seguir lista todas as invocações do comando especificado na instância especificada.

```
aws ssm get-command-invocation \
    --command-id "ef7fdfd8-9b57-4151-a15c-db9a12345678" \
    --instance-id "i-1234567890abcdef0"
```

Saída:

```
{
    "CommandId": "ef7fdfd8-9b57-4151-a15c-db9a12345678",
    "InstanceId": "i-1234567890abcdef0",
    "Comment": "b48291dd-ba76-43e0-b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-
d6ce8EXAMPLE",
    "DocumentName": "AWS-UpdateSSMAgent",
    "DocumentVersion": "",
    "PluginName": "aws:updateSsmAgent",
    "ResponseCode": 0,
    "ExecutionStartDateTime": "2020-02-19T18:18:03.419Z",
    "ExecutionElapsedTime": "PT0.091S",
    "ExecutionEndDateTime": "2020-02-19T18:18:03.419Z",
    "Status": "Success",
    "StatusDetails": "Success",
    "StandardOutputContent": "Updating amazon-ssm-agent from 2.3.842.0 to latest
\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-
east-2/ssm-agent-manifest.json\namazon-ssm-agent 2.3.842.0 has already been
 installed, update skipped\n",
    "StandardOutputUrl": "",
    "StandardErrorContent": "",
    "StandardErrorUrl": "",
    "CloudWatchOutputConfig": {
        "CloudWatchLogGroupName": "",
        "CloudWatchOutputEnabled": false
```

```
    }
}
```

Para obter mais informações, consulte Entender os status dos comandos no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte GetCommandInvocation na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo exibe os detalhes de um comando executado em uma instância.

```
Get-SSMCommandInvocationDetail -InstanceId "i-0cb2b964d3e14fd9f" -CommandId
 "b8eac879-0541-439d-94ec-47a80d554f44"
```

Saída:

```
CommandId              : b8eac879-0541-439d-94ec-47a80d554f44
Comment                : IP config
DocumentName           : AWS-RunShellScript
ExecutionElapsedTime   : PT0.004S
ExecutionEndDateTime   : 2017-02-22T20:13:16.651Z
ExecutionStartDateTime : 2017-02-22T20:13:16.651Z
InstanceId             : i-0cb2b964d3e14fd9f
PluginName             : aws:runShellScript
ResponseCode           : 0
StandardErrorContent   :
StandardErrorUrl       :
StandardOutputContent  :
StandardOutputUrl      :
Status                 : Success
StatusDetails          : Success
```

- Para obter detalhes da API, consulte GetCommandInvocation na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo exibe os detalhes de um comando executado em uma instância.

```
Get-SSMCommandInvocationDetail -InstanceId "i-0cb2b964d3e14fd9f" -CommandId
 "b8eac879-0541-439d-94ec-47a80d554f44"
```

Saída:

```
CommandId              : b8eac879-0541-439d-94ec-47a80d554f44
Comment                : IP config
DocumentName           : AWS-RunShellScript
ExecutionElapsedTime   : PT0.004S
ExecutionEndDateTime   : 2017-02-22T20:13:16.651Z
ExecutionStartDateTime : 2017-02-22T20:13:16.651Z
InstanceId             : i-0cb2b964d3e14fd9f
PluginName             : aws:runShellScript
ResponseCode           : 0
StandardErrorContent   :
StandardErrorUrl       :
StandardOutputContent  :
StandardOutputUrl      :
Status                 : Success
StatusDetails          : Success
```

- Para obter detalhes da API, consulte GetCommandInvocation na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`GetConnectionStatus`** com uma CLI

Os exemplos de código a seguir mostram como usar o `GetConnectionStatus`.

CLI

AWS CLI

Para exibir o status da conexão de uma instância gerenciada

Este exemplo de `get-connection-status` retorna o status da conexão da instância gerenciada especificada.

```
aws ssm get-connection-status \
    --target i-1234567890abcdef0
```

Saída:

```
{
    "Target": "i-1234567890abcdef0",
    "Status": "connected"
}
```

- Para obter detalhes da API, consulte GetConnectionStatus na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo recupera o status de conexão do Gerenciador de Sessões de uma instância para determinar se ela está conectada e pronta para receber conexões do Gerenciador de Sessões.

```
Get-SSMConnectionStatus -Target i-0a1caf234f12d3dc4
```

Saída:

```
Status    Target
------    ------
Connected i-0a1caf234f12d3dc4
```

- Para obter detalhes da API, consulte GetConnectionStatus na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo recupera o status de conexão do Gerenciador de Sessões de uma instância para determinar se ela está conectada e pronta para receber conexões do Gerenciador de Sessões.

```
Get-SSMConnectionStatus -Target i-0a1caf234f12d3dc4
```

Saída:

```
Status    Target
------    ------
Connected i-0a1caf234f12d3dc4
```

- Para obter detalhes da API, consulte GetConnectionStatus na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`GetDefaultPatchBaseline`** com uma CLI

Os exemplos de código a seguir mostram como usar o `GetDefaultPatchBaseline`.

CLI

AWS CLI

Exemplo 1: para exibir a lista de referência de patches padrão do Windows

O exemplo de `get-default-patch-baseline` a seguir recupera detalhes da lista de referência de patches padrão para o Windows Server.

```
aws ssm get-default-patch-baseline
```

Saída:

```
{
  "BaselineId": "pb-0713accee01612345",
  "OperatingSystem": "WINDOWS"
}
```

Exemplo 2: para exibir a lista de referência de patches padrão do Amazon Linux

O exemplo de `get-default-patch-baseline` a seguir recupera detalhes da lista de referência de patches padrão para o Amazon Linux.

```
aws ssm get-default-patch-baseline \
    --operating-system AMAZON_LINUX
```

Saída:

```
{
    "BaselineId": "pb-047c6eb9c8fc12345",
    "OperatingSystem": "AMAZON_LINUX"
}
```

Para obter mais informações, consulte Sobre listas de referência de patches predefinidas e personalizadas <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-baselines.html>__ e Definir uma lista de referência de patches existente como padrão no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte GetDefaultPatchBaseline na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo exibe a lista de referência de patches padrão.

```
Get-SSMDefaultPatchBaseline
```

Saída:

```
arn:aws:ssm:us-west-2:123456789012:patchbaseline/pb-04fb4ae6142167966
```

- Para obter detalhes da API, consulte GetDefaultPatchBaseline na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo exibe a lista de referência de patches padrão.

```
Get-SSMDefaultPatchBaseline
```

Saída:

```
arn:aws:ssm:us-west-2:123456789012:patchbaseline/pb-04fb4ae6142167966
```

- Para obter detalhes da API, consulte GetDefaultPatchBaseline na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`GetDeployablePatchSnapshotForInstance`** com uma CLI

Os exemplos de código a seguir mostram como usar o `GetDeployablePatchSnapshotForInstance`.

CLI

AWS CLI

Para recuperar o instantâneo atual da lista de referência de patches usado por uma instância

O exemplo de `get-deployable-patch-snapshot-for-instance` a seguir recupera detalhes do instantâneo atual da lista de referência de patches especificada usada por uma instância. Esse comando deve ser executado da instância usando as credenciais da instância. Para garantir que ele use as credenciais da instância, execute `aws configure` e especifique somente a região da sua instância. Deixe os campos `Access Key` e `Secret Key` vazios.

Dica: use `uuidgen` para gerar um `snapshot-id`.

```
aws ssm get-deployable-patch-snapshot-for-instance \
    --instance-id "i-1234567890abcdef0" \
    --snapshot-id "521c3536-930c-4aa9-950e-01234567abcd"
```

Saída:

```
{
    "InstanceId": "i-1234567890abcdef0",
    "SnapshotId": "521c3536-930c-4aa9-950e-01234567abcd",
    "Product": "AmazonLinux2018.03",
```

```
    "SnapshotDownloadUrl": "https://patch-baseline-snapshot-us-
east-1.s3.amazonaws.com/
ed85194ef27214f5984f28b4d664d14f7313568fea7d4b6ac6c10ad1f729d7e7-773304212436/
AMAZON_LINUX-521c3536-930c-4aa9-950e-01234567abcd?X-Amz-
Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20190215T164031Z&X-Amz-
SignedHeaders=host&X-Amz-Expires=86400&X-Amz-Credential=AKIAJ5C56P35AEBRX2QQ
%2F20190215%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-
Signature=efaaaf6e3878e77f48a6697e015efdbda9c426b09c5822055075c062f6ad2149"
}
```

Para obter mais informações, consulte Nome do parâmetro: ID do instantâneo no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte GetDeployablePatchSnapshotForInstance na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo exibe o instantâneo atual da lista de referência de patches usada por uma instância. Esse comando deve ser executado da instância usando as credenciais da instância. Para garantir que use as credenciais da instância, o exemplo passa um objeto **`Amazon.Runtime.InstanceProfileAWSCredentials`** para o parâmetro Credentials.

```
$credentials = [Amazon.Runtime.InstanceProfileAWSCredentials]::new()
Get-SSMDeployablePatchSnapshotForInstance -SnapshotId "4681775b-098f-4435-
a956-0ef33373ac11" -InstanceId "i-0cb2b964d3e14fd9f" -Credentials $credentials
```

Saída:

```
InstanceId          SnapshotDownloadUrl
----------          -------------------
i-0cb2b964d3e14fd9f https://patch-baseline-snapshot-us-west-2.s3-us-
west-2.amazonaws.com/853d0d3db0f0cafe...1692/4681775b-098f-4435...
```

Exemplo 2: esse exemplo mostra como obter o SnapshotDownloadUrl completo. Esse comando deve ser executado da instância usando as credenciais da instância. Para garantir que ele use as credenciais da instância, o exemplo configura a sessão do PowerShell para usar um objeto **`Amazon.Runtime.InstanceProfileAWSCredentials`**.

```
Set-AWSCredential -Credential
 ([Amazon.Runtime.InstanceProfileAWSCredentials]::new())
(Get-SSMDeployablePatchSnapshotForInstance -SnapshotId "4681775b-098f-4435-
a956-0ef33373ac11" -InstanceId "i-0cb2b964d3e14fd9f").SnapshotDownloadUrl
```

Saída:

```
https://patch-baseline-snapshot-us-west-2.s3-us-
west-2.amazonaws.com/853d0d3db0f0cafe...
```

- Para obter detalhes da API, consulte GetDeployablePatchSnapshotForInstance na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo exibe o instantâneo atual da lista de referência de patches usada por uma instância. Esse comando deve ser executado da instância usando as credenciais da instância. Para garantir que use as credenciais da instância, o exemplo passa um objeto **`Amazon.Runtime.InstanceProfileAWSCredentials`** para o parâmetro Credentials.

```
$credentials = [Amazon.Runtime.InstanceProfileAWSCredentials]::new()
Get-SSMDeployablePatchSnapshotForInstance -SnapshotId "4681775b-098f-4435-
a956-0ef33373ac11" -InstanceId "i-0cb2b964d3e14fd9f" -Credentials $credentials
```

Saída:

```
InstanceId          SnapshotDownloadUrl
----------          -------------------
i-0cb2b964d3e14fd9f https://patch-baseline-snapshot-us-west-2.s3-us-
west-2.amazonaws.com/853d0d3db0f0cafe...1692/4681775b-098f-4435...
```

Exemplo 2: esse exemplo mostra como obter o SnapshotDownloadUrl completo. Esse comando deve ser executado da instância usando as credenciais da instância. Para garantir que ele use as credenciais da instância, o exemplo configura a sessão do PowerShell para usar um objeto **`Amazon.Runtime.InstanceProfileAWSCredentials`**.

```
Set-AWSCredential -Credential
 ([Amazon.Runtime.InstanceProfileAWSCredentials]::new())
(Get-SSMDeployablePatchSnapshotForInstance -SnapshotId "4681775b-098f-4435-
a956-0ef33373ac11" -InstanceId "i-0cb2b964d3e14fd9f").SnapshotDownloadUrl
```

Saída:

```
https://patch-baseline-snapshot-us-west-2.s3-us-
west-2.amazonaws.com/853d0d3db0f0cafe...
```

- Para obter detalhes da API, consulte GetDeployablePatchSnapshotForInstance na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`GetDocument`** com uma CLI

Os exemplos de código a seguir mostram como usar o `GetDocument`.

CLI

AWS CLI

Para obter conteúdo do documento

O exemplo de `get-document` a seguir exibe o conteúdo de um documento do Systems Manager.

```
aws ssm get-document \
    --name "AWS-RunShellScript"
```

Saída:

```
{
    "Name": "AWS-RunShellScript",
    "DocumentVersion": "1",
    "Status": "Active",
    "Content": "{\n    \"schemaVersion\":\"1.2\",\n    \"description\":\"Run
 a shell script or specify the commands to run.\",\n    \"parameters\":{\n
        \"commands\":{\n            \"type\":\"StringList\",\n
 \"description\":\"(Required) Specify a shell script or a command to run.\",
\n            \"minItems\":1,\n            \"displayType\":\"textarea\"\n
      },\n        \"workingDirectory\":{\n            \"type\":\"String\",\n
```

```
            \"default\":\"\",\n            \"description\":\"(Optional) The
 path to the working directory on your instance.\",\n            \"maxChars
\":4096\n        },\n        \"executionTimeout\":{\n            \"type\":
\"String\",\n            \"default\":\"3600\",\n            \"description
\":\"(Optional) The time in seconds for a command to complete before it is
 considered to have failed. Default is 3600 (1 hour). Maximum is 172800 (48
 hours).\",\n            \"allowedPattern\":\"([1-9][0-9]{0,4})|(1[0-6][0-9]
{4})|(17[0-1][0-9]{3})|(172[0-7][0-9]{2})|(172800)\"\n        }\n    },\n
 \"runtimeConfig\":{\n        \"aws:runShellScript\":{\n            \"properties
\":[\n                {\n                    \"id\":\"0.aws:runShellScript
\",\n                    \"runCommand\":\"{{ commands }}\",\n
      \"workingDirectory\":\"{{ workingDirectory }}\",\n
 \"timeoutSeconds\":\"{{ executionTimeout }}\"\n                }\n
  ]\n        }\n    }\n}\n",
    "DocumentType": "Command",
    "DocumentFormat": "JSON"
}
```

Para obter mais informações, consulte Documentos do AWS Systems Manager no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte GetDocument na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo retorna o conteúdo de um documento.

```
Get-SSMDocument -Name "RunShellScript"
```

Saída:

```
Content
-------
{...
```

Exemplo 2: esse exemplo exibe o conteúdo completo de um documento.

```
(Get-SSMDocument -Name "RunShellScript").Content
```

```
{
   "schemaVersion":"2.0",
   "description":"Run an updated script",
   "parameters":{
      "commands":{
         "type":"StringList",
         "description":"(Required) Specify a shell script or a command to run.",
         "minItems":1,
         "displayType":"textarea"
      }
   },
   "mainSteps":[
      {
         "action":"aws:runShellScript",
         "name":"runShellScript",
         "inputs":{
            "commands":"{{ commands }}"
         }
      },
      {
         "action":"aws:runPowerShellScript",
         "name":"runPowerShellScript",
         "inputs":{
            "commands":"{{ commands }}"
         }
      }
   ]
}
```

- Para obter detalhes da API, consulte GetDocument na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo retorna o conteúdo de um documento.

```
Get-SSMDocument -Name "RunShellScript"
```

Saída:

```
Content
-------
{...
```

Exemplo 2: esse exemplo exibe o conteúdo completo de um documento.

```
(Get-SSMDocument -Name "RunShellScript").Content
{
   "schemaVersion":"2.0",
   "description":"Run an updated script",
   "parameters":{
      "commands":{
         "type":"StringList",
         "description":"(Required) Specify a shell script or a command to run.",
         "minItems":1,
         "displayType":"textarea"
      }
   },
   "mainSteps":[
      {
         "action":"aws:runShellScript",
         "name":"runShellScript",
         "inputs":{
            "commands":"{{ commands }}"
         }
      },
      {
         "action":"aws:runPowerShellScript",
         "name":"runPowerShellScript",
         "inputs":{
            "commands":"{{ commands }}"
         }
      }
   ]
}
```

- Para obter detalhes da API, consulte GetDocument na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`GetInventory`** com uma CLI

Os exemplos de código a seguir mostram como usar o `GetInventory`.

CLI

AWS CLI

Para visualizar o inventário

Este exemplo obtém os metadados personalizados do seu inventário.

Comando:

```
aws ssm get-inventory
```

Saída:

```
{
  "Entities": [
      {
          "Data": {
              "AWS:InstanceInformation": {
                  "Content": [
                      {
                          "ComputerName": "ip-172-31-44-222.us-
west-2.compute.internal",
                          "InstanceId": "i-0cb2b964d3e14fd9f",
                          "IpAddress": "172.31.44.222",
                          "AgentType": "amazon-ssm-agent",
                          "ResourceType": "EC2Instance",
                          "AgentVersion": "2.0.672.0",
                          "PlatformVersion": "2016.09",
                          "PlatformName": "Amazon Linux AMI",
                          "PlatformType": "Linux"
                      }
                  ],
                  "TypeName": "AWS:InstanceInformation",
                  "SchemaVersion": "1.0",
                  "CaptureTime": "2017-02-20T18:03:58Z"
              }
          },
          "Id": "i-0cb2b964d3e14fd9f"
      }
  ]
}
```

- Para obter detalhes da API, consulte GetInventory na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo obtém os metadados personalizados do seu inventário.

```
Get-SSMInventory
```

Saída:

```
Data
     Id
----
     --
{[AWS:InstanceInformation,
 Amazon.SimpleSystemsManagement.Model.InventoryResultItem]} i-0cb2b964d3e14fd9f
```

- Para obter detalhes da API, consulte GetInventory na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo obtém os metadados personalizados do seu inventário.

```
Get-SSMInventory
```

Saída:

```
Data
     Id
----
     --
{[AWS:InstanceInformation,
 Amazon.SimpleSystemsManagement.Model.InventoryResultItem]} i-0cb2b964d3e14fd9f
```

- Para obter detalhes da API, consulte GetInventory na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`GetInventorySchema`** com uma CLI

Os exemplos de código a seguir mostram como usar o `GetInventorySchema`.

CLI

AWS CLI

Para visualizar seu esquema de inventário

Este exemplo retorna uma lista de nomes de tipos de inventário para a conta.

Comando:

```
aws ssm get-inventory-schema
```

Saída:

```
{
  "Schemas": [
      {
          "TypeName": "AWS:AWSComponent",
          "Version": "1.0",
          "Attributes": [
              {
                  "Name": "Name",
                  "DataType": "STRING"
              },
              {
                  "Name": "ApplicationType",
                  "DataType": "STRING"
              },
              {
                  "Name": "Publisher",
                  "DataType": "STRING"
              },
              {
                  "Name": "Version",
                  "DataType": "STRING"
```

```
                },
                {
                    "Name": "InstalledTime",
                    "DataType": "STRING"
                },
                {
                    "Name": "Architecture",
                    "DataType": "STRING"
                },
                {
                    "Name": "URL",
                    "DataType": "STRING"
                }
            ]
        },
        ...
    ],
    "NextToken": "--token string truncated--"
}
```

Para visualizar o esquema de inventário de um tipo de inventário específico

Este exemplo retorna o esquema de inventário para o tipo de inventário AWS:AWSComponent.

Comando:

```
aws ssm get-inventory-schema --type-name "AWS:AWSComponent"
```

- Para obter detalhes da API, consulte GetInventorySchema na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo retorna uma lista de nomes de tipos de inventário para a conta.

```
Get-SSMInventorySchema
```

- Para obter detalhes da API, consulte GetInventorySchema na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo retorna uma lista de nomes de tipos de inventário para a conta.

```
Get-SSMInventorySchema
```

- Para obter detalhes da API, consulte GetInventorySchema na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`GetMaintenanceWindow`** com uma CLI

Os exemplos de código a seguir mostram como usar o `GetMaintenanceWindow`.

CLI

AWS CLI

Para obter informações sobre uma janela de manutenção

O exemplo de `get-maintenance-window` a seguir recupera detalhes sobre a janela de manutenção especificada.

```
aws ssm get-maintenance-window \
    --window-id "mw-03eb9db428EXAMPLE"
```

Saída:

```
{
    "AllowUnassociatedTargets": true,
    "CreatedDate": 1515006912.957,
    "Cutoff": 1,
    "Duration": 6,
    "Enabled": true,
    "ModifiedDate": 2020-01-01T10:04:04.099Z,
    "Name": "My-Maintenance-Window",
    "Schedule": "rate(3 days)",
    "WindowId": "mw-03eb9db428EXAMPLE",
    "NextExecutionTime": "2020-02-25T00:08:15.099Z"
```

```
}
```

Para obter mais informações, consulte Visualizar informações sobre janelas de manutenção (AWS CLI) no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte GetMaintenanceWindow na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo obtém detalhes sobre uma janela de manutenção.

```
Get-SSMMaintenanceWindow -WindowId "mw-03eb9db42890fb82d"
```

Saída:

```
AllowUnassociatedTargets : False
CreatedDate              : 2/20/2017 6:14:05 PM
Cutoff                   : 1
Duration                 : 2
Enabled                  : True
ModifiedDate             : 2/20/2017 6:14:05 PM
Name                     : TestMaintWin
Schedule                 : cron(0 */30 * * * ? *)
WindowId                 : mw-03eb9db42890fb82d
```

- Para obter detalhes da API, consulte GetMaintenanceWindow na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo obtém detalhes sobre uma janela de manutenção.

```
Get-SSMMaintenanceWindow -WindowId "mw-03eb9db42890fb82d"
```

Saída:

```
AllowUnassociatedTargets : False
CreatedDate              : 2/20/2017 6:14:05 PM
Cutoff                   : 1
```

```
Duration                : 2
Enabled                 : True
ModifiedDate            : 2/20/2017 6:14:05 PM
Name                    : TestMaintWin
Schedule                : cron(0 */30 * * * ? *)
WindowId                : mw-03eb9db42890fb82d
```

- Para obter detalhes da API, consulte GetMaintenanceWindow na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`GetMaintenanceWindowExecution`** com uma CLI

Os exemplos de código a seguir mostram como usar o `GetMaintenanceWindowExecution`.

CLI

AWS CLI

Para obter informações sobre a execução de uma tarefa da janela de manutenção

O exemplo de `get-maintenance-window-execution` a seguir lista informações sobre uma tarefa que é executada como parte da execução da janela de manutenção especificada.

```
aws ssm get-maintenance-window-execution \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE"
```

Saída:

```
{
    "Status": "SUCCESS",
    "TaskIds": [
        "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"
    ],
    "StartTime": 1487692834.595,
    "EndTime": 1487692835.051,
    "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2EXAMPLE",
}
```

Para obter mais informações, consulte Visualizar informações sobre tarefas e execuções de tarefas (AWS CLI) no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte GetMaintenanceWindowExecution na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo lista informações sobre uma tarefa que foi executada como parte da execução de uma janela de manutenção.

```
Get-SSMMaintenanceWindowExecution -WindowExecutionId "518d5565-5969-4cca-8f0e-
da3b2a638355"
```

Saída:

```
EndTime           : 2/21/2017 4:00:35 PM
StartTime         : 2/21/2017 4:00:34 PM
Status            : FAILED
StatusDetails     : One or more tasks in the orchestration failed.
TaskIds           : {ac0c6ae1-daa3-4a89-832e-d384503b6586}
WindowExecutionId : 518d5565-5969-4cca-8f0e-da3b2a638355
```

- Para obter detalhes da API, consulte GetMaintenanceWindowExecution na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo lista informações sobre uma tarefa que foi executada como parte da execução de uma janela de manutenção.

```
Get-SSMMaintenanceWindowExecution -WindowExecutionId "518d5565-5969-4cca-8f0e-
da3b2a638355"
```

Saída:

```
EndTime           : 2/21/2017 4:00:35 PM
StartTime         : 2/21/2017 4:00:34 PM
Status            : FAILED
StatusDetails     : One or more tasks in the orchestration failed.
```

```
TaskIds           : {ac0c6ae1-daa3-4a89-832e-d384503b6586}
WindowExecutionId : 518d5565-5969-4cca-8f0e-da3b2a638355
```

- Para obter detalhes da API, consulte GetMaintenanceWindowExecution na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`GetMaintenanceWindowExecutionTask`** com uma CLI

Os exemplos de código a seguir mostram como usar o `GetMaintenanceWindowExecutionTask`.

CLI

AWS CLI

Para obter informações sobre a execução de uma tarefa da janela de manutenção

O exemplo de `get-maintenance-window-execution-task` a seguir lista informações sobre uma tarefa que faz parte da execução da janela de manutenção especificada.

```
aws ssm get-maintenance-window-execution-task \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE" \
    --task-id "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"
```

Saída:

```
{
    "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2EXAMPLE",
    "TaskExecutionId": "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE",
    "TaskArn": "AWS-RunPatchBaseline",
    "ServiceRole": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
    "Type": "RUN_COMMAND",
    "TaskParameters": [
        {
            "BaselineOverride": {
                "Values": [
                    ""
```

```
                ]
            },
            "InstallOverrideList": {
                "Values": [
                    ""
                ]
            },
            "Operation": {
                "Values": [
                    "Scan"
                ]
            },
            "RebootOption": {
                "Values": [
                    "RebootIfNeeded"
                ]
            },
            "SnapshotId": {
                "Values": [
                    "{{ aws:ORCHESTRATION_ID }}"
                ]
            },
            "aws:InstanceId": {
                "Values": [
                    "i-02573cafcfEXAMPLE",
                    "i-0471e04240EXAMPLE",
                    "i-07782c72faEXAMPLE"
                ]
            }
        }
    ],
    "Priority": 1,
    "MaxConcurrency": "1",
    "MaxErrors": "3",
    "Status": "SUCCESS",
    "StartTime": "2021-08-04T11:45:35.088000-07:00",
    "EndTime": "2021-08-04T11:53:09.079000-07:00"
}
```

Para obter mais informações, consulte Visualizar informações sobre tarefas e execuções de tarefas (AWS CLI) no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte GetMaintenanceWindowExecutionTask na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo lista informações sobre uma tarefa que fazia parte da execução de uma janela de manutenção.

```
Get-SSMMaintenanceWindowExecutionTask -TaskId "ac0c6ae1-daa3-4a89-832e-
d384503b6586" -WindowExecutionId "518d5565-5969-4cca-8f0e-da3b2a638355"
```

Saída:

```
EndTime           : 2/21/2017 4:00:35 PM
MaxConcurrency    : 1
MaxErrors         : 1
Priority          : 10
ServiceRole       : arn:aws:iam::123456789012:role/MaintenanceWindowsRole
StartTime         : 2/21/2017 4:00:34 PM
Status            : FAILED
StatusDetails     : The maximum error count was exceeded.
TaskArn           : AWS-RunShellScript
TaskExecutionId   : ac0c6ae1-daa3-4a89-832e-d384503b6586
TaskParameters    :
 {Amazon.Runtime.Internal.Util.AlwaysSendDictionary`2[System.String,Amazon.SimpleSystemsM
                   meterValueExpression]}
Type              : RUN_COMMAND
WindowExecutionId : 518d5565-5969-4cca-8f0e-da3b2a638355
```

- Para obter detalhes da API, consulte GetMaintenanceWindowExecutionTask na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo lista informações sobre uma tarefa que fazia parte da execução de uma janela de manutenção.

```
Get-SSMMaintenanceWindowExecutionTask -TaskId "ac0c6ae1-daa3-4a89-832e-
d384503b6586" -WindowExecutionId "518d5565-5969-4cca-8f0e-da3b2a638355"
```

Saída:

```
EndTime           : 2/21/2017 4:00:35 PM
MaxConcurrency    : 1
```

```
MaxErrors         : 1
Priority          : 10
ServiceRole       : arn:aws:iam::123456789012:role/MaintenanceWindowsRole
StartTime         : 2/21/2017 4:00:34 PM
Status            : FAILED
StatusDetails     : The maximum error count was exceeded.
TaskArn           : AWS-RunShellScript
TaskExecutionId   : ac0c6ae1-daa3-4a89-832e-d384503b6586
TaskParameters    :
 {Amazon.Runtime.Internal.Util.AlwaysSendDictionary`2[System.String,Amazon.SimpleSystemsM
                   meterValueExpression]}
Type              : RUN_COMMAND
WindowExecutionId : 518d5565-5969-4cca-8f0e-da3b2a638355
```

- Para obter detalhes da API, consulte GetMaintenanceWindowExecutionTask na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`GetParameter`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `GetParameter`.

CLI

AWS CLI

Exemplo 1: exibir o valor de um parâmetro

O exemplo de `get-parameter` a seguir lista o valor do parâmetro único especificado.

```
aws ssm get-parameter \
    --name "MyStringParameter"
```

Saída:

```
{
    "Parameter": {
        "Name": "MyStringParameter",
        "Type": "String",
```

```
        "Value": "Veni",
        "Version": 1,
        "LastModifiedDate": 1530018761.888,
        "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/MyStringParameter"
        "DataType": "text"
    }
}
```

Para obter mais informações, consulte Trabalhar com o Parameter Store no Guia do usuário do AWS Systems Manager.

Exemplo 2: descriptografar o valor de um parâmetro SecureString

O exemplo de `get-parameter` a seguir descriptografa o valor do parâmetro `SecureString` especificado.

```
aws ssm get-parameter \
    --name "MySecureStringParameter" \
    --with-decryption
```

Saída:

```
{
    "Parameter": {
        "Name": "MySecureStringParameter",
        "Type": "SecureString",
        "Value": "16679b88-310b-4895-a943-e0764EXAMPLE",
        "Version": 2,
        "LastModifiedDate": 1582155479.205,
        "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/
MySecureStringParameter"
        "DataType": "text"
    }
}
```

Para obter mais informações, consulte Trabalhar com o Parameter Store no Guia do usuário do AWS Systems Manager.

Exemplo 3: exibir o valor de um parâmetro usando rótulos

O exemplo de `get-parameter` a seguir lista o valores do parâmetros especificado com um rótulo especificado.

```
aws ssm get-parameter \
    --name "MyParameter:label"
```

Saída:

```
{
    "Parameter": {
        "Name": "MyParameter",
        "Type": "String",
        "Value": "parameter version 2",
        "Version": 2,
        "Selector": ":label",
        "LastModifiedDate": "2021-07-12T09:49:15.865000-07:00",
        "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/MyParameter",
        "DataType": "text"
    }
}
```

Para obter mais informações, consulte Trabalhar com rótulos de parâmetros no Guia do usuário do AWS Systems Manager.

Exemplo 4: exibir o valor de um parâmetro usando versões

O exemplo de `get-parameter` a seguir lista o valor da versão do parâmetro único especificado.

```
aws ssm get-parameter \
    --name "MyParameter:2"
```

Saída:

```
{
    "Parameter": {
        "Name": "MyParameter",
        "Type": "String",
        "Value": "parameter version 2",
        "Version": 2,
        "Selector": ":2",
        "LastModifiedDate": "2021-07-12T09:49:15.865000-07:00",
        "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/MyParameter",
        "DataType": "text"
```

```
    }
}
```

Para obter mais informações, consulte Trabalhar com rótulos de parâmetros no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte GetParameter na Referência de comandos da AWS CLI.

Rust

SDK para Rust

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
    pub async fn list_path(&self, path: &str) -> Result<Vec<Parameter>, EC2Error>
{
        let maybe_params: Vec<Result<Parameter, _>> = TryFlatMap::new(
            self.inner
                .get_parameters_by_path()
                .path(path)
                .into_paginator()
                .send(),
        )
        .flat_map(|item| item.parameters.unwrap_or_default())
        .collect()
        .await;
        // Fail on the first error
        let params = maybe_params
            .into_iter()
            .collect::<Result<Vec<Parameter>, _>>()?;
        Ok(params)
    }
```

- Para obter detalhes da API, consulte GetParameter na Referência de API do AWS SDK para Rust.

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`GetParameterHistory`** com uma CLI

Os exemplos de código a seguir mostram como usar o `GetParameterHistory`.

CLI

AWS CLI

Para obter o histórico de valores de um parâmetro

O exemplo de `get-parameter-history` a seguir lista o histórico de alterações do parâmetro especificado, incluindo seu valor.

```
aws ssm get-parameter-history \
    --name "MyStringParameter"
```

Saída:

```
{
    "Parameters": [
        {
            "Name": "MyStringParameter",
            "Type": "String",
            "LastModifiedDate": 1582154711.976,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
            "Description": "This is the first version of my String parameter",
            "Value": "Veni",
            "Version": 1,
            "Labels": [],
            "Tier": "Standard",
            "Policies": []
        },
        {
            "Name": "MyStringParameter",
            "Type": "String",
            "LastModifiedDate": 1582156093.471,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
            "Description": "This is the second version of my String parameter",
```

```
            "Value": "Vidi",
            "Version": 2,
            "Labels": [],
            "Tier": "Standard",
            "Policies": []
        },
        {
            "Name": "MyStringParameter",
            "Type": "String",
            "LastModifiedDate": 1582156117.545,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
            "Description": "This is the third version of my String parameter",
            "Value": "Vici",
            "Version": 3,
            "Labels": [],
            "Tier": "Standard",
            "Policies": []
        }
    ]
}
```

Para obter mais informações, consulte Trabalhar com versões de parâmetros no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte GetParameterHistory na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo lista o histórico de valores de um parâmetro.

```
Get-SSMParameterHistory -Name "Welcome"
```

Saída:

```
Description      :
KeyId            :
LastModifiedDate : 3/3/2017 6:55:25 PM
LastModifiedUser : arn:aws:iam::123456789012:user/admin
Name             : Welcome
```

```
Type             : String
Value            : helloWorld
```

- Para obter detalhes da API, consulte GetParameterHistory na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo lista o histórico de valores de um parâmetro.

```
Get-SSMParameterHistory -Name "Welcome"
```

Saída:

```
Description      :
KeyId            :
LastModifiedDate : 3/3/2017 6:55:25 PM
LastModifiedUser : arn:aws:iam::123456789012:user/admin
Name             : Welcome
Type             : String
Value            : helloWorld
```

- Para obter detalhes da API, consulte GetParameterHistory na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`GetParameters`** com uma CLI

Os exemplos de código a seguir mostram como usar o `GetParameters`.

CLI

AWS CLI

Exemplo 1: listar os valores de um parâmetro

O exemplo de `get-parameters` a seguir lista os valores dos três parâmetros especificados.

```
aws ssm get-parameters \
```

```
    --names "MyStringParameter" "MyStringListParameter" "MyInvalidParameterName"
```

Saída:

```
{
    "Parameters": [
        {
            "Name": "MyStringListParameter",
            "Type": "StringList",
            "Value": "alpha,beta,gamma",
            "Version": 1,
            "LastModifiedDate": 1582154764.222,
            "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/
MyStringListParameter"
            "DataType": "text"
        },
        {
            "Name": "MyStringParameter",
            "Type": "String",
            "Value": "Vici",
            "Version": 3,
            "LastModifiedDate": 1582156117.545,
            "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/
MyStringParameter"
            "DataType": "text"
        }
    ],
    "InvalidParameters": [
        "MyInvalidParameterName"
    ]
}
```

Para obter mais informações, consulte Trabalhar com o Parameter Store no Guia do usuário do AWS Systems Manager.

Exemplo 2: listar nomes e valores de vários parâmetros usando a opção "--query"

O exemplo de `get-parameters` a seguir lista os nomes e valores dos parâmetros especificados.

```
aws ssm get-parameters \
    --names MyStringParameter MyStringListParameter \
```

```
    --query "Parameters[*].{Name:Name,Value:Value}"
```

Saída:

```
[
    {
        "Name": "MyStringListParameter",
        "Value": "alpha,beta,gamma"
    },
    {
        "Name": "MyStringParameter",
        "Value": "Vidi"
    }
]
```

Para obter mais informações, consulte Trabalhar com o Parameter Store no Guia do usuário do AWS Systems Manager.

Exemplo 3: exibir o valor de um parâmetro usando rótulos

O exemplo de `get-parameter` a seguir lista o valores do parâmetros especificado com um rótulo especificado.

```
aws ssm get-parameter \
    --name "MyParameter:label"
```

Saída:

```
{
    "Parameters": [
        {
            "Name": "MyLabelParameter",
            "Type": "String",
            "Value": "parameter by label",
            "Version": 1,
            "Selector": ":label",
            "LastModifiedDate": "2021-07-12T09:49:15.865000-07:00",
            "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/MyParameter",
            "DataType": "text"
        },
        {
            "Name": "MyVersionParameter",
```

```
            "Type": "String",
            "Value": "parameter by version",
            "Version": 2,
            "Selector": ":2",
            "LastModifiedDate": "2021-03-24T16:20:28.236000-07:00",
            "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/unlabel-param",
            "DataType": "text"
        }
    ],
    "InvalidParameters": []
}
```

Para obter mais informações, consulte Trabalhar com rótulos de parâmetros no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte GetParameters na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo lista os valores de um parâmetro.

```
Get-SSMParameterValue -Name "Welcome"
```

Saída:

```
InvalidParameters Parameters
----------------- ----------
{}                {Welcome}
```

Exemplo 2: esse exemplo retorna os detalhes do valor.

```
(Get-SSMParameterValue -Name "Welcome").Parameters
```

Saída:

```
Name    Type   Value
----    ----   -----
Welcome String Good day, Sunshine!
```

- Para obter detalhes da API, consulte GetParameters na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo lista os valores de um parâmetro.

```
Get-SSMParameterValue -Name "Welcome"
```

Saída:

```
InvalidParameters Parameters
----------------- ----------
{}                {Welcome}
```

Exemplo 2: esse exemplo retorna os detalhes do valor.

```
(Get-SSMParameterValue -Name "Welcome").Parameters
```

Saída:

```
Name    Type   Value
----    ----   -----
Welcome String Good day, Sunshine!
```

- Para obter detalhes da API, consulte GetParameters na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`GetPatchBaseline`** com uma CLI

Os exemplos de código a seguir mostram como usar o `GetPatchBaseline`.

CLI

AWS CLI

Para exibir uma lista de referência de patches

O exemplo de `get-patch-baseline` a seguir recupera os detalhes da lista de referência de patches especificada.

```
aws ssm get-patch-baseline \
    --baseline-id "pb-0123456789abcdef0"
```

Saída:

```
{
    "BaselineId": "pb-0123456789abcdef0",
    "Name": "WindowsPatching",
    "OperatingSystem": "WINDOWS",
    "GlobalFilters": {
        "PatchFilters": []
    },
    "ApprovalRules": {
        "PatchRules": [
            {
                "PatchFilterGroup": {
                    "PatchFilters": [
                        {
                            "Key": "PRODUCT",
                            "Values": [
                                "WindowsServer2016"
                            ]
                        }
                    ]
                },
                "ComplianceLevel": "CRITICAL",
                "ApproveAfterDays": 0,
                "EnableNonSecurity": false
            }
        ]
    },
    "ApprovedPatches": [],
    "ApprovedPatchesComplianceLevel": "UNSPECIFIED",
    "ApprovedPatchesEnableNonSecurity": false,
    "RejectedPatches": [],
    "RejectedPatchesAction": "ALLOW_AS_DEPENDENCY",
    "PatchGroups": [
        "QA",
        "DEV"
    ],
```

```
    "CreatedDate": 1550244180.465,
    "ModifiedDate": 1550244180.465,
    "Description": "Patches for Windows Servers",
    "Sources": []
}
```

Para obter mais informações, consulte Sobre listas de referência de patches no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte GetPatchBaseline na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo exibe os detalhes de uma lista de referência de patches.

```
Get-SSMPatchBaselineDetail -BaselineId "pb-03da896ca3b68b639"
```

Saída:

```
ApprovalRules   : Amazon.SimpleSystemsManagement.Model.PatchRuleGroup
ApprovedPatches : {}
BaselineId      : pb-03da896ca3b68b639
CreatedDate     : 3/3/2017 5:02:19 PM
Description     : Baseline containing all updates approved for production systems
GlobalFilters   : Amazon.SimpleSystemsManagement.Model.PatchFilterGroup
ModifiedDate    : 3/3/2017 5:02:19 PM
Name            : Production-Baseline
PatchGroups     : {}
RejectedPatches : {}
```

- Para obter detalhes da API, consulte GetPatchBaseline na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo exibe os detalhes de uma lista de referência de patches.

```
Get-SSMPatchBaselineDetail -BaselineId "pb-03da896ca3b68b639"
```

Saída:

```
ApprovalRules   : Amazon.SimpleSystemsManagement.Model.PatchRuleGroup
ApprovedPatches : {}
BaselineId      : pb-03da896ca3b68b639
CreatedDate     : 3/3/2017 5:02:19 PM
Description     : Baseline containing all updates approved for production systems
GlobalFilters   : Amazon.SimpleSystemsManagement.Model.PatchFilterGroup
ModifiedDate    : 3/3/2017 5:02:19 PM
Name            : Production-Baseline
PatchGroups     : {}
RejectedPatches : {}
```

- Para obter detalhes da API, consulte GetPatchBaseline na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`GetPatchBaselineForPatchGroup`** com uma CLI

Os exemplos de código a seguir mostram como usar o `GetPatchBaselineForPatchGroup`.

CLI

AWS CLI

Para exibir uma lista de referência de patches de um grupo de patches

O exemplo de `get-patch-baseline-for-patch-group` a seguir recupera detalhes sobre a lista de referência de patches para a instância especificada.

```
aws ssm get-patch-baseline-for-patch-group \
    --patch-group "DEV"
```

Saída:

```
{
    "PatchGroup": "DEV",
    "BaselineId": "pb-0123456789abcdef0",
    "OperatingSystem": "WINDOWS"
```

```
}
```

Para obter mais informações, consulte Criar um grupo de patches <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html>__ e Adicionar um grupo de patches a uma lista de referência de patches no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte GetPatchBaselineForPatchGroup na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo exibe a lista de referência de patches para um grupo de patches.

```
Get-SSMPatchBaselineForPatchGroup -PatchGroup "Production"
```

Saída:

```
BaselineId           PatchGroup
----------           ----------
pb-045f10b4f382baeda Production
```

- Para obter detalhes da API, consulte GetPatchBaselineForPatchGroup na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo exibe a lista de referência de patches para um grupo de patches.

```
Get-SSMPatchBaselineForPatchGroup -PatchGroup "Production"
```

Saída:

```
BaselineId           PatchGroup
----------           ----------
pb-045f10b4f382baeda Production
```

- Para obter detalhes da API, consulte GetPatchBaselineForPatchGroup na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`ListAssociationVersions`** com uma CLI

Os exemplos de código a seguir mostram como usar o `ListAssociationVersions`.

CLI

AWS CLI

Para listar todas as versões de uma associação de um ID de associação específico

O exemplo de `list-association-versions` a seguir lista todas as versões das associações especificadas.

```
aws ssm list-association-versions \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Saída:

```
{
"AssociationVersions": [
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "CreatedDate": 1550505536.726,
            "Name": "AWS-UpdateSSMAgent",
            "Parameters": {
                "allowDowngrade": [
                    "false"
                ],
                "version": [
                    ""
                ]
            },
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-1234567890abcdef0"
                    ]
```

```
                }
            ],
            "ScheduleExpression": "cron(0 00 12 ? * SUN *)",
            "AssociationName": "UpdateSSMAgent"
        }
    ]
}
```

Para obter mais informações, consulte Trabalhar com associações no Systems Manager no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte ListAssociationVersions na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo recupera todas as versões da associação fornecida.

```
Get-SSMAssociationVersionList -AssociationId 123a45a0-c678-9012-3456-78901234db5e
```

Saída:

```
AssociationId      : 123a45a0-c678-9012-3456-78901234db5e
AssociationName    :
AssociationVersion : 2
ComplianceSeverity :
CreatedDate        : 3/12/2019 9:21:01 AM
DocumentVersion    :
MaxConcurrency     :
MaxErrors          :
Name               : AWS-GatherSoftwareInventory
OutputLocation     :
Parameters         : {}
ScheduleExpression :
Targets            : {InstanceIds}

AssociationId      : 123a45a0-c678-9012-3456-78901234db5e
AssociationName    : test-case-1234567890
AssociationVersion : 1
ComplianceSeverity :
```

```
CreatedDate        : 3/2/2019 8:53:29 AM
DocumentVersion    :
MaxConcurrency     :
MaxErrors          :
Name               : AWS-GatherSoftwareInventory
OutputLocation     :
Parameters         : {}
ScheduleExpression : rate(30minutes)
Targets            : {InstanceIds}
```

- Para obter detalhes da API, consulte ListAssociationVersions na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo recupera todas as versões da associação fornecida.

```
Get-SSMAssociationVersionList -AssociationId 123a45a0-c678-9012-3456-78901234db5e
```

Saída:

```
AssociationId      : 123a45a0-c678-9012-3456-78901234db5e
AssociationName    :
AssociationVersion : 2
ComplianceSeverity :
CreatedDate        : 3/12/2019 9:21:01 AM
DocumentVersion    :
MaxConcurrency     :
MaxErrors          :
Name               : AWS-GatherSoftwareInventory
OutputLocation     :
Parameters         : {}
ScheduleExpression :
Targets            : {InstanceIds}

AssociationId      : 123a45a0-c678-9012-3456-78901234db5e
AssociationName    : test-case-1234567890
AssociationVersion : 1
ComplianceSeverity :
CreatedDate        : 3/2/2019 8:53:29 AM
DocumentVersion    :
MaxConcurrency     :
MaxErrors          :
```

```
Name               : AWS-GatherSoftwareInventory
OutputLocation     :
Parameters         : {}
ScheduleExpression : rate(30minutes)
Targets            : {InstanceIds}
```

- Para obter detalhes da API, consulte ListAssociationVersions na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`ListAssociations`** com uma CLI

Os exemplos de código a seguir mostram como usar o `ListAssociations`.

CLI

AWS CLI

Exemplo 1: listar suas associações para uma instância específica

O exemplo de associações de lista a seguir lista todas as associações com AssociationName, UpdateSSMAgent.

```
aws ssm list-associations /
    --association-filter-list "key=AssociationName,value=UpdateSSMAgent"
```

Saída:

```
{
    "Associations": [
        {
            "Name": "AWS-UpdateSSMAgent",
            "InstanceId": "i-1234567890abcdef0",
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "Targets": [
                {
                    "Key": "InstanceIds",
```

```
                    "Values": [
                        "i-016648b75dd622dab"
                    ]
                }
            ],
            "Overview": {
                "Status": "Pending",
                "DetailedStatus": "Associated",
                "AssociationStatusAggregatedCount": {
                    "Pending": 1
                }
            },
            "ScheduleExpression": "cron(0 00 12 ? * SUN *)",
            "AssociationName": "UpdateSSMAgent"
        }
    ]
}
```

Para obter mais informações, consulte Trabalhar com associações no Systems Manager no Guia do usuário do Systems Manager.

Exemplo 2: listar suas associações para um documento específico

O exemplo de associações de lista a seguir lista todas as associações do documento especificado.

```
aws ssm list-associations /
    --association-filter-list "key=Name,value=AWS-UpdateSSMAgent"
```

Saída:

```
{
    "Associations": [
        {
            "Name": "AWS-UpdateSSMAgent",
            "InstanceId": "i-1234567890abcdef0",
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
```

```
                        "i-1234567890abcdef0"
                    ]
                }
            ],
            "LastExecutionDate": 1550505828.548,
            "Overview": {
                "Status": "Success",
                "DetailedStatus": "Success",
                "AssociationStatusAggregatedCount": {
                    "Success": 1
                }
            },
            "ScheduleExpression": "cron(0 00 12 ? * SUN *)",
            "AssociationName": "UpdateSSMAgent"
        },
    {
            "Name": "AWS-UpdateSSMAgent",
            "InstanceId": "i-9876543210abcdef0",
            "AssociationId": "fbc07ef7-b985-4684-b82b-0123456789ab",
            "AssociationVersion": "1",
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-9876543210abcdef0"
                    ]
                }
            ],
            "LastExecutionDate": 1550507531.0,
            "Overview": {
                "Status": "Success",
                "AssociationStatusAggregatedCount": {
                    "Success": 1
                }
            }
        }
    ]
}
```

Para obter mais informações, consulte Trabalhar com associações no Systems Manager no Guia do usuário do Systems Manager.

- Para obter detalhes da API, consulte ListAssociations na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo lista todas as associações para uma instância. A sintaxe usada nesse exemplo requer o PowerShell versão 3 ou posterior.

```
$filter1 = @{Key="InstanceId";Value=@("i-0000293ffd8c57862")}
Get-SSMAssociationList -AssociationFilterList $filter1
```

Saída:

```
AssociationId      : d8617c07-2079-4c18-9847-1655fc2698b0
DocumentVersion    :
InstanceId         : i-0000293ffd8c57862
LastExecutionDate  : 2/20/2015 8:31:11 AM
Name               : AWS-UpdateSSMAgent
Overview           : Amazon.SimpleSystemsManagement.Model.AssociationOverview
ScheduleExpression :
Targets            : {InstanceIds}
```

Exemplo 2: esse exemplo lista todas as associações para um documento de configuração. A sintaxe usada nesse exemplo requer o PowerShell versão 3 ou posterior.

```
$filter2 = @{Key="Name";Value=@("AWS-UpdateSSMAgent")}
Get-SSMAssociationList -AssociationFilterList $filter2
```

Saída:

```
AssociationId      : d8617c07-2079-4c18-9847-1655fc2698b0
DocumentVersion    :
InstanceId         : i-0000293ffd8c57862
LastExecutionDate  : 2/20/2015 8:31:11 AM
Name               : AWS-UpdateSSMAgent
Overview           : Amazon.SimpleSystemsManagement.Model.AssociationOverview
ScheduleExpression :
Targets            : {InstanceIds}
```

Exemplo 3: com o PowerShell versão 2, é necessário usar New-Object para criar cada filtro.

```
$filter1 = New-Object Amazon.SimpleSystemsManagement.Model.AssociationFilter
```

```
$filter1.Key = "InstanceId"
$filter1.Value = "i-0000293ffd8c57862"

Get-SSMAssociationList -AssociationFilterList $filter1
```

Saída:

```
AssociationId      : d8617c07-2079-4c18-9847-1655fc2698b0
DocumentVersion    :
InstanceId         : i-0000293ffd8c57862
LastExecutionDate  : 2/20/2015 8:31:11 AM
Name               : AWS-UpdateSSMAgent
Overview           : Amazon.SimpleSystemsManagement.Model.AssociationOverview
ScheduleExpression :
Targets            : {InstanceIds}
```

- Para obter detalhes da API, consulte ListAssociations na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo lista todas as associações para uma instância. A sintaxe usada nesse exemplo requer o PowerShell versão 3 ou posterior.

```
$filter1 = @{Key="InstanceId";Value=@("i-0000293ffd8c57862")}
Get-SSMAssociationList -AssociationFilterList $filter1
```

Saída:

```
AssociationId      : d8617c07-2079-4c18-9847-1655fc2698b0
DocumentVersion    :
InstanceId         : i-0000293ffd8c57862
LastExecutionDate  : 2/20/2015 8:31:11 AM
Name               : AWS-UpdateSSMAgent
Overview           : Amazon.SimpleSystemsManagement.Model.AssociationOverview
ScheduleExpression :
Targets            : {InstanceIds}
```

Exemplo 2: esse exemplo lista todas as associações para um documento de configuração. A sintaxe usada nesse exemplo requer o PowerShell versão 3 ou posterior.

```
$filter2 = @{Key="Name";Value=@("AWS-UpdateSSMAgent")}
```

```
Get-SSMAssociationList -AssociationFilterList $filter2
```

Saída:

```
AssociationId      : d8617c07-2079-4c18-9847-1655fc2698b0
DocumentVersion    :
InstanceId         : i-0000293ffd8c57862
LastExecutionDate  : 2/20/2015 8:31:11 AM
Name               : AWS-UpdateSSMAgent
Overview           : Amazon.SimpleSystemsManagement.Model.AssociationOverview
ScheduleExpression :
Targets            : {InstanceIds}
```

Exemplo 3: com o PowerShell versão 2, é necessário usar New-Object para criar cada filtro.

```
$filter1 = New-Object Amazon.SimpleSystemsManagement.Model.AssociationFilter
$filter1.Key = "InstanceId"
$filter1.Value = "i-0000293ffd8c57862"

Get-SSMAssociationList -AssociationFilterList $filter1
```

Saída:

```
AssociationId      : d8617c07-2079-4c18-9847-1655fc2698b0
DocumentVersion    :
InstanceId         : i-0000293ffd8c57862
LastExecutionDate  : 2/20/2015 8:31:11 AM
Name               : AWS-UpdateSSMAgent
Overview           : Amazon.SimpleSystemsManagement.Model.AssociationOverview
ScheduleExpression :
Targets            : {InstanceIds}
```

- Para obter detalhes da API, consulte ListAssociations na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`ListCommandInvocations`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `ListCommandInvocations`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Conheça os conceitos básicos

CLI

AWS CLI

Para listar as invocações de um comando específico

O exemplo de `list-command-invocations` a seguir lista todas as invocações de um comando.

```
aws ssm list-command-invocations \
    --command-id "ef7fdfd8-9b57-4151-a15c-db9a12345678" \
    --details
```

Saída:

```
{
    "CommandInvocations": [
        {
            "CommandId": "ef7fdfd8-9b57-4151-a15c-db9a12345678",
            "InstanceId": "i-02573cafcfEXAMPLE",
            "InstanceName": "",
            "Comment": "b48291dd-ba76-43e0-
b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",
            "DocumentName": "AWS-UpdateSSMAgent",
            "DocumentVersion": "",
            "RequestedDateTime": 1582136283.089,
            "Status": "Success",
            "StatusDetails": "Success",
            "StandardOutputUrl": "",
            "StandardErrorUrl": "",
            "CommandPlugins": [
                {
                    "Name": "aws:updateSsmAgent",
```

```
                    "Status": "Success",
                    "StatusDetails": "Success",
                    "ResponseCode": 0,
                    "ResponseStartDateTime": 1582136283.419,
                    "ResponseFinishDateTime": 1582136283.51,
                    "Output": "Updating amazon-ssm-agent from 2.3.842.0 to latest
\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-
east-2/ssm-agent-manifest.json\namazon-ssm-agent 2.3.842.0 has already been
 installed, update skipped\n",
                    "StandardOutputUrl": "",
                    "StandardErrorUrl": "",
                    "OutputS3Region": "us-east-2",
                    "OutputS3BucketName": "",
                    "OutputS3KeyPrefix": ""
                }
            ],
            "ServiceRole": "",
            "NotificationConfig": {
                "NotificationArn": "",
                "NotificationEvents": [],
                "NotificationType": ""
            },
            "CloudWatchOutputConfig": {
                "CloudWatchLogGroupName": "",
                "CloudWatchOutputEnabled": false
            }
        },
        {
            "CommandId": "ef7fdfd8-9b57-4151-a15c-db9a12345678",
            "InstanceId": "i-0471e04240EXAMPLE",
            "InstanceName": "",
            "Comment": "b48291dd-ba76-43e0-
b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",
            "DocumentName": "AWS-UpdateSSMAgent",
            "DocumentVersion": "",
            "RequestedDateTime": 1582136283.02,
            "Status": "Success",
            "StatusDetails": "Success",
            "StandardOutputUrl": "",
            "StandardErrorUrl": "",
            "CommandPlugins": [
                {
                    "Name": "aws:updateSsmAgent",
                    "Status": "Success",
```

```
                    "StatusDetails": "Success",
                    "ResponseCode": 0,
                    "ResponseStartDateTime": 1582136283.812,
                    "ResponseFinishDateTime": 1582136295.031,
                    "Output": "Updating amazon-ssm-agent from 2.3.672.0 to
 latest\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-
ssm-us-east-2/ssm-agent-manifest.json\nSuccessfully downloaded https://s3.us-
east-2.amazonaws.com/amazon-ssm-us-east-2/amazon-ssm-agent-updater/2.3.842.0/
amazon-ssm-agent-updater-snap-amd64.tar.gz\nSuccessfully downloaded https://
s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/amazon-ssm-agent/2.3.672.0/
amazon-ssm-agent-snap-amd64.tar.gz\nSuccessfully downloaded https://s3.us-
east-2.amazonaws.com/amazon-ssm-us-east-2/amazon-ssm-agent/2.3.842.0/amazon-ssm-
agent-snap-amd64.tar.gz\nInitiating amazon-ssm-agent update to 2.3.842.0\namazon-
ssm-agent updated successfully to 2.3.842.0",
                    "StandardOutputUrl": "",
                    "StandardErrorUrl": "",
                    "OutputS3Region": "us-east-2",
                    "OutputS3BucketName": "",
                    "OutputS3KeyPrefix": "8bee3135-398c-4d31-99b6-e42d2EXAMPLE/
i-0471e04240EXAMPLE/awsupdateSsmAgent"
                }
            ],
            "ServiceRole": "",
            "NotificationConfig": {
                "NotificationArn": "",
                "NotificationEvents": [],
                "NotificationType": ""
            },
            "CloudWatchOutputConfig": {
                "CloudWatchLogGroupName": "",
                "CloudWatchOutputEnabled": false
            }
        }
    ]
}
```


Para obter mais informações, consulte Entender os status dos comandos no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte ListCommandInvocations na Referência de comandos da AWS CLI.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
import { paginateListCommandInvocations, SSMClient } from "@aws-sdk/client-ssm";
import { parseArgs } from "node:util";

/**
 * List SSM command invocations on an instance.
 * @param {{ instanceId: string }}
 */
export const main = async ({ instanceId }) => {
  const client = new SSMClient({});
  try {
    const listCommandInvocationsPaginated = [];
    // The paginate function is a wrapper around the base command.
    const paginator = paginateListCommandInvocations(
      { client },
      {
        InstanceId: instanceId,
      },
    );
    for await (const page of paginator) {
      listCommandInvocationsPaginated.push(...page.CommandInvocations);
    }
    console.log("Here is the list of command invocations:");
    console.log(listCommandInvocationsPaginated);
    return { CommandInvocations: listCommandInvocationsPaginated };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ValidationError") {
      console.warn(`${caught.message}. Did you provide a valid instance ID?`);
    }
    throw caught;
  }
};
```

- Para obter detalhes da API, consulte ListCommandInvocations na Referência da API do AWS SDK para JavaScript.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo lista todas as invocações de um comando.

```
Get-SSMCommandInvocation -CommandId "b8eac879-0541-439d-94ec-47a80d554f44" -
Detail $true
```

Saída:

```
CommandId          : b8eac879-0541-439d-94ec-47a80d554f44
CommandPlugins     : {aws:runShellScript}
Comment            : IP config
DocumentName       : AWS-RunShellScript
InstanceId         : i-0cb2b964d3e14fd9f
InstanceName       :
NotificationConfig : Amazon.SimpleSystemsManagement.Model.NotificationConfig
RequestedDateTime  : 2/22/2017 8:13:16 PM
ServiceRole        :
StandardErrorUrl   :
StandardOutputUrl  :
Status             : Success
StatusDetails      : Success
TraceOutput        :
```

Exemplo 2: esse exemplo lista CommandPlugins para invocação do ID de comando e1eb2e3c-ed4c-5123-45c1-234f5612345f

```
Get-SSMCommandInvocation -CommandId e1eb2e3c-ed4c-5123-45c1-234f5612345f -Detail:
$true | Select-Object -ExpandProperty CommandPlugins
```

Saída:

```
Name                  : aws:runPowerShellScript
Output                : Completed 17.7 KiB/17.7 KiB (40.1 KiB/s) with 1 file(s)
 remainingdownload: s3://dd-aess-r-ctmer/KUMO.png to ..\..\programdata\KUMO.png
```

```
                          kumo available

OutputS3BucketName     :
OutputS3KeyPrefix      :
OutputS3Region         : eu-west-1
ResponseCode           : 0
ResponseFinishDateTime : 4/3/2019 11:53:23 AM
ResponseStartDateTime  : 4/3/2019 11:53:21 AM
StandardErrorUrl       :
StandardOutputUrl      :
Status                 : Success
StatusDetails          : Success
```

- Para obter detalhes da API, consulte ListCommandInvocations na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo lista todas as invocações de um comando.

```
Get-SSMCommandInvocation -CommandId "b8eac879-0541-439d-94ec-47a80d554f44" -
Detail $true
```

Saída:

```
CommandId          : b8eac879-0541-439d-94ec-47a80d554f44
CommandPlugins     : {aws:runShellScript}
Comment            : IP config
DocumentName       : AWS-RunShellScript
InstanceId         : i-0cb2b964d3e14fd9f
InstanceName       :
NotificationConfig : Amazon.SimpleSystemsManagement.Model.NotificationConfig
RequestedDateTime  : 2/22/2017 8:13:16 PM
ServiceRole        :
StandardErrorUrl   :
StandardOutputUrl  :
Status             : Success
StatusDetails      : Success
TraceOutput        :
```

Exemplo 2: esse exemplo lista CommandPlugins para invocação do ID de comando e1eb2e3c-ed4c-5123-45c1-234f5612345f

```
Get-SSMCommandInvocation -CommandId e1eb2e3c-ed4c-5123-45c1-234f5612345f -Detail:
$true | Select-Object -ExpandProperty CommandPlugins
```

Saída:

```
Name                   : aws:runPowerShellScript
Output                 : Completed 17.7 KiB/17.7 KiB (40.1 KiB/s) with 1 file(s)
 remainingdownload: s3://dd-aess-r-ctmer/KUMO.png to ..\..\programdata\KUMO.png
                          kumo available

OutputS3BucketName     :
OutputS3KeyPrefix      :
OutputS3Region         : eu-west-1
ResponseCode           : 0
ResponseFinishDateTime : 4/3/2019 11:53:23 AM
ResponseStartDateTime  : 4/3/2019 11:53:21 AM
StandardErrorUrl       :
StandardOutputUrl      :
Status                 : Success
StatusDetails          : Success
```

- Para obter detalhes da API, consulte ListCommandInvocations na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
class DocumentWrapper:
    """Encapsulates AWS Systems Manager Document actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
```

```
        """
        self.ssm_client = ssm_client
        self.name = None

    @classmethod
    def from_client(cls):
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def list_command_invocations(self, instance_id):
        """
        Lists the commands for an instance.

        :param instance_id: The ID of the instance.
        :return: The list of commands.
        """
        try:
            paginator = self.ssm_client.get_paginator("list_command_invocations")
            command_invocations = []
            for page in paginator.paginate(InstanceId=instance_id):
                command_invocations.extend(page["CommandInvocations"])
            num_of_commands = len(command_invocations)
            print(
                f"{num_of_commands} command invocation(s) found for instance
{instance_id}."
            )

            if num_of_commands > 10:
                print("Displaying the first 10 commands:")
                num_of_commands = 10
            date_format = "%A, %d %B %Y %I:%M%p"
            for command in command_invocations[:num_of_commands]:
                print(
                    f"   The time of command invocation is
{command['RequestedDateTime'].strftime(date_format)}"
                )
        except ClientError as err:
            logger.error(
                "Couldn't list commands for %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
```

```
            raise
```

- Para obter detalhes da API, consulte ListCommandInvocations na Referência da API AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`ListCommands`** com uma CLI

Os exemplos de código a seguir mostram como usar o `ListCommands`.

CLI

AWS CLI

Exemplo 1: obter o status de um comando específico

O exemplo de `list-commands` a seguir recupera e exibe o status do comando especificado.

```
aws ssm list-commands \
    --command-id "0831e1a8-a1ac-4257-a1fd-c831bEXAMPLE"
```

Exemplo 2: para obter o status dos comandos solicitados após uma data específica

O exemplo de `list-commands` a seguir recupera os detalhes dos comandos solicitados após a data especificada.

```
aws ssm list-commands \
    --filter "key=InvokedAfter,value=2020-02-01T00:00:00Z"
```

Exemplo 3: listar todos os comandos solicitados em uma conta da AWS

O exemplo de `list-commands` a seguir lista todos os comandos solicitados pelos usuários na conta e na região da AWS atuais.

```
aws ssm list-commands
```

Saída:


```
{
    "Commands": [
        {
            "CommandId": "8bee3135-398c-4d31-99b6-e42d2EXAMPLE",
            "DocumentName": "AWS-UpdateSSMAgent",
            "DocumentVersion": "",
            "Comment": "b48291dd-ba76-43e0-
b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",
            "ExpiresAfter": "2020-02-19T11:28:02.500000-08:00",
            "Parameters": {},
            "InstanceIds": [
                "i-028ea792daEXAMPLE",
                "i-02feef8c46EXAMPLE",
                "i-038613f3f0EXAMPLE",
                "i-03a530a2d4EXAMPLE",
                "i-083b678d37EXAMPLE",
                "i-0dee81debaEXAMPLE"
            ],
            "Targets": [],
            "RequestedDateTime": "2020-02-19T10:18:02.500000-08:00",
            "Status": "Success",
            "StatusDetails": "Success",
            "OutputS3BucketName": "",
            "OutputS3KeyPrefix": "",
            "MaxConcurrency": "50",
            "MaxErrors": "100%",
            "TargetCount": 6,
            "CompletedCount": 6,
            "ErrorCount": 0,
            "DeliveryTimedOutCount": 0,
            "ServiceRole": "",
            "NotificationConfig": {
                "NotificationArn": "",
                "NotificationEvents": [],
                "NotificationType": ""
            },
            "CloudWatchOutputConfig": {
                "CloudWatchLogGroupName": "",
                "CloudWatchOutputEnabled": false
            }
        }
        {
```

```
            "CommandId": "e9ade581-c03d-476b-9b07-26667EXAMPLE",
            "DocumentName": "AWS-FindWindowsUpdates",
            "DocumentVersion": "1",
            "Comment": "",
            "ExpiresAfter": "2020-01-24T12:37:31.874000-08:00",
            "Parameters": {
                "KbArticleIds": [
                    ""
                ],
                "UpdateLevel": [
                    "All"
                ]
            },
            "InstanceIds": [],
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-00ec29b21eEXAMPLE",
                        "i-09911ddd90EXAMPLE"
                    ]
                }
            ],
            "RequestedDateTime": "2020-01-24T11:27:31.874000-08:00",
            "Status": "Success",
            "StatusDetails": "Success",
            "OutputS3BucketName": "my-us-east-2-bucket",
            "OutputS3KeyPrefix": "my-rc-output",
            "MaxConcurrency": "50",
            "MaxErrors": "0",
            "TargetCount": 2,
            "CompletedCount": 2,
            "ErrorCount": 0,
            "DeliveryTimedOutCount": 0,
            "ServiceRole": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "NotificationConfig": {
                "NotificationArn": "arn:aws:sns:us-east-2:111222333444:my-us-
east-2-notification-arn",
                "NotificationEvents": [
                    "All"
                ],
                "NotificationType": "Invocation"
            },
```

```
            "CloudWatchOutputConfig": {
                "CloudWatchLogGroupName": "",
                "CloudWatchOutputEnabled": false
            }
        }
        {
            "CommandId": "d539b6c3-70e8-4853-80e5-0ce4fEXAMPLE",
            "DocumentName": "AWS-RunPatchBaseline",
            "DocumentVersion": "1",
            "Comment": "",
            "ExpiresAfter": "2020-01-24T12:21:04.350000-08:00",
            "Parameters": {
                "InstallOverrideList": [
                    ""
                ],
                "Operation": [
                    "Install"
                ],
                "RebootOption": [
                    "RebootIfNeeded"
                ],
                "SnapshotId": [
                    ""
                ]
            },
            "InstanceIds": [],
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-00ec29b21eEXAMPLE",
                        "i-09911ddd90EXAMPLE"
                    ]
                }
            ],
            "RequestedDateTime": "2020-01-24T11:11:04.350000-08:00",
            "Status": "Success",
            "StatusDetails": "Success",
            "OutputS3BucketName": "my-us-east-2-bucket",
            "OutputS3KeyPrefix": "my-rc-output",
            "MaxConcurrency": "50",
            "MaxErrors": "0",
            "TargetCount": 2,
            "CompletedCount": 2,
```

```
            "ErrorCount": 0,
            "DeliveryTimedOutCount": 0,
            "ServiceRole": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "NotificationConfig": {
                "NotificationArn": "arn:aws:sns:us-east-2:111222333444:my-us-
east-2-notification-arn",
                "NotificationEvents": [
                    "All"
                ],
                "NotificationType": "Invocation"
            },
            "CloudWatchOutputConfig": {
                "CloudWatchLogGroupName": "",
                "CloudWatchOutputEnabled": false
            }
        }
    ]
}
```

Para obter mais informações, consulte Executar comandos usando o Systems Manager Run Command no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte ListCommands na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo lista todos os comandos solicitados.

```
Get-SSMCommand
```

Saída:

```
CommandId          : 4b75a163-d39a-4d97-87c9-98ae52c6be35
Comment            : Apply association with id at update time: 4cc73e42-
d5ae-4879-84f8-57e09c0efcd0
CompletedCount     : 1
DocumentName       : AWS-RefreshAssociation
ErrorCount         : 0
```

```
ExpiresAfter       : 2/24/2017 3:19:08 AM
InstanceIds        : {i-0cb2b964d3e14fd9f}
MaxConcurrency     : 50
MaxErrors          : 0
NotificationConfig : Amazon.SimpleSystemsManagement.Model.NotificationConfig
OutputS3BucketName :
OutputS3KeyPrefix  :
OutputS3Region     :
Parameters         : {[associationIds,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
RequestedDateTime  : 2/24/2017 3:18:08 AM
ServiceRole        :
Status             : Success
StatusDetails      : Success
TargetCount        : 1
Targets            : {}
```

Exemplo 2: esse exemplo obtém o status de um comando específico

```
Get-SSMCommand -CommandId "4b75a163-d39a-4d97-87c9-98ae52c6be35"
```

Exemplo 3: esse exemplo recupera todos os comandos SSM invocados após 2019-04-01T00:00:00Z

```
Get-SSMCommand -Filter @{Key="InvokedAfter";Value="2019-04-01T00:00:00Z"} |
 Select-Object CommandId, DocumentName, Status, RequestedDateTime | Sort-Object -
Property RequestedDateTime -Descending
```

Saída:

```
CommandId                            DocumentName               Status
 RequestedDateTime
---------                            ------------               ------
 -----------------
edb1b23e-456a-7adb-aef8-90e-012ac34f AWS-RunPowerShellScript    Cancelled
 4/16/2019 5:45:23 AM
1a2dc3fb-4567-890d-a1ad-234b5d6bc7d9 AWS-ConfigureAWSPackage    Success
 4/6/2019 9:19:42 AM
12c3456c-7e90-4f12-1232-1234f5b67893 KT-Retrieve-Cloud-Type-Win Failed
 4/2/2019 4:13:07 AM
fe123b45-240c-4123-a2b3-234bdd567ecf AWS-RunInspecChecks        Failed
 4/1/2019 2:27:31 PM
```

```
1eb23aa4-567d-4123-12a3-4c1c2ab34561 AWS-RunPowerShellScript    Success
 4/1/2019 1:05:55 PM
1c2f3bb4-ee12-4bc1-1a23-12345eea123e AWS-RunInspecChecks        Failed
 4/1/2019 11:13:09 AM
```

- Para obter detalhes da API, consulte ListCommands na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo lista todos os comandos solicitados.

```
Get-SSMCommand
```

Saída:

```
CommandId          : 4b75a163-d39a-4d97-87c9-98ae52c6be35
Comment            : Apply association with id at update time: 4cc73e42-
d5ae-4879-84f8-57e09c0efcd0
CompletedCount     : 1
DocumentName       : AWS-RefreshAssociation
ErrorCount         : 0
ExpiresAfter       : 2/24/2017 3:19:08 AM
InstanceIds        : {i-0cb2b964d3e14fd9f}
MaxConcurrency     : 50
MaxErrors          : 0
NotificationConfig : Amazon.SimpleSystemsManagement.Model.NotificationConfig
OutputS3BucketName :
OutputS3KeyPrefix  :
OutputS3Region     :
Parameters         : {[associationIds,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
RequestedDateTime  : 2/24/2017 3:18:08 AM
ServiceRole        :
Status             : Success
StatusDetails      : Success
TargetCount        : 1
Targets            : {}
```

Exemplo 2: esse exemplo obtém o status de um comando específico

```
Get-SSMCommand -CommandId "4b75a163-d39a-4d97-87c9-98ae52c6be35"
```

Exemplo 3: esse exemplo recupera todos os comandos SSM invocados após 2019-04-01T00:00:00Z

```
Get-SSMCommand -Filter @{Key="InvokedAfter";Value="2019-04-01T00:00:00Z"} |
 Select-Object CommandId, DocumentName, Status, RequestedDateTime | Sort-Object -
Property RequestedDateTime -Descending
```

Saída:

```
CommandId                            DocumentName              Status
 RequestedDateTime
---------                            ------------              ------
 -----------------
edb1b23e-456a-7adb-aef8-90e-012ac34f AWS-RunPowerShellScript    Cancelled
 4/16/2019 5:45:23 AM
1a2dc3fb-4567-890d-a1ad-234b5d6bc7d9 AWS-ConfigureAWSPackage    Success
 4/6/2019 9:19:42 AM
12c3456c-7e90-4f12-1232-1234f5b67893 KT-Retrieve-Cloud-Type-Win Failed
 4/2/2019 4:13:07 AM
fe123b45-240c-4123-a2b3-234bdd567ecf AWS-RunInspecChecks        Failed
 4/1/2019 2:27:31 PM
1eb23aa4-567d-4123-12a3-4c1c2ab34561 AWS-RunPowerShellScript    Success
 4/1/2019 1:05:55 PM
1c2f3bb4-ee12-4bc1-1a23-12345eea123e AWS-RunInspecChecks        Failed
 4/1/2019 11:13:09 AM
```

- Para obter detalhes da API, consulte ListCommands na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`ListComplianceItems`** com uma CLI

Os exemplos de código a seguir mostram como usar o `ListComplianceItems`.

CLI

AWS CLI

Para listar itens de conformidade para uma instância específica

Este exemplo lista todos os itens de conformidade para a instância especificada.

Comando:

```
aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-types "ManagedInstance"
```

Saída:

```
{
  "ComplianceItems": [
      {
          "ComplianceType": "Association",
          "ResourceType": "ManagedInstance",
          "ResourceId": "i-1234567890abcdef0",
          "Id": "8dfe3659-4309-493a-8755-0123456789ab",
          "Title": "",
          "Status": "COMPLIANT",
          "Severity": "UNSPECIFIED",
          "ExecutionSummary": {
              "ExecutionTime": 1550408470.0
          },
          "Details": {
              "DocumentName": "AWS-GatherSoftwareInventory",
              "DocumentVersion": "1"
          }
      },
      {
          "ComplianceType": "Association",
          "ResourceType": "ManagedInstance",
          "ResourceId": "i-1234567890abcdef0",
          "Id": "e4c2ed6d-516f-41aa-aa2a-0123456789ab",
          "Title": "",
          "Status": "COMPLIANT",
          "Severity": "UNSPECIFIED",
          "ExecutionSummary": {
              "ExecutionTime": 1550508475.0
```

```
            },
            "Details": {
                "DocumentName": "AWS-UpdateSSMAgent",
                "DocumentVersion": "1"
            }
        },
                ...
    ],
    "NextToken": "--token string truncated--"
}
```

Para listar itens de conformidade para uma instância e um ID de associação específicos

Este exemplo lista todos os itens de conformidade para a instância e o ID de associação especificados.

Comando:

```
aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-
types "ManagedInstance" --
filters "Key=ComplianceType,Values=Association,Type=EQUAL" "Key=Id,Values=e4c2ed6d-516f-4
aa2a-0123456789ab,Type=EQUAL"
```

Para listar itens de conformidade para uma instância específica após uma data e uma hora específicas

Este exemplo lista todos os itens de conformidade para uma instância após a data e a hora especificadas.

Comando:

```
aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-
types "ManagedInstance" --
filters "Key=ExecutionTime,Values=2019-02-18T16:00:00Z,Type=GREATER_THAN"
```

- Para obter detalhes da API, consulte ListComplianceItems na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo mostra a lista de itens de conformidade para o ID e tipo de recurso fornecidos, filtrados pelo tipo de conformidade "Associação"

```
Get-SSMComplianceItemList -ResourceId i-1a2caf345f67d0dc2 -ResourceType
 ManagedInstance -Filter @{Key="ComplianceType";Values="Association"}
```

Saída:

```
ComplianceType   : Association
Details          : {[DocumentName, AWS-GatherSoftwareInventory],
 [DocumentVersion, 1]}
ExecutionSummary :
 Amazon.SimpleSystemsManagement.Model.ComplianceExecutionSummary
Id               : 123a45a1-c234-1234-1245-67891236db4e
ResourceId       : i-1a2caf345f67d0dc2
ResourceType     : ManagedInstance
Severity         : UNSPECIFIED
Status           : COMPLIANT
Title            :
```

- Para obter detalhes da API, consulte ListComplianceItems na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo mostra a lista de itens de conformidade para o ID e tipo de recurso fornecidos, filtrados pelo tipo de conformidade "Associação"

```
Get-SSMComplianceItemList -ResourceId i-1a2caf345f67d0dc2 -ResourceType
 ManagedInstance -Filter @{Key="ComplianceType";Values="Association"}
```

Saída:

```
ComplianceType   : Association
Details          : {[DocumentName, AWS-GatherSoftwareInventory],
 [DocumentVersion, 1]}
ExecutionSummary :
 Amazon.SimpleSystemsManagement.Model.ComplianceExecutionSummary
Id               : 123a45a1-c234-1234-1245-67891236db4e
```

```
ResourceId       : i-1a2caf345f67d0dc2
ResourceType     : ManagedInstance
Severity         : UNSPECIFIED
Status           : COMPLIANT
Title            :
```

- Para obter detalhes da API, consulte ListComplianceItems na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`ListComplianceSummaries`** com uma CLI

Os exemplos de código a seguir mostram como usar o `ListComplianceSummaries`.

CLI

AWS CLI

Para listar resumos de conformidade para todos os tipos de conformidade

Este exemplo lista resumos de conformidade para todos os tipos de conformidade em sua conta.

Comando:

```
aws ssm list-compliance-summaries
```

Saída:

```
{
  "ComplianceSummaryItems": [
      {
          "ComplianceType": "Association",
          "CompliantSummary": {
              "CompliantCount": 2,
              "SeveritySummary": {
                  "CriticalCount": 0,
                  "HighCount": 0,
                  "MediumCount": 0,
```

```
                "LowCount": 0,
                "InformationalCount": 0,
                "UnspecifiedCount": 2
            }
        },
        "NonCompliantSummary": {
            "NonCompliantCount": 0,
            "SeveritySummary": {
                "CriticalCount": 0,
                "HighCount": 0,
                "MediumCount": 0,
                "LowCount": 0,
                "InformationalCount": 0,
                "UnspecifiedCount": 0
            }
        }
    },
    {
        "ComplianceType": "Patch",
        "CompliantSummary": {
            "CompliantCount": 1,
            "SeveritySummary": {
                "CriticalCount": 0,
                "HighCount": 0,
                "MediumCount": 0,
                "LowCount": 0,
                "InformationalCount": 0,
                "UnspecifiedCount": 1
            }
        },
        "NonCompliantSummary": {
            "NonCompliantCount": 1,
            "SeveritySummary": {
                "CriticalCount": 1,
                "HighCount": 0,
                "MediumCount": 0,
                "LowCount": 0,
                "InformationalCount": 0,
                "UnspecifiedCount": 0
            }
        }
    },
            ...
],
```

```
  "NextToken": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

Para listar resumos de conformidade para um tipo de conformidade específico

Este exemplo lista o resumo de conformidade para o tipo de conformidade do patch.

Comando:

```
aws ssm list-compliance-summaries --
filters "Key=ComplianceType,Values=Patch,Type=EQUAL"
```

- Para obter detalhes da API, consulte ListComplianceSummaries na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo devolve uma contagem resumida de recursos em ou fora de conformidade para todos os tipos de conformidade.

```
Get-SSMComplianceSummaryList
```

Saída:

```
ComplianceType CompliantSummary
 NonCompliantSummary
-------------- ----------------
 ------------------
FleetTotal     Amazon.SimpleSystemsManagement.Model.CompliantSummary
 Amazon.SimpleSystemsManagement.Model.NonCompliantSummary
Association    Amazon.SimpleSystemsManagement.Model.CompliantSummary
 Amazon.SimpleSystemsManagement.Model.NonCompliantSummary
Custom:InSpec  Amazon.SimpleSystemsManagement.Model.CompliantSummary
 Amazon.SimpleSystemsManagement.Model.NonCompliantSummary
Patch          Amazon.SimpleSystemsManagement.Model.CompliantSummary
 Amazon.SimpleSystemsManagement.Model.NonCompliantSummary
```

- Para obter detalhes da API, consulte ListComplianceSummaries na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo devolve uma contagem resumida de recursos em ou fora de conformidade para todos os tipos de conformidade.

```
Get-SSMComplianceSummaryList
```

Saída:

```
ComplianceType CompliantSummary
 NonCompliantSummary
-------------- ---------------
 ------------------
FleetTotal     Amazon.SimpleSystemsManagement.Model.CompliantSummary
 Amazon.SimpleSystemsManagement.Model.NonCompliantSummary
Association    Amazon.SimpleSystemsManagement.Model.CompliantSummary
 Amazon.SimpleSystemsManagement.Model.NonCompliantSummary
Custom:InSpec  Amazon.SimpleSystemsManagement.Model.CompliantSummary
 Amazon.SimpleSystemsManagement.Model.NonCompliantSummary
Patch          Amazon.SimpleSystemsManagement.Model.CompliantSummary
 Amazon.SimpleSystemsManagement.Model.NonCompliantSummary
```

- Para obter detalhes da API, consulte ListComplianceSummaries na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`ListDocumentVersions`** com uma CLI

Os exemplos de código a seguir mostram como usar o `ListDocumentVersions`.

CLI

AWS CLI

Para listar versões de documentos

O exemplo de `list-document-versions` a seguir lista todas as versões de um documento do Systems Manager.

```
aws ssm list-document-versions \
    --name "Example"
```

Saída:

```
{
    "DocumentVersions": [
        {
            "Name": "Example",
            "DocumentVersion": "1",
            "CreatedDate": 1583257938.266,
            "IsDefaultVersion": true,
            "DocumentFormat": "YAML",
            "Status": "Active"
        }
    ]
}
```

Para obter mais informações, consulte Enviar comandos que usam o parâmetro de versão do documento no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte ListDocumentVersions na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo lista todas as versões de um documento.

```
Get-SSMDocumentVersionList -Name "AWS-UpdateSSMAgent"
```

Saída:

```
CreatedDate       : 6/1/2021 5:19:10 PM
DocumentFormat    : JSON
DocumentVersion   : 1
IsDefaultVersion  : True
Name              : AWS-UpdateSSMAgent
Status            : Active
```

- Para obter detalhes da API, consulte ListDocumentVersions na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo lista todas as versões de um documento.

```
Get-SSMDocumentVersionList -Name "AWS-UpdateSSMAgent"
```

Saída:

```
CreatedDate       : 6/1/2021 5:19:10 PM
DocumentFormat    : JSON
DocumentVersion   : 1
IsDefaultVersion  : True
Name              : AWS-UpdateSSMAgent
Status            : Active
```

- Para obter detalhes da API, consulte ListDocumentVersions na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`ListDocuments`** com uma CLI

Os exemplos de código a seguir mostram como usar o `ListDocuments`.

CLI

AWS CLI

Exemplo 1: listar documentos

O exemplo de `list-documents` a seguir lista documentos pertencentes à conta solicitante marcados com a tag personalizada.

```
aws ssm list-documents \
    --filters Key=Owner,Values=Self Key=tag:DocUse,Values=Testing
```

Saída:

```
{
    "DocumentIdentifiers": [
        {
            "Name": "Example",
            "Owner": "29884EXAMPLE",
            "PlatformTypes": [
                "Windows",
                "Linux"
            ],
            "DocumentVersion": "1",
            "DocumentType": "Automation",
            "SchemaVersion": "0.3",
            "DocumentFormat": "YAML",
            "Tags": [
                {
                    "Key": "DocUse",
                    "Value": "Testing"
                }
            ]
        }
    ]
}
```

Para obter mais informações, consulte Documentos do AWS Systems Manager no Guia do usuário do AWS Systems Manager.

Exemplo 2: listar documentos compartilhados

O exemplo de `list-documents` a seguir lista documentos compartilhados, incluindo documentos compartilhados privados que não pertencem à AWS.

```
aws ssm list-documents \
    --filters Key=Name,Values=sharedDocNamePrefix  Key=Owner,Values=Private
```

Saída:

```
{
    "DocumentIdentifiers": [
        {
            "Name": "Example",
```

```
            "Owner": "12345EXAMPLE",
            "PlatformTypes": [
                "Windows",
                "Linux"
            ],
            "DocumentVersion": "1",
            "DocumentType": "Command",
            "SchemaVersion": "0.3",
            "DocumentFormat": "YAML",
            "Tags": []
        }
    ]
}
```

Para obter mais informações, consulte Documentos do AWS Systems Manager no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte ListDocuments na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: lista todos os documentos de configuração em sua conta.

```
Get-SSMDocumentList
```

Saída:

```
DocumentType    : Command
DocumentVersion : 1
Name            : AWS-ApplyPatchBaseline
Owner           : Amazon
PlatformTypes   : {Windows}
SchemaVersion   : 1.2

DocumentType    : Command
DocumentVersion : 1
Name            : AWS-ConfigureAWSPackage
Owner           : Amazon
PlatformTypes   : {Windows, Linux}
```

```
SchemaVersion   : 2.0

DocumentType    : Command
DocumentVersion : 1
Name            : AWS-ConfigureCloudWatch
Owner           : Amazon
PlatformTypes   : {Windows}
SchemaVersion   : 1.2
...
```

Exemplo 2: esse exemplo recupera todos os documentos de automação com o nome correspondente a "Plataform"

```
Get-SSMDocumentList -DocumentFilterList @{Key="DocumentType";Value="Automation"}
 | Where-Object Name -Match "Platform"
```

Saída:

```
DocumentFormat  : JSON
DocumentType    : Automation
DocumentVersion : 7
Name            : KT-Get-Platform
Owner           : 987654123456
PlatformTypes   : {Windows, Linux}
SchemaVersion   : 0.3
Tags            : {}
TargetType      :
VersionName     :
```

- Para obter detalhes da API, consulte ListDocuments na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: lista todos os documentos de configuração em sua conta.

```
Get-SSMDocumentList
```

Saída:

```
DocumentType    : Command
```

```
DocumentVersion : 1
Name            : AWS-ApplyPatchBaseline
Owner           : Amazon
PlatformTypes   : {Windows}
SchemaVersion   : 1.2

DocumentType    : Command
DocumentVersion : 1
Name            : AWS-ConfigureAWSPackage
Owner           : Amazon
PlatformTypes   : {Windows, Linux}
SchemaVersion   : 2.0

DocumentType    : Command
DocumentVersion : 1
Name            : AWS-ConfigureCloudWatch
Owner           : Amazon
PlatformTypes   : {Windows}
SchemaVersion   : 1.2
...
```

Exemplo 2: esse exemplo recupera todos os documentos de automação com o nome correspondente a "Plataform"

```
Get-SSMDocumentList -DocumentFilterList @{Key="DocumentType";Value="Automation"}
 | Where-Object Name -Match "Platform"
```

Saída:

```
DocumentFormat  : JSON
DocumentType    : Automation
DocumentVersion : 7
Name            : KT-Get-Platform
Owner           : 987654123456
PlatformTypes   : {Windows, Linux}
SchemaVersion   : 0.3
Tags            : {}
TargetType      :
VersionName     :
```

- Para obter detalhes da API, consulte ListDocuments na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`ListInventoryEntries`** com uma CLI

Os exemplos de código a seguir mostram como usar o `ListInventoryEntries`.

CLI

AWS CLI

Exemplo 1: visualizar entradas específicas do tipo de inventário para uma instância

O exemplo de `list-inventory-entries` a seguir lista as entradas de inventário do tipo de inventário AWS:Application em uma instância específica.

```
aws ssm list-inventory-entries \
    --instance-id "i-1234567890abcdef0" \
    --type-name "AWS:Application"
```

Saída:

```
{
  "TypeName": "AWS:Application",
  "InstanceId": "i-1234567890abcdef0",
  "SchemaVersion": "1.1",
  "CaptureTime": "2019-02-15T12:17:55Z",
  "Entries": [
    {
      "Architecture": "i386",
      "Name": "Amazon SSM Agent",
      "PackageId": "{88a60be2-89a1-4df8-812a-80863c2a2b68}",
      "Publisher": "Amazon Web Services",
      "Version": "2.3.274.0"
    },
    {
      "Architecture": "x86_64",
      "InstalledTime": "2018-05-03T13:42:34Z",
      "Name": "AmazonCloudWatchAgent",
      "Publisher": "",
      "Version": "1.200442.0"
```

```
    }
  ]
}
```

Exemplo 2: visualizar entradas de inventário personalizadas atribuídas a uma instância

O exemplo de `list-inventory-entries` a seguir lista uma entrada de inventário personalizada atribuída a uma instância.

```
aws ssm list-inventory-entries \
    --instance-id "i-1234567890abcdef0" \
    --type-name "Custom:RackInfo"
```

Saída:

```
{
  "TypeName": "Custom:RackInfo",
  "InstanceId": "i-1234567890abcdef0",
  "SchemaVersion": "1.0",
  "CaptureTime": "2021-05-22T10:01:01Z",
  "Entries": [
    {
      "RackLocation": "Bay B/Row C/Rack D/Shelf E"
    }
  ]
}
```

- Para obter detalhes da API, consulte ListInventoryEntries na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo lista todas as entradas de inventário personalizadas para uma instância.

```
Get-SSMInventoryEntriesList -InstanceId "i-0cb2b964d3e14fd9f" -TypeName
 "Custom:RackInfo"
```

Saída:

```
CaptureTime   : 2016-08-22T10:01:01Z
Entries       :
 {Amazon.Runtime.Internal.Util.AlwaysSendDictionary`2[System.String,System.String]}
InstanceId    : i-0cb2b964d3e14fd9f
NextToken     :
SchemaVersion : 1.0
TypeName      : Custom:RackInfo
```

Exemplo 2: esse exemplo lista os detalhes.

```
(Get-SSMInventoryEntriesList -InstanceId "i-0cb2b964d3e14fd9f" -TypeName
 "Custom:RackInfo").Entries
```

Saída:

```
Key          Value
---          -----
RackLocation Bay B/Row C/Rack D/Shelf E
```

- Para obter detalhes da API, consulte ListInventoryEntries na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo lista todas as entradas de inventário personalizadas para uma instância.

```
Get-SSMInventoryEntriesList -InstanceId "i-0cb2b964d3e14fd9f" -TypeName
 "Custom:RackInfo"
```

Saída:

```
CaptureTime   : 2016-08-22T10:01:01Z
Entries       :
 {Amazon.Runtime.Internal.Util.AlwaysSendDictionary`2[System.String,System.String]}
InstanceId    : i-0cb2b964d3e14fd9f
NextToken     :
SchemaVersion : 1.0
TypeName      : Custom:RackInfo
```

Exemplo 2: esse exemplo lista os detalhes.

```
(Get-SSMInventoryEntriesList -InstanceId "i-0cb2b964d3e14fd9f" -TypeName
 "Custom:RackInfo").Entries
```

Saída:

```
Key          Value
---          -----
RackLocation Bay B/Row C/Rack D/Shelf E
```

- Para obter detalhes da API, consulte ListInventoryEntries na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`ListResourceComplianceSummaries`** com uma CLI

Os exemplos de código a seguir mostram como usar o `ListResourceComplianceSummaries`.

CLI

AWS CLI

Para listar as contagens resumidas de conformidade em nível de recurso

Este exemplo lista as contagens resumidas de conformidade em nível de recurso.

Comando:

```
aws ssm list-resource-compliance-summaries
```

Saída:

```
{
  "ResourceComplianceSummaryItems": [
      {
          "ComplianceType": "Association",
          "ResourceType": "ManagedInstance",
          "ResourceId": "i-1234567890abcdef0",
          "Status": "COMPLIANT",
```

```
            "OverallSeverity": "UNSPECIFIED",
            "ExecutionSummary": {
                "ExecutionTime": 1550509273.0
            },
            "CompliantSummary": {
                "CompliantCount": 2,
                "SeveritySummary": {
                    "CriticalCount": 0,
                    "HighCount": 0,
                    "MediumCount": 0,
                    "LowCount": 0,
                    "InformationalCount": 0,
                    "UnspecifiedCount": 2
                }
            },
            "NonCompliantSummary": {
                "NonCompliantCount": 0,
                "SeveritySummary": {
                    "CriticalCount": 0,
                    "HighCount": 0,
                    "MediumCount": 0,
                    "LowCount": 0,
                    "InformationalCount": 0,
                    "UnspecifiedCount": 0
                }
            }
        },
        {
            "ComplianceType": "Patch",
            "ResourceType": "ManagedInstance",
            "ResourceId": "i-9876543210abcdef0",
            "Status": "COMPLIANT",
            "OverallSeverity": "UNSPECIFIED",
            "ExecutionSummary": {
                "ExecutionTime": 1550248550.0,
                "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",
                "ExecutionType": "Command"
            },
            "CompliantSummary": {
                "CompliantCount": 397,
                "SeveritySummary": {
                    "CriticalCount": 0,
                    "HighCount": 0,
                    "MediumCount": 0,
```

```
                "LowCount": 0,
                "InformationalCount": 0,
                "UnspecifiedCount": 397
            }
        },
        "NonCompliantSummary": {
            "NonCompliantCount": 0,
            "SeveritySummary": {
                "CriticalCount": 0,
                "HighCount": 0,
                "MediumCount": 0,
                "LowCount": 0,
                "InformationalCount": 0,
                "UnspecifiedCount": 0
            }
        }
    }
  ],
  "NextToken": "--token string truncated--"
}
```

Para listar resumos de conformidade em nível de recurso para um tipo de conformidade específico

Este exemplo lista resumos de conformidade em nível de recurso para o tipo de conformidade do patch.

Comando:

```
aws ssm list-resource-compliance-summaries --
filters "Key=ComplianceType,Values=Patch,Type=EQUAL"
```

- Para obter detalhes da API, consulte ListResourceComplianceSummaries na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo obtém uma contagem resumida em nível de recurso. O resumo inclui informações sobre status de conformidade e não conformidade e contagens detalhadas

de gravidade de itens de conformidade para produtos que correspondem a "Windows10". Como o padrão de MaxResult é 100 quando o parâmetro não é especificado, e esse valor não é válido, o parâmetro MaxResult é adicionado e o valor é definido como 50.

```
$FilterValues = @{
  "Key"="Product"
        "Type"="EQUAL"
        "Values"="Windows10"
}
        Get-SSMResourceComplianceSummaryList -Filter $FilterValues -MaxResult 50
```

- Para obter detalhes da API, consulte ListResourceComplianceSummaries na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo obtém uma contagem resumida em nível de recurso. O resumo inclui informações sobre status de conformidade e não conformidade e contagens detalhadas de gravidade de itens de conformidade para produtos que correspondem a "Windows10". Como o padrão de MaxResult é 100 quando o parâmetro não é especificado, e esse valor não é válido, o parâmetro MaxResult é adicionado e o valor é definido como 50.

```
$FilterValues = @{
  "Key"="Product"
        "Type"="EQUAL"
        "Values"="Windows10"
}
        Get-SSMResourceComplianceSummaryList -Filter $FilterValues -MaxResult 50
```

- Para obter detalhes da API, consulte ListResourceComplianceSummaries na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`ListTagsForResource`** com uma CLI

Os exemplos de código a seguir mostram como usar o `ListTagsForResource`.

CLI

AWS CLI

Para listar as tags aplicadas a uma lista de referência de patches

O exemplo de `list-tags-for-resource` a seguir lista as tags para uma lista de referência de patches.

```
aws ssm list-tags-for-resource \
    --resource-type "PatchBaseline" \
    --resource-id "pb-0123456789abcdef0"
```

Saída:

```
{
    "TagList": [
        {
            "Key": "Environment",
            "Value": "Production"
        },
        {
            "Key": "Region",
            "Value": "EMEA"
        }
    ]
}
```

Para obter mais informações, consulte Marcar recursos da AWS na Referência geral da AWS.

- Para obter detalhes sobre a API, consulte ListTagsForResource na AWS CLI Command Reference.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo lista as tags para uma janela de manutenção.

```
Get-SSMResourceTag -ResourceId "mw-03eb9db42890fb82d" -ResourceType
 "MaintenanceWindow"
```

Saída:

```
Key   Value
---   -----
Stack Production
```

- Para obter detalhes da API, consulte ListTagsForResource na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo lista as tags para uma janela de manutenção.

```
Get-SSMResourceTag -ResourceId "mw-03eb9db42890fb82d" -ResourceType
 "MaintenanceWindow"
```

Saída:

```
Key   Value
---   -----
Stack Production
```

- Para obter detalhes da API, consulte ListTagsForResource na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`ModifyDocumentPermission`** com uma CLI

Os exemplos de código a seguir mostram como usar o `ModifyDocumentPermission`.

CLI

AWS CLI

Para modificar as permissões do documento

O exemplo de `modify-document-permission` a seguir compartilha publicamente um documento do Systems Manager.

```
aws ssm modify-document-permission \
    --name "Example" \
    --permission-type "Share" \
    --account-ids-to-add "All"
```

Este comando não produz saída.

Para obter mais informações, consulte Compartilhar um documento do Systems Manager no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte ModifyDocumentPermission na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo adiciona permissões de "compartilhamento" a todas as contas para um documento. Não haverá saída se o comando for bem-sucedido.

```
Edit-SSMDocumentPermission -Name "RunShellScript" -PermissionType "Share" -
AccountIdsToAdd all
```

Exemplo 2: esse exemplo adiciona permissões de "compartilhamento" a uma conta específica para um documento. Não haverá saída se o comando for bem-sucedido.

```
Edit-SSMDocumentPermission -Name "RunShellScriptNew" -PermissionType "Share" -
AccountIdsToAdd "123456789012"
```

- Para obter detalhes da API, consulte ModifyDocumentPermission na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo adiciona permissões de "compartilhamento" a todas as contas para um documento. Não haverá saída se o comando for bem-sucedido.

```
Edit-SSMDocumentPermission -Name "RunShellScript" -PermissionType "Share" -
AccountIdsToAdd all
```

Exemplo 2: esse exemplo adiciona permissões de "compartilhamento" a uma conta específica para um documento. Não haverá saída se o comando for bem-sucedido.

```
Edit-SSMDocumentPermission -Name "RunShellScriptNew" -PermissionType "Share" -
AccountIdsToAdd "123456789012"
```

- Para obter detalhes da API, consulte ModifyDocumentPermission na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`PutComplianceItems`** com uma CLI

Os exemplos de código a seguir mostram como usar o `PutComplianceItems`.

CLI

AWS CLI

Para registrar um tipo de conformidade e detalhes de conformidade em uma instância designada

Este exemplo registra o tipo de conformidade `Custom:AVCheck` na instância gerenciada especificada. Não haverá saída se o comando for bem-sucedido.

Comando:

```
aws ssm put-compliance-items --resource-id "i-1234567890abcdef0" --
resource-type "ManagedInstance" --compliance-type "Custom:AVCheck"
 --execution-summary "ExecutionTime=2019-02-18T16:00:00Z" --
items "Id=Version2.0,Title=ScanHost,Severity=CRITICAL,Status=COMPLIANT"
```

- Para obter detalhes da API, consulte PutComplianceItems na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo grava um item de conformidade personalizado para a instância gerenciada especificada

```
$item = [Amazon.SimpleSystemsManagement.Model.ComplianceItemEntry]::new()
$item.Id = "07Jun2019-3"
$item.Severity="LOW"
$item.Status="COMPLIANT"
$item.Title="Fin-test-1 - custom"
Write-SSMComplianceItem -ResourceId mi-012dcb3ecea45b678 -ComplianceType
 Custom:VSSCompliant2 -ResourceType ManagedInstance -Item $item -
ExecutionSummary_ExecutionTime "07-Jun-2019"
```

- Para obter detalhes da API, consulte PutComplianceItems na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo grava um item de conformidade personalizado para a instância gerenciada especificada

```
$item = [Amazon.SimpleSystemsManagement.Model.ComplianceItemEntry]::new()
$item.Id = "07Jun2019-3"
$item.Severity="LOW"
$item.Status="COMPLIANT"
$item.Title="Fin-test-1 - custom"
Write-SSMComplianceItem -ResourceId mi-012dcb3ecea45b678 -ComplianceType
 Custom:VSSCompliant2 -ResourceType ManagedInstance -Item $item -
ExecutionSummary_ExecutionTime "07-Jun-2019"
```

- Para obter detalhes da API, consulte PutComplianceItems na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`PutInventory`** com uma CLI

Os exemplos de código a seguir mostram como usar o `PutInventory`.

CLI

AWS CLI

Para atribuir metadados de inventário personalizados a uma instância

Este exemplo atribui informações de localização de rack a uma instância. Não haverá saída se o comando for bem-sucedido.

Comando (Linux):

```
aws ssm put-inventory --instance-id "i-016648b75dd622dab" --items
 '[{"TypeName": "Custom:RackInfo","SchemaVersion": "1.0","CaptureTime":
 "2019-01-22T10:01:01Z","Content":[{"RackLocation": "Bay B/Row C/Rack D/Shelf
 E"}]}]'
```

Comando (Windows):

```
aws ssm put-inventory --instance-id "i-016648b75dd622dab" --
items "TypeName=Custom:RackInfo,SchemaVersion=1.0,CaptureTime=2019-01-22T10:01:01Z,Conten
 B/Row C/Rack D/Shelf F'}]"
```

- Para obter detalhes da API, consulte PutInventory na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo atribui informações de localização de rack a uma instância. Não haverá saída se o comando for bem-sucedido.

```
$data = New-Object
 "System.Collections.Generic.Dictionary[System.String,System.String]"
$data.Add("RackLocation", "Bay B/Row C/Rack D/Shelf F")

$items = New-Object
 "System.Collections.Generic.List[System.Collections.Generic.Dictionary[System.String,
 System.String]]"
```

```
$items.Add($data)

$customInventoryItem = New-Object
 Amazon.SimpleSystemsManagement.Model.InventoryItem
$customInventoryItem.CaptureTime = "2016-08-22T10:01:01Z"
$customInventoryItem.Content = $items
$customInventoryItem.TypeName = "Custom:TestRackInfo2"
$customInventoryItem.SchemaVersion = "1.0"

$inventoryItems = @($customInventoryItem)

Write-SSMInventory -InstanceId "i-0cb2b964d3e14fd9f" -Item $inventoryItems
```

- Para obter detalhes da API, consulte PutInventory na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo atribui informações de localização de rack a uma instância. Não haverá saída se o comando for bem-sucedido.

```
$data = New-Object
 "System.Collections.Generic.Dictionary[System.String,System.String]"
$data.Add("RackLocation", "Bay B/Row C/Rack D/Shelf F")

$items = New-Object
 "System.Collections.Generic.List[System.Collections.Generic.Dictionary[System.String,
 System.String]]"
$items.Add($data)

$customInventoryItem = New-Object
 Amazon.SimpleSystemsManagement.Model.InventoryItem
$customInventoryItem.CaptureTime = "2016-08-22T10:01:01Z"
$customInventoryItem.Content = $items
$customInventoryItem.TypeName = "Custom:TestRackInfo2"
$customInventoryItem.SchemaVersion = "1.0"

$inventoryItems = @($customInventoryItem)

Write-SSMInventory -InstanceId "i-0cb2b964d3e14fd9f" -Item $inventoryItems
```

- Para obter detalhes da API, consulte PutInventory na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`PutParameter`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `PutParameter`.

CLI

AWS CLI

Exemplo 1: como alterar o valor de um parâmetro

O exemplo de `put-parameter` a seguir altera o valor do parâmetro especificado.

```
aws ssm put-parameter \
    --name "MyStringParameter" \
    --type "String" \
    --value "Vici" \
    --overwrite
```

Saída:

```
{
    "Version": 2,
    "Tier": "Standard"
}
```

Para ter mais informações, consulte Criar um parâmetro do Systems Manager (AWS CLI), Gerenciar camadas de parâmetros e Working with parameter policies no Guia do usuário do AWS Systems Manager.

Exemplo 2: como criar um parâmetro avançado

O exemplo de `put-parameter` a seguir cria um parâmetro avançado.

```
aws ssm put-parameter \
    --name "MyAdvancedParameter" \
    --description "This is an advanced parameter" \
    --value "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
 eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim
```

```
veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
consequat [truncated]" \
    --type "String" \
    --tier Advanced
```

Saída:

```
{
    "Version": 1,
    "Tier": "Advanced"
}
```

Para ter mais informações, consulte Criar um parâmetro do Systems Manager (AWS CLI), Gerenciar camadas de parâmetros e Working with parameter policies no Guia do usuário do AWS Systems Manager.

Exemplo 3: como converter um parâmetro padrão em um parâmetro avançado

O exemplo `put-parameter` a seguir converte um parâmetro padrão existente em um parâmetro avançado.

```
aws ssm put-parameter \
    --name "MyConvertedParameter" \
    --value "abc123" \
    --type "String" \
    --tier Advanced \
    --overwrite
```

Saída:

```
{
    "Version": 2,
    "Tier": "Advanced"
}
```

Para ter mais informações, consulte Criar um parâmetro do Systems Manager (AWS CLI), Gerenciar camadas de parâmetros e Working with parameter policies no Guia do usuário do AWS Systems Manager.

Exemplo 4: como criar um parâmetro com uma política anexada

O exemplo de `put-parameter` a seguir cria um parâmetro avançado com uma política de parâmetros anexada.

```
aws ssm put-parameter \
    --name "/Finance/Payroll/q2accesskey" \
    --value "P@sSwW)rd" \
    --type "SecureString" \
    --tier Advanced \
    --policies "[{\"Type\":\"Expiration\",\"Version\":\"1.0\",\"Attributes\":
{\"Timestamp\":\"2020-06-30T00:00:00.000Z\"}},{\"Type\":\"ExpirationNotification
\",\"Version\":\"1.0\",\"Attributes\":{\"Before\":\"5\",\"Unit\":\"Days\"}},
{\"Type\":\"NoChangeNotification\",\"Version\":\"1.0\",\"Attributes\":{\"After\":
\"60\",\"Unit\":\"Days\"}}]"
```

Saída:

```
{
    "Version": 1,
    "Tier": "Advanced"
}
```

Para ter mais informações, consulte Criar um parâmetro do Systems Manager (AWS CLI), Gerenciar camadas de parâmetros e Working with parameter policies no Guia do usuário do AWS Systems Manager.

Exemplo 5: como adicionar uma política a um parâmetro existente

O exemplo de `put-parameter` a seguir anexa uma política a um parâmetro avançado existente.

```
aws ssm put-parameter \
    --name "/Finance/Payroll/q2accesskey" \
    --value "N3wP@sSwW)rd" \
    --type "SecureString" \
    --tier Advanced \
    --policies "[{\"Type\":\"Expiration\",\"Version\":\"1.0\",\"Attributes\":
{\"Timestamp\":\"2020-06-30T00:00:00.000Z\"}},{\"Type\":\"ExpirationNotification
\",\"Version\":\"1.0\",\"Attributes\":{\"Before\":\"5\",\"Unit\":\"Days\"}},
{\"Type\":\"NoChangeNotification\",\"Version\":\"1.0\",\"Attributes\":{\"After\":
\"60\",\"Unit\":\"Days\"}}]"
    --overwrite
```

Saída:

```
{
    "Version": 2,
    "Tier": "Advanced"
}
```

Para ter mais informações, consulte Criar um parâmetro do Systems Manager (AWS CLI), Gerenciar camadas de parâmetros e Working with parameter policies no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte PutParameter na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.ssm.SsmClient;
import software.amazon.awssdk.services.ssm.model.ParameterType;
import software.amazon.awssdk.services.ssm.model.PutParameterRequest;
import software.amazon.awssdk.services.ssm.model.SsmException;

public class PutParameter {

    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <paraName>

                Where:
                    paraName - The name of the parameter.
                    paraValue - The value of the parameter.
```

```
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String paraName = args[0];
        String paraValue = args[1];
        Region region = Region.US_EAST_1;
        SsmClient ssmClient = SsmClient.builder()
                .region(region)
                .build();

        putParaValue(ssmClient, paraName, paraValue);
        ssmClient.close();
    }

    public static void putParaValue(SsmClient ssmClient, String paraName, String
 value) {
        try {
            PutParameterRequest parameterRequest = PutParameterRequest.builder()
                    .name(paraName)
                    .type(ParameterType.STRING)
                    .value(value)
                    .build();

            ssmClient.putParameter(parameterRequest);
            System.out.println("The parameter was successfully added.");

        } catch (SsmException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte PutParameter na Referência da API AWS SDK for Java 2.x.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo cria um parâmetro. Não haverá saída se o comando for bem-sucedido.

```
Write-SSMParameter -Name "Welcome" -Type "String" -Value "helloWorld"
```

Exemplo 2: esse exemplo altera um parâmetro. Não haverá saída se o comando for bem-sucedido.

```
Write-SSMParameter -Name "Welcome" -Type "String" -Value "Good day, Sunshine!" -
Overwrite $true
```

- Para obter detalhes da API, consulte PutParameter na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo cria um parâmetro. Não haverá saída se o comando for bem-sucedido.

```
Write-SSMParameter -Name "Welcome" -Type "String" -Value "helloWorld"
```

Exemplo 2: esse exemplo altera um parâmetro. Não haverá saída se o comando for bem-sucedido.

```
Write-SSMParameter -Name "Welcome" -Type "String" -Value "Good day, Sunshine!" -
Overwrite $true
```

- Para obter detalhes da API, consulte PutParameter na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Rust

SDK para Rust

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
async fn make_parameter(
    client: &Client,
    name: &str,
    value: &str,
    description: &str,
) -> Result<(), Error> {
    let resp = client
        .put_parameter()
        .overwrite(true)
        .r#type(ParameterType::String)
        .name(name)
        .value(value)
        .description(description)
        .send()
        .await?;

    println!("Success! Parameter now has version: {}", resp.version());

    Ok(())
}
```

- Para obter detalhes da API, consulte PutParameter na Referência da API AWS SDK para Rust.

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`RegisterDefaultPatchBaseline`** com uma CLI

Os exemplos de código a seguir mostram como usar o `RegisterDefaultPatchBaseline`.

CLI

AWS CLI

Para definir a lista de referência de patches padrão

O exemplo de `register-default-patch-baseline` a seguir registra a lista de referência de patches personalizada especificada como a lista de referência de patches padrão para o tipo de sistema operacional ao qual ela oferece suporte.

```
aws ssm register-default-patch-baseline \
    --baseline-id "pb-abc123cf9bEXAMPLE"
```

Saída:

```
{
    "BaselineId":"pb-abc123cf9bEXAMPLE"
}
```

O exemplo de `register-default-patch-baseline` a seguir registra a lista de referência de patches padrão fornecida pela AWS para o CentOS como a lista de referência de patches padrão.

```
aws ssm register-default-patch-baseline \
    --baseline-id "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-0574b43a65ea646ed"
```

Saída:

```
{
    "BaselineId":"pb-abc123cf9bEXAMPLE"
}
```

Para obter mais informações, consulte Sobre listas de referência de patches predefinidas e personalizadas no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte RegisterDefaultPatchBaseline na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: este exemplo registra uma lista de referência de patches como a lista de referência de patches padrão.

```
Register-SSMDefaultPatchBaseline -BaselineId "pb-03da896ca3b68b639"
```

Saída:

```
pb-03da896ca3b68b639
```

- Para obter detalhes da API, consulte RegisterDefaultPatchBaseline na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: este exemplo registra uma lista de referência de patches como a lista de referência de patches padrão.

```
Register-SSMDefaultPatchBaseline -BaselineId "pb-03da896ca3b68b639"
```

Saída:

```
pb-03da896ca3b68b639
```

- Para obter detalhes da API, consulte RegisterDefaultPatchBaseline na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **RegisterPatchBaselineForPatchGroup** com uma CLI

Os exemplos de código a seguir mostram como usar o `RegisterPatchBaselineForPatchGroup`.

CLI

AWS CLI

Para registrar uma lista de referência de patches de um grupo de patches

O exemplo de `register-patch-baseline-for-patch-group` a seguir registra uma lista de referência de patches para um grupo de patches.

```
aws ssm register-patch-baseline-for-patch-group \
    --baseline-id "pb-045f10b4f382baeda" \
    --patch-group "Production"
```

Saída:

```
{
    "BaselineId": "pb-045f10b4f382baeda",
    "PatchGroup": "Production"
}
```

Para obter mais informações, consulte Criar um grupo de patches <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html>__ e Adicionar um grupo de patches a uma lista de referência de patches no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte RegisterPatchBaselineForPatchGroup na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo registra uma lista de referência de patches para um grupo de patches.

```
Register-SSMPatchBaselineForPatchGroup -BaselineId "pb-03da896ca3b68b639" -
PatchGroup "Production"
```

Saída:

```
BaselineId           PatchGroup
----------           ----------
pb-03da896ca3b68b639 Production
```

- Para obter detalhes da API, consulte RegisterPatchBaselineForPatchGroup na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo registra uma lista de referência de patches para um grupo de patches.

```
Register-SSMPatchBaselineForPatchGroup -BaselineId "pb-03da896ca3b68b639" -
PatchGroup "Production"
```

Saída:

```
BaselineId           PatchGroup
----------           ----------
pb-03da896ca3b68b639 Production
```

- Para obter detalhes da API, consulte RegisterPatchBaselineForPatchGroup na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`RegisterTargetWithMaintenanceWindow`** com uma CLI

Os exemplos de código a seguir mostram como usar o `RegisterTargetWithMaintenanceWindow`.

CLI

AWS CLI

Exemplo 1: registrar um único destino com uma janela de manutenção

O exemplo de `register-target-with-maintenance-window` a seguir registra uma instância com uma janela de manutenção.

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-ab12cd34ef56gh78" \
    --target "Key=InstanceIds,Values=i-0000293ffd8c57862" \
    --owner-information "Single instance" \
    --resource-type "INSTANCE"
```

Saída:

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Exemplo 2: registrar vários destinos em uma janela de manutenção usando IDs de instância

O exemplo de `register-target-with-maintenance-window` a seguir registra duas instâncias com uma janela de manutenção especificando seus IDs de instância.

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-ab12cd34ef56gh78" \
    --target "Key=InstanceIds,Values=i-0000293ffd8c57862,i-0cb2b964d3e14fd9f" \
    --owner-information "Two instances in a list" \
    --resource-type "INSTANCE"
```

Saída:

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Exemplo 3: registrar destinos com uma janela de manutenção usando tags de recursos

O exemplo de `register-target-with-maintenance-window` a seguir registra instâncias com uma janela de manutenção especificando tags de recursos que foram aplicadas às instâncias.

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-06cf17cbefcb4bf4f" \
    --targets "Key=tag:Environment,Values=Prod" "Key=Role,Values=Web" \
    --owner-information "Production Web Servers" \
    --resource-type "INSTANCE"
```

Saída:

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Exemplo 4: registrar destinos usando um grupo de chaves de tag

O exemplo de `register-target-with-maintenance-window` a seguir registra instâncias que têm uma ou mais chaves de tags atribuídas a elas, independentemente de seus valores de chave.

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --resource-type "INSTANCE" \
    --target "Key=tag-key,Values=Name,Instance-Type,CostCenter"
```

Saída:

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Exemplo 5: registrar destinos usando um nome de grupo de recursos

O exemplo de `register-target-with-maintenance-window` a seguir registra um grupo de recursos especificado, independentemente do tipo de recurso que ele contém.

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
```

```
    --resource-type "RESOURCE_GROUP" \
    --target "Key=resource-groups:Name,Values=MyResourceGroup"
```

Saída:

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Para obter mais informações, consulte Registrar uma instância de destino com a janela de manutenção (AWS CLI) no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte RegisterTargetWithMaintenanceWindow na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo registra uma instância com uma janela de manutenção.

```
$option1 = @{Key="InstanceIds";Values=@("i-0000293ffd8c57862")}
Register-SSMTargetWithMaintenanceWindow -WindowId "mw-06cf17cbefcb4bf4f" -Target
 $option1 -OwnerInformation "Single instance" -ResourceType "INSTANCE"
```

Saída:

```
d8e47760-23ed-46a5-9f28-927337725398
```

Exemplo 2: esse exemplo registra várias instâncias com uma janela de manutenção.

```
$option1 =
 @{Key="InstanceIds";Values=@("i-0000293ffd8c57862","i-0cb2b964d3e14fd9f")}
Register-SSMTargetWithMaintenanceWindow -WindowId "mw-06cf17cbefcb4bf4f" -Target
 $option1 -OwnerInformation "Single instance" -ResourceType "INSTANCE"
```

Saída:

```
6ab5c208-9fc4-4697-84b7-b02a6cc25f7d
```

Exemplo 3: esse exemplo registra uma instância com uma janela de manutenção usando tags do EC2.

```
$option1 = @{Key="tag:Environment";Values=@("Production")}
Register-SSMTargetWithMaintenanceWindow -WindowId "mw-06cf17cbefcb4bf4f" -Target
 $option1 -OwnerInformation "Production Web Servers" -ResourceType "INSTANCE"
```

Saída:

```
2994977e-aefb-4a71-beac-df620352f184
```

- Para obter detalhes da API, consulte RegisterTargetWithMaintenanceWindow na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo registra uma instância com uma janela de manutenção.

```
$option1 = @{Key="InstanceIds";Values=@("i-0000293ffd8c57862")}
Register-SSMTargetWithMaintenanceWindow -WindowId "mw-06cf17cbefcb4bf4f" -Target
 $option1 -OwnerInformation "Single instance" -ResourceType "INSTANCE"
```

Saída:

```
d8e47760-23ed-46a5-9f28-927337725398
```

Exemplo 2: esse exemplo registra várias instâncias com uma janela de manutenção.

```
$option1 =
 @{Key="InstanceIds";Values=@("i-0000293ffd8c57862","i-0cb2b964d3e14fd9f")}
Register-SSMTargetWithMaintenanceWindow -WindowId "mw-06cf17cbefcb4bf4f" -Target
 $option1 -OwnerInformation "Single instance" -ResourceType "INSTANCE"
```

Saída:

```
6ab5c208-9fc4-4697-84b7-b02a6cc25f7d
```

Exemplo 3: esse exemplo registra uma instância com uma janela de manutenção usando tags do EC2.

```
$option1 = @{Key="tag:Environment";Values=@("Production")}
```

```
Register-SSMTargetWithMaintenanceWindow -WindowId "mw-06cf17cbefcb4bf4f" -Target
 $option1 -OwnerInformation "Production Web Servers" -ResourceType "INSTANCE"
```

Saída:

```
2994977e-aefb-4a71-beac-df620352f184
```

- Para obter detalhes da API, consulte RegisterTargetWithMaintenanceWindow na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`RegisterTaskWithMaintenanceWindow`** com uma CLI

Os exemplos de código a seguir mostram como usar o `RegisterTaskWithMaintenanceWindow`.

CLI

AWS CLI

Exemplo 1: registrar uma tarefa do Automation com uma janela de manutenção

O exemplo de `register-task-with-maintenance-window` a seguir registra uma tarefa do Automation com uma janela de manutenção voltada para uma instância.

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-082dcd7649EXAMPLE" \
    --targets Key=InstanceIds,Values=i-1234520122EXAMPLE \
    --task-arn AWS-RestartEC2Instance \
    --service-role-arn arn:aws:iam::111222333444:role/SSM --task-type AUTOMATION
 \
    --task-invocation-parameters "{\"Automation\":{\"DocumentVersion\":\"\$LATEST
\",\"Parameters\":{\"InstanceId\":[\"{{RESOURCE_ID}}\"]}}}" \
    --priority 0 \
    --max-concurrency 1 \
    --max-errors 1 \
    --name "AutomationExample" \
    --description "Restarting EC2 Instance for maintenance"
```

Saída:

```
{
    "WindowTaskId":"11144444-5555-6666-7777-88888888"
}
```

Para obter mais informações, consulte Registrar uma tarefa com a janela de manutenção (AWS CLI) no Guia do usuário do AWS Systems Manager.

Exemplo 2: registrar uma tarefa do Lambda com uma janela de manutenção

O exemplo de `register-task-with-maintenance-window` a seguir registra uma tarefa do Lambda com uma janela de manutenção voltada para uma instância.

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-082dcd7649dee04e4" \
    --targets Key=InstanceIds,Values=i-12344d305eEXAMPLE \
    --task-arn arn:aws:lambda:us-east-1:111222333444:function:SSMTestLAMBDA \
    --service-role-arn arn:aws:iam::111222333444:role/SSM \
    --task-type LAMBDA \
    --task-invocation-parameters '{"Lambda":{"Payload":"{\"InstanceId\":
\"{{RESOURCE_ID}}\",\"targetType\":\"{{TARGET_TYPE}}\"}","Qualifier":"$LATEST"}}'
 \
    --priority 0 \
    --max-concurrency 10 \
    --max-errors 5 \
    --name "Lambda_Example" \
    --description "My Lambda Example"
```

Saída:

```
{
    "WindowTaskId":"22244444-5555-6666-7777-88888888"
}
```

Para obter mais informações, consulte Registrar uma tarefa com a janela de manutenção (AWS CLI) no Guia do usuário do AWS Systems Manager.

Exemplo 3: registrar uma tarefa do Run Command com uma janela de manutenção

O exemplo de `register-task-with-maintenance-window` a seguir registra uma tarefa do Run Command com uma janela de manutenção voltada para uma instância.

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-082dcd7649dee04e4" \
    --targets "Key=InstanceIds,Values=i-12344d305eEXAMPLE" \
    --service-role-arn "arn:aws:iam::111222333444:role/SSM" \
    --task-type "RUN_COMMAND" \
    --name "SSMInstallPowerShellModule" \
    --task-arn "AWS-InstallPowerShellModule" \
    --task-invocation-parameters "{\"RunCommand\":{\"Comment\":\"\",
\"OutputS3BucketName\":\"runcommandlogs\",\"Parameters\":{\"commands\":[\"Get-
Module -ListAvailable\"],\"executionTimeout\":[\"3600\"],\"source\":[\"https:\/
\/gallery.technet.microsoft.com\/EZOut-33ae0fb7\/file\/110351\/1\/EZOut.zip\"],
\"workingDirectory\":[\"\\\\\"]},\"TimeoutSeconds\":600}}" \
    --max-concurrency 1 \
    --max-errors 1 \
    --priority 10
```

Saída:

```
{
    "WindowTaskId":"33344444-5555-6666-7777-88888888"
}
```

Para obter mais informações, consulte Registrar uma tarefa com a janela de manutenção (AWS CLI) no Guia do usuário do AWS Systems Manager.

Exemplo 4: registrar uma tarefa do Step Functions com uma janela de manutenção

O exemplo de `register-task-with-maintenance-window` a seguir registra uma tarefa do Step Functions com uma janela de manutenção voltada para uma instância.

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-1234d787d6EXAMPLE" \
    --targets Key=WindowTargetIds,Values=12347414-69c3-49f8-95b8-ed2dcEXAMPLE \
    --task-arn arn:aws:states:us-
east-1:111222333444:stateMachine:SSMTestStateMachine \
    --service-role-arn arn:aws:iam::111222333444:role/MaintenanceWindows \
    --task-type STEP_FUNCTIONS \
    --task-invocation-parameters '{"StepFunctions":{"Input":"{\"InstanceId\":
\"{{RESOURCE_ID}}\"}"}}' \
    --priority 0 \
    --max-concurrency 10 \
    --max-errors 5 \
```

```
    --name "Step_Functions_Example" \
    --description "My Step Functions Example"
```

Saída:

```
{
    "WindowTaskId":"44444444-5555-6666-7777-88888888"
}
```

Para obter mais informações, consulte Registrar uma tarefa com a janela de manutenção (AWS CLI) no Guia do usuário do AWS Systems Manager.

Exemplo 5: registrar uma tarefa usando um ID de destino de janela de manutenção

O exemplo de `register-task-with-maintenance-window` a seguir registrar uma tarefa usando um ID de destino de janela de manutenção. O ID de destino da janela de manutenção estava presente na saída do comando `aws ssm register-target-with-maintenance-window`. Também é possível recuperá-lo da saída do comando `aws ssm describe-maintenance-window-targets`.

```
aws ssm register-task-with-maintenance-window \
    --targets "Key=WindowTargetIds,Values=350d44e6-28cc-44e2-951f-4b2c9EXAMPLE" \
    --task-arn "AWS-RunShellScript" \
    --service-role-arn "arn:aws:iam::111222333444:role/MaintenanceWindowsRole" \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --task-type "RUN_COMMAND" \
    --task-parameters  "{\"commands\":{\"Values\":[\"df\"]}}" \
    --max-concurrency 1 \
    --max-errors 1 \
    --priority 10
```

Saída:

```
{
    "WindowTaskId":"33344444-5555-6666-7777-88888888"
}
```

Para obter mais informações, consulte Registrar uma tarefa com a janela de manutenção (AWS CLI) no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte RegisterTaskWithMaintenanceWindow na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo registra uma tarefa com uma janela de manutenção usando um ID de instância. A saída é o ID da tarefa.

```
$parameters = @{}
$parameterValues = New-Object
 Amazon.SimpleSystemsManagement.Model.MaintenanceWindowTaskParameterValueExpression
$parameterValues.Values = @("Install")
$parameters.Add("Operation", $parameterValues)

Register-SSMTaskWithMaintenanceWindow -WindowId "mw-03a342e62c96d31b0"
 -ServiceRoleArn "arn:aws:iam::123456789012:role/MaintenanceWindowsRole"
 -MaxConcurrency 1 -MaxError 1 -TaskArn "AWS-RunShellScript" -Target
 @{ Key="InstanceIds";Values="i-0000293ffd8c57862" } -TaskType "RUN_COMMAND" -
Priority 10 -TaskParameter $parameters
```

Saída:

```
f34a2c47-ddfd-4c85-a88d-72366b69af1b
```

Exemplo 2: esse exemplo registra uma tarefa com uma janela de manutenção usando um ID de destino. A saída é o ID da tarefa.

```
$parameters = @{}
$parameterValues = New-Object
 Amazon.SimpleSystemsManagement.Model.MaintenanceWindowTaskParameterValueExpression
$parameterValues.Values = @("Install")
$parameters.Add("Operation", $parameterValues)

register-ssmtaskwithmaintenancewindow -WindowId "mw-03a342e62c96d31b0"
 -ServiceRoleArn "arn:aws:iam::123456789012:role/MaintenanceWindowsRole"
 -MaxConcurrency 1 -MaxError 1 -TaskArn "AWS-RunShellScript" -Target
 @{ Key="WindowTargetIds";Values="350d44e6-28cc-44e2-951f-4b2c985838f6" } -
TaskType "RUN_COMMAND" -Priority 10 -TaskParameter $parameters
```

Saída:

```
f34a2c47-ddfd-4c85-a88d-72366b69af1b
```

Exemplo 3: esse exemplo cria um objeto de parâmetro para o documento de comandos de execução **AWS-RunPowerShellScript** e cria uma tarefa com uma janela de manutenção determinada usando o ID de destino. A saída devolvida é o ID da tarefa.

```
$parameters =
 [Collections.Generic.Dictionary[String,Collections.Generic.List[String]]]::new()
$parameters.Add("commands",@("ipconfig","dir env:\computername"))
$parameters.Add("executionTimeout",@(3600))

$props = @{
    WindowId = "mw-0123e4cce56ff78ae"
    ServiceRoleArn = "arn:aws:iam::123456789012:role/MaintenanceWindowsRole"
    MaxConcurrency = 1
    MaxError = 1
    TaskType = "RUN_COMMAND"
    TaskArn = "AWS-RunPowerShellScript"
    Target =
 @{Key="WindowTargetIds";Values="fe1234ea-56d7-890b-12f3-456b789bee0f"}
    Priority = 1
    RunCommand_Parameter = $parameters
    Name = "set-via-cmdlet"
}

Register-SSMTaskWithMaintenanceWindow @props
```

Saída:

```
f1e2ef34-5678-12e3-456a-12334c5c6cbe
```

Exemplo 4: esse exemplo registra uma tarefa do AWS Systems Manager Automation usando um documento chamado**Create-Snapshots**.

```
$automationParameters = @{}
$automationParameters.Add( "instanceId", @("{{ TARGET_ID }}") )
$automationParameters.Add( "AutomationAssumeRole",
 @("{arn:aws:iam::111111111111:role/AutomationRole}") )
$automationParameters.Add( "SnapshotTimeout", @("PT20M") )
```

```
Register-SSMTaskWithMaintenanceWindow -WindowId mw-123EXAMPLE456`
    -ServiceRoleArn "arn:aws:iam::123456789012:role/MW-Role"`
    -MaxConcurrency 1 -MaxError 1 -TaskArn "CreateVolumeSnapshots"`
    -Target @{ Key="WindowTargetIds";Values="4b5acdf4-946c-4355-
bd68-4329a43a5fd1" }`
    -TaskType "AUTOMATION"`
    -Priority 4`
    -Automation_DocumentVersion '$DEFAULT' -Automation_Parameter
 $automationParameters -Name "Create-Snapshots"
```

- Para obter detalhes da API, consulte RegisterTaskWithMaintenanceWindow na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo registra uma tarefa com uma janela de manutenção usando um ID de instância. A saída é o ID da tarefa.

```
$parameters = @{}
$parameterValues = New-Object
 Amazon.SimpleSystemsManagement.Model.MaintenanceWindowTaskParameterValueExpression
$parameterValues.Values = @("Install")
$parameters.Add("Operation", $parameterValues)

Register-SSMTaskWithMaintenanceWindow -WindowId "mw-03a342e62c96d31b0"
 -ServiceRoleArn "arn:aws:iam::123456789012:role/MaintenanceWindowsRole"
 -MaxConcurrency 1 -MaxError 1 -TaskArn "AWS-RunShellScript" -Target
 @{ Key="InstanceIds";Values="i-0000293ffd8c57862" } -TaskType "RUN_COMMAND" -
Priority 10 -TaskParameter $parameters
```

Saída:

```
f34a2c47-ddfd-4c85-a88d-72366b69af1b
```

Exemplo 2: esse exemplo registra uma tarefa com uma janela de manutenção usando um ID de destino. A saída é o ID da tarefa.

```
$parameters = @{}
$parameterValues = New-Object
 Amazon.SimpleSystemsManagement.Model.MaintenanceWindowTaskParameterValueExpression
$parameterValues.Values = @("Install")
$parameters.Add("Operation", $parameterValues)
```

```
register-ssmtaskwithmaintenancewindow -WindowId "mw-03a342e62c96d31b0"
 -ServiceRoleArn "arn:aws:iam::123456789012:role/MaintenanceWindowsRole"
 -MaxConcurrency 1 -MaxError 1 -TaskArn "AWS-RunShellScript" -Target
 @{ Key="WindowTargetIds";Values="350d44e6-28cc-44e2-951f-4b2c985838f6" } -
TaskType "RUN_COMMAND" -Priority 10 -TaskParameter $parameters
```

Saída:

```
f34a2c47-ddfd-4c85-a88d-72366b69af1b
```

Exemplo 3: esse exemplo cria um objeto de parâmetro para o documento de comandos de execução **AWS-RunPowerShellScript** e cria uma tarefa com uma janela de manutenção determinada usando o ID de destino. A saída devolvida é o ID da tarefa.

```
$parameters =
 [Collections.Generic.Dictionary[String,Collections.Generic.List[String]]]::new()
$parameters.Add("commands",@("ipconfig","dir env:\computername"))
$parameters.Add("executionTimeout",@(3600))

$props = @{
    WindowId = "mw-0123e4cce56ff78ae"
    ServiceRoleArn = "arn:aws:iam::123456789012:role/MaintenanceWindowsRole"
    MaxConcurrency = 1
    MaxError = 1
    TaskType = "RUN_COMMAND"
    TaskArn = "AWS-RunPowerShellScript"
    Target =
 @{Key="WindowTargetIds";Values="fe1234ea-56d7-890b-12f3-456b789bee0f"}
    Priority = 1
    RunCommand_Parameter = $parameters
    Name = "set-via-cmdlet"
}

Register-SSMTaskWithMaintenanceWindow @props
```

Saída:

```
f1e2ef34-5678-12e3-456a-12334c5c6cbe
```

Exemplo 4: esse exemplo registra uma tarefa do AWS Systems Manager Automation usando um documento chamado**Create-Snapshots**.

```
$automationParameters = @{}
$automationParameters.Add( "instanceId", @("{{ TARGET_ID }}") )
$automationParameters.Add( "AutomationAssumeRole",
 @("{arn:aws:iam::111111111111:role/AutomationRole}") )
$automationParameters.Add( "SnapshotTimeout", @("PT20M") )
Register-SSMTaskWithMaintenanceWindow -WindowId mw-123EXAMPLE456`
    -ServiceRoleArn "arn:aws:iam::123456789012:role/MW-Role"`
    -MaxConcurrency 1 -MaxError 1 -TaskArn "CreateVolumeSnapshots"`
    -Target @{ Key="WindowTargetIds";Values="4b5acdf4-946c-4355-
bd68-4329a43a5fd1" }`
    -TaskType "AUTOMATION"`
    -Priority 4`
    -Automation_DocumentVersion '$DEFAULT' -Automation_Parameter
 $automationParameters -Name "Create-Snapshots"
```

- Para obter detalhes da API, consulte RegisterTaskWithMaintenanceWindow na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`RemoveTagsFromResource`** com uma CLI

Os exemplos de código a seguir mostram como usar o `RemoveTagsFromResource`.

CLI

AWS CLI

Remover uma tag de uma lista de referência de patches

O exemplo de `remove-tags-from-resource` a seguir remove duas tags de uma lista de referência de patches.

```
aws ssm remove-tags-from-resource \
    --resource-type "PatchBaseline" \
    --resource-id "pb-0123456789abcdef0" \
    --tag-keys "Region"
```

Este comando não produz saída.

Para obter mais informações, consulte Marcar recursos da AWS na Referência geral da AWS.

- Para obter detalhes da API, consulte RemoveTagsFromResource na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo remove uma tag de uma janela de manutenção. Não haverá saída se o comando for bem-sucedido.

```
Remove-SSMResourceTag -ResourceId "mw-03eb9db42890fb82d" -ResourceType
 "MaintenanceWindow" -TagKey "Production"
```

- Para obter detalhes da API, consulte RemoveTagsFromResource na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo remove uma tag de uma janela de manutenção. Não haverá saída se o comando for bem-sucedido.

```
Remove-SSMResourceTag -ResourceId "mw-03eb9db42890fb82d" -ResourceType
 "MaintenanceWindow" -TagKey "Production"
```

- Para obter detalhes da API, consulte RemoveTagsFromResource na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **SendCommand** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `SendCommand`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Conheça os conceitos básicos

CLI

AWS CLI

Exemplo 1: executar um comando em uma ou mais instâncias remotas

O exemplo de `send-command` a seguir executa um comando `echo` em uma instância de destino.

```
aws ssm send-command \
    --document-name "AWS-RunShellScript" \
    --parameters 'commands=["echo HelloWorld"]' \
    --targets "Key=instanceids,Values=i-1234567890abcdef0" \
    --comment "echo HelloWorld"
```

Saída:

```
{
    "Command": {
        "CommandId": "92853adf-ba41-4cd6-9a88-142d1EXAMPLE",
        "DocumentName": "AWS-RunShellScript",
        "DocumentVersion": "",
        "Comment": "echo HelloWorld",
        "ExpiresAfter": 1550181014.717,
        "Parameters": {
            "commands": [
                "echo HelloWorld"
            ]
        },
        "InstanceIds": [
            "i-0f00f008a2dcbefe2"
        ],
        "Targets": [],
        "RequestedDateTime": 1550173814.717,
        "Status": "Pending",
        "StatusDetails": "Pending",
        "OutputS3BucketName": "",
        "OutputS3KeyPrefix": "",
        "MaxConcurrency": "50",
        "MaxErrors": "0",
```

```
        "TargetCount": 1,
        "CompletedCount": 0,
        "ErrorCount": 0,
        "DeliveryTimedOutCount": 0,
        "ServiceRole": "",
        "NotificationConfig": {
            "NotificationArn": "",
            "NotificationEvents": [],
            "NotificationType": ""
        },
        "CloudWatchOutputConfig": {
            "CloudWatchLogGroupName": "",
            "CloudWatchOutputEnabled": false
        }
    }
}
```

Para obter mais informações, consulte Executar comandos usando o Systems Manager Run Command no Guia do usuário do AWS Systems Manager.

Exemplo 2: obter informações de IP sobre uma instância

O exemplo de `send-command` a seguir retorna as informações de IP`sobre uma instância.

```
aws ssm send-command \
    --instance-ids "i-1234567890abcdef0" \
    --document-name "AWS-RunShellScript" \
    --comment "IP config" \
    --parameters "commands=ifconfig"
```

Consulte a amostra de saída no exemplo 1.

Para obter mais informações, consulte Executar comandos usando o Systems Manager Run Command no Guia do usuário do AWS Systems Manager.

Exemplo 3: executar um comando em instâncias com tags específicas

O exemplo de `send-command` a seguir executa um comando em instâncias que têm a chave de tag "ENV" e o valor "Dev".

```
aws ssm send-command \
    --targets "Key=tag:ENV,Values=Dev" \
    --document-name "AWS-RunShellScript" \
```

```
    --parameters "commands=ifconfig"
```

Consulte a amostra de saída no exemplo 1.

Para obter mais informações, consulte Executar comandos usando o Systems Manager Run Command no Guia do usuário do AWS Systems Manager.

Exemplo 4: executar um comando que envia notificações do SNS

O exemplo de `send-command` a seguir executa um comando que envia notificações do SNS para todos os eventos de notificação e o tipo de notificação `Command`.

```
aws ssm send-command \
    --instance-ids "i-1234567890abcdef0" \
    --document-name "AWS-RunShellScript" \
    --comment "IP config" \
    --parameters "commands=ifconfig" \
    --service-role-arn "arn:aws:iam::123456789012:role/SNS_Role" \
    --notification-config "NotificationArn=arn:aws:sns:us-
east-1:123456789012:SNSTopicName,NotificationEvents=All,NotificationType=Command"
```

Consulte a amostra de saída no exemplo 1.

Para obter mais informações, consulte Executar comandos usando o Systems Manager Run Command no Guia do usuário do AWS Systems Manager.

Exemplo 5: executar um comando que retorna para o S3 e o CloudWatch

O exemplo de `send-command` a seguir executa um comando que envia detalhes do comando para um bucket do S3 e para um grupo de logs do CloudWatch Logs.

```
aws ssm send-command \
    --instance-ids "i-1234567890abcdef0" \
    --document-name "AWS-RunShellScript" \
    --comment "IP config" \
    --parameters "commands=ifconfig" \
    --output-s3-bucket-name "s3-bucket-name" \
    --output-s3-key-prefix "runcommand" \
    --cloud-watch-output-
config "CloudWatchOutputEnabled=true,CloudWatchLogGroupName=CWLGroupName"
```

Consulte a amostra de saída no exemplo 1.

Para obter mais informações, consulte Executar comandos usando o Systems Manager Run Command no Guia do usuário do AWS Systems Manager.

Exemplo 6: executar comandos em várias instâncias com tags diferentes

O exemplo de `send-command` a seguir executa um comando em instâncias com duas chaves e valores de tag diferentes.

```
aws ssm send-command \
    --document-name "AWS-RunPowerShellScript" \
    --parameters commands=["echo helloWorld"] \
    --targets Key=tag:Env,Values=Dev Key=tag:Role,Values=WebServers
```

Consulte a amostra de saída no exemplo 1.

Para obter mais informações, consulte Executar comandos usando o Systems Manager Run Command no Guia do usuário do AWS Systems Manager.

Exemplo 7: usar várias instâncias com a mesma chave de tag

O exemplo de `send-command` a seguir executa um comando em instâncias que têm a mesma chave de tag, mas com valores diferentes.

```
aws ssm send-command \
    --document-name "AWS-RunPowerShellScript" \
    --parameters commands=["echo helloWorld"] \
    --targets Key=tag:Env,Values=Dev,Test
```

Consulte a amostra de saída no exemplo 1.

Para obter mais informações, consulte Executar comandos usando o Systems Manager Run Command no Guia do usuário do AWS Systems Manager.

Exemplo 8: executar um comando que usa um documento compartilhado

O exemplo de `send-command` a seguir executa um comando compartilhado em uma instância de destino.

```
aws ssm send-command \
    --document-name "arn:aws:ssm:us-east-1:123456789012:document/ExampleDocument"
 \
    --targets "Key=instanceids,Values=i-1234567890abcdef0"
```

Consulte a amostra de saída no exemplo 1.

Para obter mais informações, consulte Usar documentos do SSM compartilhados no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte SendCommand na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
    /**
     * Sends a SSM command to a managed node asynchronously.
     *
     * @param documentName The name of the document to use.
     * @param instanceId The ID of the instance to send the command to.
     * @return The command ID.
     * <p>
     * This method initiates asynchronous requests to send a SSM command to a
managed node.
     * It waits until the document is active, sends the command, and checks the
command execution status.
     */
    public String sendSSMCommand(String documentName, String instanceId) throws
InterruptedException, SsmException {
        // Before we use Document to send a command - make sure it is active.
        CompletableFuture<Void> documentActiveFuture =
CompletableFuture.runAsync(() -> {
            boolean isDocumentActive = false;
            DescribeDocumentRequest request = DescribeDocumentRequest.builder()
                    .name(documentName)
                    .build();

            while (!isDocumentActive) {
```

```
                CompletableFuture<DescribeDocumentResponse> response =
getAsyncClient().describeDocument(request);
                String documentStatus =
response.join().document().statusAsString();
                if (documentStatus.equals("Active")) {
                    System.out.println("The SSM document is active and ready to
use.");
                    isDocumentActive = true;
                } else {
                    System.out.println("The SSM document is not active. Status: "
+ documentStatus);
                    try {
                        Thread.sleep(5000);
                    } catch (InterruptedException e) {
                        throw new RuntimeException(e);
                    }
                }
            }
        });

        documentActiveFuture.join();

        // Create the SendCommandRequest.
        SendCommandRequest commandRequest = SendCommandRequest.builder()
                .documentName(documentName)
                .instanceIds(instanceId)
                .build();

        // Send the command.
        CompletableFuture<SendCommandResponse> commandFuture =
getAsyncClient().sendCommand(commandRequest);
        final String[] commandId = {null};

        commandFuture.whenComplete((commandResponse, ex) -> {
            if (commandResponse != null) {
                commandId[0] = commandResponse.command().commandId();
                System.out.println("Command ID: " + commandId[0]);

                // Wait for the command execution to complete.
                GetCommandInvocationRequest invocationRequest =
GetCommandInvocationRequest.builder()
                        .commandId(commandId[0])
                        .instanceId(instanceId)
                        .build();
```

```
                try {
                    System.out.println("Wait 5 secs");
                    TimeUnit.SECONDS.sleep(5);

                    // Retrieve the command execution details.
                    CompletableFuture<GetCommandInvocationResponse>
invocationFuture = getAsyncClient().getCommandInvocation(invocationRequest);
                    invocationFuture.whenComplete((commandInvocationResponse,
invocationEx) -> {
                        if (commandInvocationResponse != null) {
                            // Check the status of the command execution.
                            CommandInvocationStatus status =
commandInvocationResponse.status();
                            if (status == CommandInvocationStatus.SUCCESS) {
                                System.out.println("Command execution
successful");
                            } else {
                                System.out.println("Command execution failed.
Status: " + status);
                            }
                        } else {
                            Throwable invocationCause = (invocationEx instanceof
CompletionException) ? invocationEx.getCause() : invocationEx;
                            throw new CompletionException(invocationCause);
                        }
                    }).join();
                } catch (InterruptedException e) {
                    throw new RuntimeException(e);
                }
            } else {
                Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
                if (cause instanceof SsmException) {
                    throw (SsmException) cause;
                } else {
                    throw new RuntimeException(cause);
                }
            }
        }).join();

        return commandId[0];
    }
```

- Para obter detalhes da API, consulte SendCommand na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
import { SendCommandCommand, SSMClient } from "@aws-sdk/client-ssm";
import { parseArgs } from "node:util";

/**
 * Send an SSM command to a managed node.
 * @param {{ documentName: string }}
 */
export const main = async ({ documentName }) => {
  const client = new SSMClient({});
  try {
    await client.send(
      new SendCommandCommand({
        DocumentName: documentName,
      }),
    );
    console.log("Command sent successfully.");
    return { Success: true };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ValidationError") {
      console.warn(`${caught.message}. Did you provide a valid document name?`);
    } else {
      throw caught;
    }
  }
};
```

- Para obter detalhes da API, consulte SendCommand na Referência da API do AWS SDK para JavaScript.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo executa um comando echo em uma instância de destino.

```
Send-SSMCommand -DocumentName "AWS-RunPowerShellScript" -Parameter @{commands =
 "echo helloWorld"} -Target @{Key="instanceids";Values=@("i-0cb2b964d3e14fd9f")}
```

Saída:

```
CommandId          : d8d190fc-32c1-4d65-a0df-ff5ff3965524
Comment            :
CompletedCount     : 0
DocumentName       : AWS-RunPowerShellScript
ErrorCount         : 0
ExpiresAfter       : 3/7/2017 10:48:37 PM
InstanceIds        : {}
MaxConcurrency     : 50
MaxErrors          : 0
NotificationConfig : Amazon.SimpleSystemsManagement.Model.NotificationConfig
OutputS3BucketName :
OutputS3KeyPrefix  :
OutputS3Region     :
Parameters         : {[commands,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
RequestedDateTime  : 3/7/2017 9:48:37 PM
ServiceRole        :
Status             : Pending
StatusDetails      : Pending
TargetCount        : 0
Targets            : {instanceids}
```

Exemplo 2: esse exemplo mostra como executar um comando que aceita parâmetros aninhados.

```
Send-SSMCommand -DocumentName "AWS-RunRemoteScript" -Parameter
 @{ sourceType="GitHub";sourceInfo='{"owner": "me","repository": "amazon-
```

```
ssm","path": "Examples/Install-Win32OpenSSH"}'; "commandLine"=".\Install-
Win32OpenSSH.ps1"} -InstanceId i-0cb2b964d3e14fd9f
```

- Para obter detalhes da API, consulte SendCommand na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo executa um comando echo em uma instância de destino.

```
Send-SSMCommand -DocumentName "AWS-RunPowerShellScript" -Parameter @{commands =
 "echo helloWorld"} -Target @{Key="instanceids";Values=@("i-0cb2b964d3e14fd9f")}
```

Saída:

```
CommandId          : d8d190fc-32c1-4d65-a0df-ff5ff3965524
Comment            :
CompletedCount     : 0
DocumentName       : AWS-RunPowerShellScript
ErrorCount         : 0
ExpiresAfter       : 3/7/2017 10:48:37 PM
InstanceIds        : {}
MaxConcurrency     : 50
MaxErrors          : 0
NotificationConfig : Amazon.SimpleSystemsManagement.Model.NotificationConfig
OutputS3BucketName :
OutputS3KeyPrefix  :
OutputS3Region     :
Parameters         : {[commands,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
RequestedDateTime  : 3/7/2017 9:48:37 PM
ServiceRole        :
Status             : Pending
StatusDetails      : Pending
TargetCount        : 0
Targets            : {instanceids}
```

Exemplo 2: esse exemplo mostra como executar um comando que aceita parâmetros aninhados.

```
Send-SSMCommand -DocumentName "AWS-RunRemoteScript" -Parameter
 @{ sourceType="GitHub";sourceInfo='{"owner": "me","repository": "amazon-
```

```
ssm","path": "Examples/Install-Win32OpenSSH"}'; "commandLine"=".\Install-
Win32OpenSSH.ps1"} -InstanceId i-0cb2b964d3e14fd9f
```

- Para obter detalhes da API, consulte SendCommand na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
class DocumentWrapper:
    """Encapsulates AWS Systems Manager Document actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.name = None

    @classmethod
    def from_client(cls):
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def send_command(self, instance_ids):
        """
        Sends a command to one or more instances.

        :param instance_ids: The IDs of the instances to send the command to.
        :return: The ID of the command.
        """
        try:
            response = self.ssm_client.send_command(
```

```
                InstanceIds=instance_ids, DocumentName=self.name,
 TimeoutSeconds=3600
            )
            return response["Command"]["CommandId"]
        except ClientError as err:
            logger.error(
                "Couldn't send command to %s. Here's why: %s: %s",
                self.name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Para obter detalhes da API, consulte SendCommand na Referência da API AWS SDK for Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`StartAutomationExecution`** com uma CLI

Os exemplos de código a seguir mostram como usar o `StartAutomationExecution`.

CLI

AWS CLI

Exemplo 1: executar um documento do Automation

O exemplo de `start-automation-execution` a seguir executa um documento do Automation.

```
aws ssm start-automation-execution \
    --document-name "AWS-UpdateLinuxAmi" \
    --parameters "AutomationAssumeRole=arn:aws:iam::123456789012:role/
SSMAutomationRole,SourceAmiId=ami-EXAMPLE,IamInstanceProfileName=EC2InstanceRole"
```

Saída:

```
{
  "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
}
```

Para obter mais informações, consulte Executar um fluxo de trabalho do Automation manualmente no Guia do usuário do AWS Systems Manager.

Exemplo 2: executar um documento do Automation compartilhado

O exemplo de `start-automation-execution` a seguir executa um documento do Automation compartilhado.

```
aws ssm start-automation-execution \
    --document-name "arn:aws:ssm:us-east-1:123456789012:document/ExampleDocument"
```

Saída:

```
{
  "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
}
```

Para obter mais informações, consulte Usar documentos do SSM compartilhados no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte StartAutomationExecution na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo executa um documento especificando um perfil do Automation, um ID de origem da AMI e um perfil de instância do Amazon EC2.

```
Start-SSMAutomationExecution -DocumentName AWS-UpdateLinuxAmi -
Parameter @{'AutomationAssumeRole'='arn:aws:iam::123456789012:role/
SSMAutomationRole';'SourceAmiId'='ami-
f173cc91';'InstanceIamRole'='EC2InstanceRole'}
```

Saída:

```
3a532a4f-0382-11e7-9df7-6f11185f6dd1
```

- Para obter detalhes da API, consulte StartAutomationExecution na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo executa um documento especificando um perfil do Automation, um ID de origem da AMI e um perfil de instância do Amazon EC2.

```
Start-SSMAutomationExecution -DocumentName AWS-UpdateLinuxAmi -
Parameter @{'AutomationAssumeRole'='arn:aws:iam::123456789012:role/
SSMAutomationRole';'SourceAmiId'='ami-
f173cc91';'InstanceIamRole'='EC2InstanceRole'}
```

Saída:

```
3a532a4f-0382-11e7-9df7-6f11185f6dd1
```

- Para obter detalhes da API, consulte StartAutomationExecution na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`StartSession`** com uma CLI

Os exemplos de código a seguir mostram como usar o `StartSession`.

CLI

AWS CLI

Exemplo 1: Para iniciar uma sessão do Session Manager

Este exemplo de `start-session` estabelece uma conexão com uma instância para uma sessão do Session Manager. Observe que esse comando interativo exige que o plug-in do Session Manager seja instalado na máquina-cliente que faz a chamada.

```
aws ssm start-session \
    --target "i-1234567890abcdef0"
```

Saída:

```
Starting session with SessionId: Jane-Roe-07a16060613c408b5
```

Exemplo 2: Para iniciar uma sessão do Session Manager usando SSH

Este exemplo de `start-session` estabelece uma conexão com uma instância para uma sessão do Session Manager usando SSH. Observe que esse comando interativo exige que o plug-in do Session Manager seja instalado na máquina-cliente que faz a chamada e que o comando use o usuário padrão na instância, como `ec2-user` nas instâncias do EC2 para Linux.

```
ssh -i /path/my-key-pair.pem ec2-user@i-02573cafcfEXAMPLE
```

Saída:

```
Starting session with SessionId: ec2-user-07a16060613c408b5
```

Para obter mais informações, consulte Iniciar uma sessão e Instale o plugin do Session Manager para AWS CLI no Guia de usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte StartSession na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: Este exemplo inicia uma conexão com um destino para uma sessão do Session Manager, habilitando o encaminhamento de portas.

```
Start-SSMSession -Target 'i-064578e5e7454488f' -DocumentName 'AWS-
StartPortForwardingSession' -Parameter @{ localPortNumber = '8080'; portNumber =
 '80' }
```

Saída:

```
SessionId    StreamUrl
----------    ----------
random-id0    wss://ssmmessages.amazonaws.com/v1/data-channel/random-id
```

- Para obter detalhes da API, consulte StartSession na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: Este exemplo inicia uma conexão com um destino para uma sessão do Session Manager, habilitando o encaminhamento de portas. Observação: você deverá adicionar o parâmetro Region, se ainda não estiver preparado para usar o AWSCredentials.

```
Start-SSMSession -Target 'i-064578e5e7454488f' -DocumentName 'AWS-
StartPortForwardingSession' -Parameter @{ localPortNumber = '8080'; portNumber =
 '80' } -Region 'us-west-1'
```

Saída:

```
Starting session with SessionId: testUser-xi4glew849asyeryde34u4dfsdfy
Port 8080 opened for sessionId testUser-xi4glew849asyeryde34u4dfsdfy.
Waiting for connections...
```

Exemplo 2: esse exemplo cria uma sessão interativa com uma instância especificada para uma sessão do Session Manager.

```
Start-SSMSession -Target 'i-1234567890abcdef0' -Region 'us-west-1'
```

Saída:

```
Starting session with SessionId : testUser-xi4glew849asyeryde34u4dfsdfy
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.



Install the latest PowerShell for new features and improvements!


PS C:\Windows\system32> whoami
```

```
ec2amaz-fnsdrwv\ec2-test-user
PS C:\Windows\system32>
```

Exemplo 3: esse exemplo cria uma sessão sem se conectar a ela e retorna as propriedades SessionId, StreamUrl e TokenValue necessárias para se conectar à sessão.

```
Start-SSMSession -Target 'i-1234567890abcdef0' -Region 'us-west-1' -
DisablePluginInvocation
```

Saída:

```
SessionId        : testUser-xi4glew849asyeryde34u4dfsdfy
StreamUrl        : {StreamUrl value redacted}
TokenValue       : {Token value redacted}
ContentLength    : 1207
HttpStatusCode   : OK
```

- Para obter detalhes da API, consulte StartSession na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`StopAutomationExecution`** com uma CLI

Os exemplos de código a seguir mostram como usar o `StopAutomationExecution`.

CLI

AWS CLI

Para interromper a execução de uma automação

O exemplo de `stop-automation-execution` a seguir interrompe um documento do Automation.

```
aws ssm stop-automation-execution
    --automation-execution-id "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
```

Este comando não produz saída.

Para obter mais informações, consulte Executar um fluxo de trabalho do Automation manualmente no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte StopAutomationExecution na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo interrompe uma execução do Automation. Não haverá saída se o comando for bem-sucedido.

```
Stop-SSMAutomationExecution -AutomationExecutionId "4105a4fc-
f944-11e6-9d32-8fb2db27a909"
```

- Para obter detalhes da API, consulte StopAutomationExecution na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo interrompe uma execução do Automation. Não haverá saída se o comando for bem-sucedido.

```
Stop-SSMAutomationExecution -AutomationExecutionId "4105a4fc-
f944-11e6-9d32-8fb2db27a909"
```

- Para obter detalhes da API, consulte StopAutomationExecution na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`UpdateAssociation`** com uma CLI

Os exemplos de código a seguir mostram como usar o `UpdateAssociation`.

CLI

AWS CLI

Exemplo 1: atualizar uma associação de documentos

O exemplo de `update-association` a seguir atualiza uma associação com uma nova versão de documento.

```
aws ssm update-association \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --document-version "\$LATEST"
```

Saída:

```
{
    "AssociationDescription": {
        "Name": "AWS-UpdateSSMAgent",
        "AssociationVersion": "2",
        "Date": 1550508093.293,
        "LastUpdateAssociationDate": 1550508106.596,
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "DocumentVersion": "$LATEST",
        "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
        "Targets": [
            {
                "Key": "tag:Name",
                "Values": [
                    "Linux"
                ]
            }
        ],
        "LastExecutionDate": 1550508094.879,
        "LastSuccessfulExecutionDate": 1550508094.879
    }
}
```

Para obter mais informações, consulte Editar e criar uma nova versão de uma associação no Guia do usuário do AWS Systems Manager.

Exemplo 2: atualizar a expressão de programação de uma associação

O exemplo de `update-association` a seguir atualiza a expressão de programação para a associação especificada.

```
aws ssm update-association \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --schedule-expression "cron(0 0 0/4 1/1 * ? *)"
```

Saída:

```
{
    "AssociationDescription": {
        "Name": "AWS-HelloWorld",
        "AssociationVersion": "2",
        "Date": "2021-02-08T13:54:19.203000-08:00",
        "LastUpdateAssociationDate": "2021-06-29T11:51:07.933000-07:00",
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "DocumentVersion": "$DEFAULT",
        "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
        "Targets": [
            {
                "Key": "aws:NoOpAutomationTag",
                "Values": [
                    "AWS-NoOpAutomationTarget-Value"
                ]
            }
        ],
        "ScheduleExpression": "cron(0 0 0/4 1/1 * ? *)",
        "LastExecutionDate": "2021-06-26T19:00:48.110000-07:00",
        "ApplyOnlyAtCronInterval": false
    }
}
```

Para obter mais informações, consulte Editar e criar uma nova versão de uma associação no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte UpdateAssociation na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo atualiza uma associação com uma nova versão de documento.

```
Update-SSMAssociation -AssociationId "93285663-92df-44cb-9f26-2292d4ecc439" -
DocumentVersion "1"
```

Saída:

```
Name                 : AWS-UpdateSSMAgent
InstanceId           :
Date                 : 3/1/2017 6:22:21 PM
Status.Name          :
Status.Date          :
Status.Message       :
Status.AdditionalInfo :
```

- Para obter detalhes sobre a API, consulte UpdateAssociation na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo atualiza uma associação com uma nova versão de documento.

```
Update-SSMAssociation -AssociationId "93285663-92df-44cb-9f26-2292d4ecc439" -
DocumentVersion "1"
```

Saída:

```
Name                 : AWS-UpdateSSMAgent
InstanceId           :
Date                 : 3/1/2017 6:22:21 PM
Status.Name          :
Status.Date          :
Status.Message       :
Status.AdditionalInfo :
```

- Para obter detalhes sobre a API, consulte UpdateAssociation na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`UpdateAssociationStatus`** com uma CLI

Os exemplos de código a seguir mostram como usar o `UpdateAssociationStatus`.

CLI

AWS CLI

Para atualizar o status da associação

O exemplo de `update-association-status` a seguir atualiza o status da associação entre uma instância e um documento.

```
aws ssm update-association-status \
    --name "AWS-UpdateSSMAgent" \
    --instance-id "i-1234567890abcdef0" \
    --association-
status "Date=1424421071.939,Name=Pending,Message=temp_status_change,AdditionalInfo=Additi
Config-Needed"
```

Saída:

```
{
    "AssociationDescription": {
        "Name": "AWS-UpdateSSMAgent",
        "InstanceId": "i-1234567890abcdef0",
        "AssociationVersion": "1",
        "Date": 1550507529.604,
        "LastUpdateAssociationDate": 1550507806.974,
        "Status": {
            "Date": 1424421071.0,
            "Name": "Pending",
            "Message": "temp_status_change",
            "AdditionalInfo": "Additional-Config-Needed"
        },
        "Overview": {
            "Status": "Success",
            "AssociationStatusAggregatedCount": {
```

```
                "Success": 1
            }
        },
        "DocumentVersion": "$DEFAULT",
        "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
        "Targets": [
            {
                "Key": "InstanceIds",
                "Values": [
                    "i-1234567890abcdef0"
                ]
            }
        ],
        "LastExecutionDate": 1550507808.0,
        "LastSuccessfulExecutionDate": 1550507808.0
    }
}
```

Para obter mais informações, consulte Trabalhar com associações no Systems Manager no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte UpdateAssociationStatus na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo atualiza o status da associação entre uma instância e um documento de configuração.

```
Update-SSMAssociationStatus -Name "AWS-UpdateSSMAgent" -InstanceId
 "i-0000293ffd8c57862" -AssociationStatus_Date "2015-02-20T08:31:11Z"
 -AssociationStatus_Name "Pending" -AssociationStatus_Message
 "temporary_status_change" -AssociationStatus_AdditionalInfo "Additional-Config-
Needed"
```

Saída:

```
Name                 : AWS-UpdateSSMAgent
InstanceId           : i-0000293ffd8c57862
Date                 : 2/23/2017 6:55:22 PM
```

```
Status.Name           : Pending
Status.Date           : 2/20/2015 8:31:11 AM
Status.Message        : temporary_status_change
Status.AdditionalInfo : Additional-Config-Needed
```

- Para obter detalhes sobre a API, consulte UpdateAssociationStatus na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo atualiza o status da associação entre uma instância e um documento de configuração.

```
Update-SSMAssociationStatus -Name "AWS-UpdateSSMAgent" -InstanceId
 "i-0000293ffd8c57862" -AssociationStatus_Date "2015-02-20T08:31:11Z"
 -AssociationStatus_Name "Pending" -AssociationStatus_Message
 "temporary_status_change" -AssociationStatus_AdditionalInfo "Additional-Config-
Needed"
```

Saída:

```
Name                  : AWS-UpdateSSMAgent
InstanceId            : i-0000293ffd8c57862
Date                  : 2/23/2017 6:55:22 PM
Status.Name           : Pending
Status.Date           : 2/20/2015 8:31:11 AM
Status.Message        : temporary_status_change
Status.AdditionalInfo : Additional-Config-Needed
```

- Para obter detalhes sobre a API, consulte UpdateAssociationStatus na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`UpdateDocument`** com uma CLI

Os exemplos de código a seguir mostram como usar o `UpdateDocument`.

CLI

AWS CLI

Para criar uma nova versão de um documento

O exemplo de `update-document` a seguir cria uma nova versão de um documento quando executado em um computador Windows. O documento especificado por `--document` deve estar em formato JSON. Observe que `file://` deve ser referenciado seguido pelo caminho do arquivo de conteúdo. Em função disso do $ no início do parâmetro `--document-version`, o valor deve ser colocado entre aspas duplas no Windows. No Linux, MacOS ou em um prompt do PowerShell, o valor deve ser colocado entre aspas simples.

Versão do Windows:

```
aws ssm update-document \
    --name "RunShellScript" \
    --content "file://RunShellScript.json" \
    --document-version "$LATEST"
```

Versão do Linux/Mac:

```
aws ssm update-document \
    --name "RunShellScript" \
    --content "file://RunShellScript.json" \
    --document-version '$LATEST'
```

Saída:

```
{
  "DocumentDescription": {
      "Status": "Updating",
      "Hash": "f775e5df4904c6fa46686c4722fae9de1950dace25cd9608ff8d622046b68d9b",
      "Name": "RunShellScript",
      "Parameters": [
          {
              "Type": "StringList",
              "Name": "commands",
              "Description": "(Required) Specify a shell script or a command to
 run."
          }
```

```
      ],
      "DocumentType": "Command",
      "PlatformTypes": [
          "Linux"
      ],
      "DocumentVersion": "2",
      "HashType": "Sha256",
      "CreatedDate": 1487899655.152,
      "Owner": "809632081692",
      "SchemaVersion": "2.0",
      "DefaultVersion": "1",
      "LatestVersion": "2",
      "Description": "Run an updated script"
  }
}
```

- Para obter detalhes da API, consulte UpdateDocument na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: isso cria uma nova versão de um documento com o conteúdo atualizado do arquivo json que você especificar. O documento deve estar em formato JSON. É possível obter a versão do documento com o cmdlet "Get-SSMDocumentVersionList".

```
Update-SSMDocument -Name RunShellScript -DocumentVersion "1" -Content (Get-
Content -Raw "c:\temp\RunShellScript.json")
```

Saída:

```
CreatedDate     : 3/1/2017 2:59:17 AM
DefaultVersion  : 1
Description     : Run an updated script
DocumentType    : Command
DocumentVersion : 2
Hash            :
 1d5ce820e999ff051eb4841ed887593daf77120fd76cae0d18a53cc42e4e22c1
HashType        : Sha256
LatestVersion   : 2
```

```
Name            : RunShellScript
Owner           : 809632081692
Parameters      : {commands}
PlatformTypes   : {Linux}
SchemaVersion   : 2.0
Sha1            :
Status          : Updating
```

- Para obter detalhes da API, consulte UpdateDocument na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: isso cria uma nova versão de um documento com o conteúdo atualizado do arquivo json que você especificar. O documento deve estar em formato JSON. É possível obter a versão do documento com o cmdlet "Get-SSMDocumentVersionList".

```
Update-SSMDocument -Name RunShellScript -DocumentVersion "1" -Content (Get-
Content -Raw "c:\temp\RunShellScript.json")
```

Saída:

```
CreatedDate     : 3/1/2017 2:59:17 AM
DefaultVersion  : 1
Description     : Run an updated script
DocumentType    : Command
DocumentVersion : 2
Hash            :
 1d5ce820e999ff051eb4841ed887593daf77120fd76cae0d18a53cc42e4e22c1
HashType        : Sha256
LatestVersion   : 2
Name            : RunShellScript
Owner           : 809632081692
Parameters      : {commands}
PlatformTypes   : {Linux}
SchemaVersion   : 2.0
Sha1            :
Status          : Updating
```

- Para obter detalhes da API, consulte UpdateDocument na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`UpdateDocumentDefaultVersion`** com uma CLI

Os exemplos de código a seguir mostram como usar o `UpdateDocumentDefaultVersion`.

CLI

AWS CLI

Para atualizar a versão padrão de um documento

O exemplo de `update-document-default-version` a seguir atualiza a versão padrão de um documento do Systems Manager.

```
aws ssm update-document-default-version \
    --name "Example" \
    --document-version "2"
```

Saída:

```
{
    "Description": {
        "Name": "Example",
        "DefaultVersion": "2"
    }
}
```

Para obter mais informações, consulte Escrever conteúdo de documentos do SSM no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte UpdateDocumentDefaultVersion na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo atualiza a versão padrão de um documento. Você pode obter as versões do documento disponíveis com o cmdlet "Get-SSMDocumentVersionList".

```
Update-SSMDocumentDefaultVersion -Name "RunShellScript" -DocumentVersion "2"
```

Saída:

```
DefaultVersion Name
-------------- ----
2              RunShellScript
```

- Para obter detalhes da API, consulte UpdateDocumentDefaultVersion na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo atualiza a versão padrão de um documento. Você pode obter as versões do documento disponíveis com o cmdlet "Get-SSMDocumentVersionList".

```
Update-SSMDocumentDefaultVersion -Name "RunShellScript" -DocumentVersion "2"
```

Saída:

```
DefaultVersion Name
-------------- ----
2              RunShellScript
```

- Para obter detalhes da API, consulte UpdateDocumentDefaultVersion na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`UpdateMaintenanceWindow`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `UpdateMaintenanceWindow`.

CLI

AWS CLI

Exemplo 1: atualizar uma janela de manutenção

O exemplo de `update-maintenance-window` a seguir atualiza o nome de uma janela de manutenção.

```
aws ssm update-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9" \
    --name "My-Renamed-MW"
```

Saída:

```
{
    "Cutoff": 1,
    "Name": "My-Renamed-MW",
    "Schedule": "cron(0 16 ? * TUE *)",
    "Enabled": true,
    "AllowUnassociatedTargets": true,
    "WindowId": "mw-1a2b3c4d5e6f7g8h9",
    "Duration": 4
}
```

Exemplo 2: desabilitar uma janela de manutenção

O exemplo de `update-maintenance-window` a seguir desabilita uma janela de manutenção.

```
aws ssm update-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9" \
    --no-enabled
```

Exemplo 3: habilitar uma janela de manutenção

O exemplo de `update-maintenance-window` a seguir habilita uma janela de manutenção.

```
aws ssm update-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9" \
    --enabled
```

Para obter mais informações, consulte Atualizar uma janela de manutenção (AWS CLI) no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte UpdateMaintenanceWindow na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
   /**
    * Updates an SSM maintenance window asynchronously.
    *
    * @param id The ID of the maintenance window to update.
    * @param name The new name for the maintenance window.
    * <p>
    * This method initiates an asynchronous request to update an SSM maintenance
window.
    * If the request is successful, it prints a success message.
    * If an exception occurs, it handles the error appropriately.
    */
   public void updateSSMMaintenanceWindow(String id, String name) throws
SsmException {
       UpdateMaintenanceWindowRequest updateRequest =
UpdateMaintenanceWindowRequest.builder()
               .windowId(id)
               .allowUnassociatedTargets(true)
               .duration(24)
               .enabled(true)
               .name(name)
               .schedule("cron(0 0 ? * MON *)")
               .build();

       CompletableFuture<UpdateMaintenanceWindowResponse> future =
getAsyncClient().updateMaintenanceWindow(updateRequest);
       future.whenComplete((response, ex) -> {
           if (response != null) {
               System.out.println("The SSM maintenance window was successfully
updated");
           } else {
               Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
```

```
                if (cause instanceof SsmException) {
                    throw new CompletionException(cause);
                } else {
                    throw new RuntimeException(cause);
                }
            }
        }).join();
    }
```

- Para obter detalhes da API, consulte UpdateMaintenanceWindow na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
import { UpdateMaintenanceWindowCommand, SSMClient } from "@aws-sdk/client-ssm";
import { parseArgs } from "node:util";

/**
 * Update an SSM maintenance window.
 * @param {{ windowId: string, allowUnassociatedTargets?: boolean, duration?:
 number, enabled?: boolean, name?: string, schedule?: string }}
 */
export const main = async ({
  windowId,
  allowUnassociatedTargets = undefined, //Allow the maintenance window to run on
 managed nodes, even if you haven't registered those nodes as targets.
  duration = undefined, //The duration of the maintenance window in hours.
  enabled = undefined,
  name = undefined,
  schedule = undefined, //The schedule of the maintenance window in the form of a
 cron or rate expression.
}) => {
```

```
  const client = new SSMClient({});
  try {
    const { opsItemArn, opsItemId } = await client.send(
      new UpdateMaintenanceWindowCommand({
        WindowId: windowId,
        AllowUnassociatedTargets: allowUnassociatedTargets,
        Duration: duration,
        Enabled: enabled,
        Name: name,
        Schedule: schedule,
      }),
    );
    console.log("Maintenance window updated.");
    return { OpsItemArn: opsItemArn, OpsItemId: opsItemId };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ValidationError") {
      console.warn(`${caught.message}. Are these values correct?`);
    } else {
      throw caught;
    }
  }
};
```

- Para obter detalhes da API, consulte UpdateMaintenanceWindow na Referência da API do AWS SDK para JavaScript.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo atualiza o nome de uma janela de manutenção.

```
Update-SSMMaintenanceWindow -WindowId "mw-03eb9db42890fb82d" -Name "My-Renamed-
MW"
```

Saída:

```
AllowUnassociatedTargets : False
Cutoff                   : 1
Duration                 : 2
Enabled                  : True
```

```
Name                        : My-Renamed-MW
Schedule                    : cron(0 */30 * * * ? *)
WindowId                    : mw-03eb9db42890fb82d
```

Exemplo 2: esse exemplo habilita uma janela de manutenção.

```
Update-SSMMaintenanceWindow -WindowId "mw-03eb9db42890fb82d" -Enabled $true
```

Saída:

```
AllowUnassociatedTargets : False
Cutoff                   : 1
Duration                 : 2
Enabled                  : True
Name                     : My-Renamed-MW
Schedule                 : cron(0 */30 * * * ? *)
WindowId                 : mw-03eb9db42890fb82d
```

Exemplo 3: esse exemplo desabilita uma janela de manutenção.

```
Update-SSMMaintenanceWindow -WindowId "mw-03eb9db42890fb82d" -Enabled $false
```

Saída:

```
AllowUnassociatedTargets : False
Cutoff                   : 1
Duration                 : 2
Enabled                  : False
Name                     : My-Renamed-MW
Schedule                 : cron(0 */30 * * * ? *)
WindowId                 : mw-03eb9db42890fb82d
```

- Para obter detalhes da API, consulte UpdateMaintenanceWindow na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo atualiza o nome de uma janela de manutenção.

```
Update-SSMMaintenanceWindow -WindowId "mw-03eb9db42890fb82d" -Name "My-Renamed-
MW"
```

Saída:

```
AllowUnassociatedTargets : False
Cutoff                   : 1
Duration                 : 2
Enabled                  : True
Name                     : My-Renamed-MW
Schedule                 : cron(0 */30 * * * ? *)
WindowId                 : mw-03eb9db42890fb82d
```

Exemplo 2: esse exemplo habilita uma janela de manutenção.

```
Update-SSMMaintenanceWindow -WindowId "mw-03eb9db42890fb82d" -Enabled $true
```

Saída:

```
AllowUnassociatedTargets : False
Cutoff                   : 1
Duration                 : 2
Enabled                  : True
Name                     : My-Renamed-MW
Schedule                 : cron(0 */30 * * * ? *)
WindowId                 : mw-03eb9db42890fb82d
```

Exemplo 3: esse exemplo desabilita uma janela de manutenção.

```
Update-SSMMaintenanceWindow -WindowId "mw-03eb9db42890fb82d" -Enabled $false
```

Saída:

```
AllowUnassociatedTargets : False
Cutoff                   : 1
Duration                 : 2
Enabled                  : False
Name                     : My-Renamed-MW
Schedule                 : cron(0 */30 * * * ? *)
WindowId                 : mw-03eb9db42890fb82d
```

- Para obter detalhes da API, consulte UpdateMaintenanceWindow na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
class MaintenanceWindowWrapper:
    """Encapsulates AWS Systems Manager maintenance window actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.window_id = None
        self.name = None

    @classmethod
    def from_client(cls):
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def update(
        self, name, enabled, schedule, duration, cutoff,
 allow_unassociated_targets
    ):
        """
        Update an AWS Systems Manager maintenance window.

        :param name: The name of the maintenance window.
        :param enabled: Whether the maintenance window is enabled to run on
 managed nodes.
        :param schedule: The schedule of the maintenance window.
        :param duration: The duration of the maintenance window.
        :param cutoff: The cutoff time of the maintenance window.
        :param allow_unassociated_targets: Allow the maintenance window to run on
 managed nodes, even
```

```
                                        if you haven't registered those nodes
as targets.
        """
        try:
            self.ssm_client.update_maintenance_window(
                WindowId=self.window_id,
                Name=name,
                Enabled=enabled,
                Schedule=schedule,
                Duration=duration,
                Cutoff=cutoff,
                AllowUnassociatedTargets=allow_unassociated_targets,
            )
            self.name = name
            logger.info("Updated maintenance window %s.", self.window_id)
        except ParamValidationError as error:
            logger.error(
                "Parameter validation error when trying to update maintenance
window %s. Here's why: %s",
                self.window_id,
                error,
            )
            raise
        except ClientError as err:
            logger.error(
                "Couldn't update maintenance window %s. Here's why: %s: %s",
                self.name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Para obter detalhes da API, consulte UpdateMaintenanceWindow na Referência da API AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **UpdateManagedInstanceRole** com uma CLI

Os exemplos de código a seguir mostram como usar o `UpdateManagedInstanceRole`.

CLI

AWS CLI

Para atualizar o perfil do IAM de uma instância gerenciada

O exemplo de `update-managed-instance-role` a seguir atualiza o perfil da instância do IAM de uma instância gerenciada.

```
aws ssm update-managed-instance-role \
    --instance-id "mi-08ab247cdfEXAMPLE" \
    --iam-role "ExampleRole"
```

Este comando não produz saída.

Para obter mais informações, consulte Etapa 4: criar um perfil de instância do IAM para o Systems Manager no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte UpdateManagedInstanceRole na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo atualiza o perfil de uma instância gerenciada. Não haverá saída se o comando for bem-sucedido.

```
Update-SSMManagedInstanceRole -InstanceId "mi-08ab247cdf1046573" -IamRole
 "AutomationRole"
```

- Para obter detalhes da API, consulte UpdateManagedInstanceRole na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo atualiza o perfil de uma instância gerenciada. Não haverá saída se o comando for bem-sucedido.

```
Update-SSMManagedInstanceRole -InstanceId "mi-08ab247cdf1046573" -IamRole
 "AutomationRole"
```

- Para obter detalhes da API, consulte UpdateManagedInstanceRole na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`UpdateOpsItem`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `UpdateOpsItem`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Conheça os conceitos básicos

CLI

AWS CLI

Como atualizar um OpsItem

O exemplo de `update-ops-item` a seguir atualiza a descrição, a prioridade e a categoria de um OpsItem. Além disso, o comando especifica um tópico do SNS para o qual as notificações são enviadas quando esse OpsItem é editado ou alterado.

```
aws ssm update-ops-item \
    --ops-item-id "oi-287b5EXAMPLE" \
    --description "Primary OpsItem for failover event 2020-01-01-fh398yf" \
    --priority 2 \
    --category "Security" \
    --notifications "Arn=arn:aws:sns:us-east-2:111222333444:my-us-east-2-topic"
```

Saída:

```
This command produces no output.
```

Para obter mais informações, consulte Gerenciamento de OpsItems no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte UpdateOpsItem na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
    /**
     * Resolves an AWS SSM OpsItem asynchronously.
     *
     * @param opsID The ID of the OpsItem to resolve.
     * <p>
     * This method initiates an asynchronous request to resolve an SSM OpsItem.
     * If an exception occurs, it handles the error appropriately.
     */
    public void resolveOpsItem(String opsID) {
        UpdateOpsItemRequest opsItemRequest = UpdateOpsItemRequest.builder()
                .opsItemId(opsID)
                .status(OpsItemStatus.RESOLVED)
                .build();

        CompletableFuture<Void> future = CompletableFuture.runAsync(() -> {
            getAsyncClient().updateOpsItem(opsItemRequest)
                    .thenAccept(response -> {
                        System.out.println("OpsItem resolved successfully.");
                    })
                    .exceptionally(ex -> {
                        throw new CompletionException(ex);
                    }).join();
        }).exceptionally(ex -> {
            Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
            if (cause instanceof SsmException) {
```

```
                throw new RuntimeException("SSM error: " + cause.getMessage(),
 cause);
            } else {
                throw new RuntimeException("Unexpected error: " +
 cause.getMessage(), cause);
            }
        });

        try {
            future.join();
        } catch (CompletionException ex) {
            throw ex.getCause() instanceof RuntimeException ? (RuntimeException)
 ex.getCause() : ex;
        }
    }
```

- Para obter detalhes da API, consulte UpdateOpsItem na Referência da API AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
import { UpdateOpsItemCommand, SSMClient } from "@aws-sdk/client-ssm";
import { parseArgs } from "node:util";

/**
 * Update an SSM OpsItem.
 * @param {{ opsItemId: string, status?: OpsItemStatus }}
 */
export const main = async ({
  opsItemId,
  status = undefined, // The OpsItem status. Status can be Open, In Progress, or
 Resolved
```

```
}) => {
  const client = new SSMClient({});
  try {
    await client.send(
      new UpdateOpsItemCommand({
        OpsItemId: opsItemId,
        Status: status,
      }),
    );
    console.log("Ops item updated.");
    return { Success: true };
  } catch (caught) {
    if (
      caught instanceof Error &&
      caught.name === "OpsItemLimitExceededException"
    ) {
      console.warn(
        `Couldn't create ops item because you have exceeded your open OpsItem
 limit. ${caught.message}.`,
      );
    } else {
      throw caught;
    }
  }
};
```

- Para obter detalhes da API, consulte UpdateOpsItem na Referência da API AWS SDK para JavaScript.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWSCode Examples Repository.

```
class OpsItemWrapper:
```

```
    """Encapsulates AWS Systems Manager OpsItem actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.id = None

    @classmethod
    def from_client(cls):
        """
        :return: A OpsItemWrapper instance.
        """
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def update(self, title=None, description=None, status=None):
        """
        Update an OpsItem.

        :param title: The new OpsItem title.
        :param description: The new OpsItem description.
        :param status: The new OpsItem status.
        :return:
        """
        args = dict(OpsItemId=self.id)
        if title is not None:
            args["Title"] = title
        if description is not None:
            args["Description"] = description
        if status is not None:
            args["Status"] = status
        try:
            self.ssm_client.update_ops_item(**args)
        except ClientError as err:
            logger.error(
                "Couldn't update ops item %s. Here's why: %s: %s",
                self.id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Para obter detalhes da API, consulte UpdateOpsItem na Referência da API AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`UpdatePatchBaseline`** com uma CLI

Os exemplos de código a seguir mostram como usar o `UpdatePatchBaseline`.

CLI

AWS CLI

Exemplo 1: atualizar uma lista de referência de patches

O exemplo de `update-patch-baseline` a seguir adiciona os dois patches especificados como rejeitados e um patch como aprovado à lista de referência de patches especificada.

```
aws ssm update-patch-baseline \
        --baseline-id "pb-0123456789abcdef0" \
        --rejected-patches "KB2032276" "MS10-048" \
        --approved-patches "KB2124261"
```

Saída:

```
{
    "BaselineId": "pb-0123456789abcdef0",
    "Name": "WindowsPatching",
    "OperatingSystem": "WINDOWS",
    "GlobalFilters": {
        "PatchFilters": []
    },
    "ApprovalRules": {
        "PatchRules": [
            {
                "PatchFilterGroup": {
                    "PatchFilters": [
```

```
                        {
                            "Key": "PRODUCT",
                            "Values": [
                                "WindowsServer2016"
                            ]
                        }
                    ]
                },
                "ComplianceLevel": "CRITICAL",
                "ApproveAfterDays": 0,
                "EnableNonSecurity": false
            }
        ]
    },
    "ApprovedPatches": [
        "KB2124261"
    ],
    "ApprovedPatchesComplianceLevel": "UNSPECIFIED",
    "ApprovedPatchesEnableNonSecurity": false,
    "RejectedPatches": [
        "KB2032276",
        "MS10-048"
    ],
    "RejectedPatchesAction": "ALLOW_AS_DEPENDENCY",
    "CreatedDate": 1550244180.465,
    "ModifiedDate": 1550244180.465,
    "Description": "Patches for Windows Servers",
    "Sources": []
}
```

Exemplo 2: renomear uma lista de referência de patches

O exemplo de `update-patch-baseline` a seguir renomeia lista de referência de patches especificada.

```
aws ssm update-patch-baseline \
    --baseline-id "pb-0713accee01234567" \
    --name "Windows-Server-2012-R2-Important-and-Critical-Security-Updates"
```

Para obter mais informações, consulte Atualizar ou excluir uma lista de referência de patches <https://docs.aws.amazon.com/systems-manager/latest/userguide/patch-baseline-update-or-delete.html>`__ no Guia do usuário do AWS Systems Manager.

- Para obter detalhes da API, consulte UpdatePatchBaseline na Referência de comandos da AWS CLI.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse exemplo adiciona dois patches como rejeitados e um patch como aprovado a uma lista de referência de patches existente.

```
Update-SSMPatchBaseline -BaselineId "pb-03da896ca3b68b639" -RejectedPatch
 "KB2032276","MS10-048" -ApprovedPatch "KB2124261"
```

Saída:

```
ApprovalRules   : Amazon.SimpleSystemsManagement.Model.PatchRuleGroup
ApprovedPatches : {KB2124261}
BaselineId      : pb-03da896ca3b68b639
CreatedDate     : 3/3/2017 5:02:19 PM
Description     : Baseline containing all updates approved for production systems
GlobalFilters   : Amazon.SimpleSystemsManagement.Model.PatchFilterGroup
ModifiedDate    : 3/3/2017 5:22:10 PM
Name            : Production-Baseline
RejectedPatches : {KB2032276, MS10-048}
```

- Para obter detalhes da API, consulte UpdatePatchBaseline na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse exemplo adiciona dois patches como rejeitados e um patch como aprovado a uma lista de referência de patches existente.

```
Update-SSMPatchBaseline -BaselineId "pb-03da896ca3b68b639" -RejectedPatch
 "KB2032276","MS10-048" -ApprovedPatch "KB2124261"
```

Saída:

```
ApprovalRules   : Amazon.SimpleSystemsManagement.Model.PatchRuleGroup
ApprovedPatches : {KB2124261}
BaselineId      : pb-03da896ca3b68b639
```

```
CreatedDate     : 3/3/2017 5:02:19 PM
Description     : Baseline containing all updates approved for production systems
GlobalFilters   : Amazon.SimpleSystemsManagement.Model.PatchFilterGroup
ModifiedDate    : 3/3/2017 5:22:10 PM
Name            : Production-Baseline
RejectedPatches : {KB2032276, MS10-048}
```

- Para obter detalhes da API, consulte UpdatePatchBaseline na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Para ver uma lista completa dos Guias do desenvolvedor e exemplos de código do SDK da AWS, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Registrar em log e monitorar no AWS Systems Manager

O monitoramento é uma parte importante da manutenção da confiabilidade, da disponibilidade e do desempenho do AWS Systems Manager e de soluções da AWS. Você deve coletar dados de monitoramento de todas as partes da solução da AWS para facilitar a depuração de uma falha de vários pontos, caso ela ocorra. Antes de começar a monitorar o Systems Manager, crie um plano de monitoramento que inclua as respostas para as seguintes perguntas:

- Quais são seus objetivos de monitoramento?
- Quais recursos você vai monitorar?
- Com que frequência você vai monitorar esses recursos?
- Quais ferramentas de monitoramento você usará?
- Quem realiza as tarefas de monitoramento?
- Quem deve ser notificado quando algo der errado?

Depois de definir seus objetivos de monitoramento e criar seu plano de monitoramento, a próxima etapa é estabelecer uma linha de base para a performance normal do Systems Manager em seu ambiente. Você deve medir a performance do Systems Manager em vários momentos e em condições de cargas diferentes. Ao monitorar o Systems Manager, você deve armazenar um histórico de dados de monitoramento coletados. Você poderá comparar a performance atual do Systems Manager com esses dados históricos para ajudar a identificar padrões de performance normais e anomalias de performance, e criar métodos para resolvê-los.

Por exemplo, você pode monitorar o sucesso ou a falha de operações, como fluxos de trabalho de automação, a aplicação de linhas de base de patch, eventos de janela de manutenção e conformidade de configuração. O Automation é uma ferramenta do AWS Systems Manager.

Você também pode monitorar a utilização da CPU, a E/S do disco e a utilização da rede dos nós gerenciados. Quando a performance estiver fora da linha de base estabelecida, talvez seja necessário reconfigurar ou otimizar o nó para reduzir a utilização da CPU, melhorar a E/S de disco ou reduzir o tráfego de rede. Para obter mais informações sobre monitoramento de instância do EC2, consulte Monitorar o Amazon EC2 no Guia do usuário do Amazon EC2.

Tópicos

- Ferramentas de monitoramento
- Enviar logs de nós para o CloudWatch Logs unificado (agente do CloudWatch)

- Enviar logs do SSM Agent ao CloudWatch Logs
- Monitoramento dos seus eventos de solicitação de alteração
- Monitoramento das automações
- Monitorar métricas do Run Command com o Amazon CloudWatch
- Registrar em log chamadas de API do AWS Systems Manager com o AWS CloudTrail
- Registro de saída de ações do Automation em log com o CloudWatch Logs
- Configurar o Amazon CloudWatch Logs para Run Command
- Monitorar eventos do Systems Manager com o Amazon EventBridge
- Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS

Ferramentas de monitoramento

O conteúdo deste capítulo fornece informações sobre como usar as ferramentas disponíveis para monitorar o Systems Manager e outros recursos da AWS. Para obter uma lista mais completa de ferramentas, consulte Registrar em log e monitorar no AWS Systems Manager.

Enviar logs de nós para o CloudWatch Logs unificado (agente do CloudWatch)

Você pode configurar e usar o agente do Amazon CloudWatch para coletar métricas e logs de seus nós em vez de usar o AWS Systems Manager Agent (SSM Agent) para essas tarefas. O agente do CloudWatch permite reunir mais métricas sobre instâncias do EC2 do que as disponíveis usando o SSM Agent. Além disso, você pode coletar métricas de servidores on-premises usando o agente do CloudWatch.

Você pode também armazenar as definições de configuração no Systems Manager Parameter Store para uso com o agente do CloudWatch. O Parameter Store é uma ferramenta do AWS Systems Manager.

> **Note**
>
> O AWS Systems Manager é compatível com a migração do SSM Agent para o agente unificado do CloudWatch a fim de coletar logs e métricas somente em versões de 64 bits do Windows. Para obter informações sobre como configurar o agente unificado do CloudWatch

em outros sistemas operacionais e para obter informações completas sobre como usar o agente do CloudWatch, consulte Coletar métricas e logs das instâncias do Amazon EC2 e dos servidores on-premises com o agente do CloudWatch no Guia do usuário do Amazon CloudWatch.

Você pode usar o agente do CloudWatch em outros sistemas operacionais compatíveis, mas não poderá usar o Systems Manager para executar uma migração de ferramenta.

O SSM Agent grava informações sobre execuções, ações programadas, erros e status de integridade em arquivos de log para cada nó. A conexão manual a um nó para visualizar arquivos de log e solucionar problemas com o SSM Agent é um processo demorado. Para obter um monitoramento de nós mais eficiente, você pode configurar o SSM Agent em si ou o agente do CloudWatch para enviar esses dados de logs para o Amazon CloudWatch Logs.

Important

O agente unificado do CloudWatch substituiu SSM Agent como a ferramenta para enviar dados de log para o Amazon CloudWatch Logs. Não há compatibilidade com o plugin aws:cloudWatch do SSM Agent. Recomendamos usar somente o agente do CloudWatch unificado nos processos de coleta de logs. Para obter mais informações, consulte os tópicos a seguir.

- Enviar logs de nós para o CloudWatch Logs unificado (agente do CloudWatch)
- Migrar a coleta de logs de nós do Windows Server para o agente do CloudWatch
- Coletar métricas, logs e rastreamentos com o agente do CloudWatch no Guia do usuário do Amazon CloudWatch.

Usando o CloudWatch Logs, você pode monitorar dados de log em tempo real, pesquisar e filtrar dados de log por meio da criação de um ou mais filtros de métrica e arquivar e recuperar dados históricos quando necessário. Para obter mais informações sobre o CloudWatch Logs, consulte o Guia do usuário do Amazon CloudWatch Logs

Configurar um agente para enviar dados de log ao Amazon CloudWatch Logs oferece os seguintes benefícios:

- Armazenamento centralizado de arquivos de log para todos os arquivos de log do SSM Agent.
- Acesso mais rápido a arquivos para investigar erros.

- Retenção indefinida de arquivos de log (configurável).
- Os logs podem ser mantidos e acessados independentemente do status do nó.
- Acesso a outros recursos do CloudWatch, como métricas e alarmes.

Para obter informações sobre o monitoramento da atividade do Session Manager, consulte Registrar a atividade de sessão de registro em log e Habilitar e desabilitar o registro em log de sessão.

Migrar a coleta de logs de nós do Windows Server para o agente do CloudWatch

Se estiver usando o SSM Agent em nós do Windows Server compatíveis para enviar arquivos de log do SSM Agent para o Amazon CloudWatch Logs, é possível usar o Systems Manager para migrar do SSM Agent para o agente do CloudWatch como sua ferramenta de coleta de logs, bem como para migrar suas definições de configuração.

O agente do CloudWatch não é compatível com as versões de 32 bits do Windows Server.

Para instâncias do EC2 de 64 bits do Windows Server, você pode executar a migração para o agente do CloudWatch de forma automática ou manual. Para servidores on-premises e máquinas virtuais, o processo deve ser realizado manualmente.

> **Note**
>
> Durante o processo de migração, os dados enviados ao CloudWatch podem ser interrompidos ou duplicados. Suas métricas e dados de log serão registrados com precisão novamente no CloudWatch depois que a migração for concluída.

Recomendamos testar a migração em um pequeno número de nós antes de migrar uma frota inteira para o agente do CloudWatch. Após a migração, se preferir realizar a coleta de logs com o SSM Agent, poderá voltar a usá-lo.

> **Important**
>
> Nos casos a seguir, você não poderá migrar para o agente do CloudWatch usando as etapas descritas neste tópico:
>
> - A configuração existente para o SSM Agent especifica várias regiões.

- A configuração existente para o SSM Agent especifica vários conjuntos de credenciais de acesso/chave secreta.

Nesses casos, será necessário desativar a coleta de logs no SSM Agent e instalar o agente do CloudWatch sem um processo de migração. Para obter mais informações, consulte os seguintes tópicos no Manual do usuário do Amazon CloudWatch:

- Instalação do agente do CloudWatch
- Instação do agente do CloudWatch em servidores no on-premises

Antes de começar

Antes de iniciar uma migração para o agente do CloudWatch, a fim de fazer a coleta de logs, confira se os nós em que você vai executar a migração cumprem estes requisitos:

- O sistema operacional é uma versão de 64 bits do Windows Server.
- O SSM Agent 2.2.93.0 ou posterior está instalado em seu nó.
- O SSM Agent é configurado para monitoramento do nó.

Tópicos

- Migre automaticamente para o agente do CloudWatch
- Migrar manualmente para o agente do CloudWatch

Migre automaticamente para o agente do CloudWatch

Somente nas instâncias do EC2 para Windows Server, você pode usar o console do AWS Systems Manager ou a AWS Command Line Interface (AWS CLI) para migrar automaticamente para o agente do CloudWatch como sua ferramenta de coleta de logs.

> **Note**
>
> O AWS Systems Manager é compatível com a migração do SSM Agent para o agente unificado do CloudWatch a fim de coletar logs e métricas somente em versões de 64 bits do Windows. Para obter informações sobre como configurar o agente unificado do CloudWatch em outros sistemas operacionais e para obter informações completas sobre como usar o

agente do CloudWatch, consulte Coletar métricas e logs das instâncias do Amazon EC2 e dos servidores on-premises com o agente do CloudWatch no Guia do usuário do Amazon CloudWatch.
Você pode usar o agente do CloudWatch em outros sistemas operacionais compatíveis, mas não poderá usar o Systems Manager para executar uma migração de ferramenta.

Após a conclusão da migração, verifique os resultados no CloudWatch para garantir o recebimento das métricas, os logs ou os logs de eventos do Windows que você espera. Se estiver satisfeito com os resultados, poderá, opcionalmente, Armazenar as configurações do agente do CloudWatch no Parameter Store. Se a migração for malsucedida ou os resultados não forem os esperados, você poderá Reverter para a coleta de logs com o SSM Agent.

Note

Se você quiser migrar um arquivo de configuração de origem que inclui uma entrada `{hostname}`, lembre-se de que a entrada `{hostname}` pode alterar o valor do campo após a conclusão da migração. Por exemplo, digamos que a entrada `"LogStream": "{hostname}"` mapeie para um servidor denominado MyLogServer001.

```
{
"Id": "CloudWatchIISLogs",
"FullName":
 "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
"Parameters": {
"AccessKey": "",
"SecretKey": "",
"Region": "us-east-1",
"LogGroup": "Production-Windows-IIS",
"LogStream": "{hostname}"
     }
}
```

Após a migração, essa entrada mapeará para um domínio, como ip-11-1-1-11.production.ExampleCompany.com. Para manter o valor do nome do host local, especifique `{local_hostname}` em vez de `{hostname}`.

Para migrar automaticamente para o agente do CloudWatch (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Run Command e Run command (Executar comando).
3. Na lista Command document (Documento do comando), escolha `AmazonCloudWatch-MigrateCloudWatchAgent`.
4. Em Status, escolha Enabled.
5. Na seção Targets (Destinos), escolha os nós gerenciados nos quais você quer executar essa operação, especificando as tags, selecionando as instâncias ou dispositivos de borda manualmente ou especificando um grupo de recursos.

   > **Tip**
   >
   > Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

6. Para Rate control (Controle de taxa):

   - Em Concurrency (Concorrência), especifique um número ou uma porcentagem de nós gerenciados nos quais executar o comando ao mesmo tempo.

     > **Note**
     >
     > Se você selecionou destinos especificando tags aplicadas a instâncias a nós gerenciados ou especificando grupos de recursos da AWS, e não tiver certeza de quantas instâncias são direcionadas, restrinja o número de instâncias que poderão executar o documento ao mesmo tempo, especificando uma porcentagem.

   - Em Error threshold (Limite de erro), especifique quando parar de executar o comando em outros nós depois de falhar em alguns ou em uma porcentagem de nós. Por exemplo, se você especificar três erros, o Systems Manager deixará de enviar o comando quando o 4° erro for recebido. Os nós gerenciados que continuam processando o comando também podem enviar erros.

7. (Opcional) Em Output options (Opções de saída), para salvar a saída do comando em um arquivo, selecione a caixa Write command output to an S3 bucket (Gravar saída do comando em um bucket do S3). Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

> **Note**
>
> As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil de instância (para instâncias do EC2) ou perfil de serviço do IAM (máquinas ativadas para ambientes híbridos) atribuído à instância, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço do IAM associado ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

8. Na seção SNS notifications (Notificações do SNS), se quiser enviar notificações sobre o status da execução do comando, marque a caixa de seleção Enable SNS notifications (Habilitar notificações do SNS).

   Para obter mais informações sobre a configuração de notificações do Amazon SNS para o Run Command, consulte Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS.

9. Escolha Executar.

Para migrar automaticamente para o agente do CloudWatch (AWS CLI)

- Execute o seguinte comando:

```
aws ssm send-command --document-name AmazonCloudWatch-MigrateCloudWatchAgent --
targets Key=instanceids,Values=ID1,ID2,ID3
```

ID1, *ID2* e *ID3* representam os IDs dos nós que você deseja atualizar, como i-02573cafcfEXAMPLE.

Migrar manualmente para o agente do CloudWatch

Para nós on-premises do Windows Server ou instâncias do EC2 para o Windows Server, siga as etapas a seguir para migrar manualmente a coleta de logs para o agente do Amazon CloudWatch.

Note

Se você quiser migrar um arquivo de configuração de origem que inclui uma entrada `{hostname}`, lembre-se de que a entrada `{hostname}` pode alterar o valor do campo após a conclusão da migração. Por exemplo, digamos que a entrada `"LogStream": "{hostname}"` mapeie para um servidor denominado MyLogServer001.

```
{
"Id": "CloudWatchIISLogs",
"FullName":
 "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
"Parameters": {
"AccessKey": "",
"SecretKey": "",
"Region": "us-east-1",
"LogGroup": "Production-Windows-IIS",
"LogStream": "{hostname}"
    }
}
```

Após a migração, essa entrada mapeará para um domínio, como ip-11-1-1-11.production.ExampleCompany.com. Para manter o valor do nome do host local, especifique `{local_hostname}` em vez de `{hostname}`.

Primeira: para instalar o agente do CloudWatch (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Run Command e Run command (Executar comando).
3. Na lista Command document (Documento do comando), escolha `AWS-ConfigureAWSPackage`.
4. Em Action (Ação), escolha `Install`.
5. Em Nome, digite **`AmazonCloudWatchAgent`**.
6. Em Version (Versão), digite **`latest`**, caso ainda não tenha sido fornecida por padrão.
7. Na seção Targets (Destinos), escolha os nós gerenciados nos quais você quer executar essa operação, especificando as tags, selecionando as instâncias ou dispositivos de borda manualmente ou especificando um grupo de recursos.

> **Tip**
>
> Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

8. Para Rate control (Controle de taxa):

   - Em Concurrency (Concorrência), especifique um número ou uma porcentagem de nós gerenciados nos quais executar o comando ao mesmo tempo.

     > **Note**
     >
     > Se você selecionou destinos especificando tags aplicadas a instâncias a nós gerenciados ou especificando grupos de recursos da AWS, e não tiver certeza de quantas instâncias são direcionadas, restrinja o número de instâncias que poderão executar o documento ao mesmo tempo, especificando uma porcentagem.

   - Em Error threshold (Limite de erro), especifique quando parar de executar o comando em outros nós depois de falhar em alguns ou em uma porcentagem de nós. Por exemplo, se você especificar três erros, o Systems Manager deixará de enviar o comando quando o 4° erro for recebido. Os nós gerenciados que continuam processando o comando também podem enviar erros.

9. (Opcional) Em Output options (Opções de saída), para salvar a saída do comando em um arquivo, selecione a caixa Write command output to an S3 bucket (Gravar saída do comando em um bucket do S3). Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

> **Note**
>
> As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil de instância (para instâncias do EC2) ou perfil de serviço do IAM (máquinas ativadas para ambientes híbridos) atribuído à instância, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a

função de serviço do IAM associado ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

10. Na seção SNS notifications (Notificações do SNS), se quiser enviar notificações sobre o status da execução do comando, marque a caixa de seleção Enable SNS notifications (Habilitar notificações do SNS).

    Para obter mais informações sobre a configuração de notificações do Amazon SNS para o Run Command, consulte Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS.

11. Escolha Executar.

Segunda: Para atualizar o formato JSON de dados de configuração

- Para atualizar a formatação JSON das definições de configuração existentes para o agente do CloudWatch, use o Run Command, uma ferramenta do AWS Systems Manager, ou faça login diretamente em seu nó com uma conexão RDP para executar os comandos do Windows PowerShell a seguir nesse nó, um de cada vez.

```
cd ${Env:ProgramFiles}\\Amazon\\AmazonCloudWatchAgent
```

```
.\\amazon-cloudwatch-agent-config-wizard.exe --isNonInteractiveWindowsMigration
```

  {Env:ProgramFiles} representa o local em que o diretório da Amazon pode ser encontrado, e que contém o agente do CloudWatch, normalmente `C:\Program Files`.

Terceira: para configurar e iniciar o agente do CloudWatch (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Run Command e Run command (Executar comando).
3. Na lista Command document (Documento do comando), escolha `AWS-RunPowerShellScript`.
4. Em Commands (Comandos), insira os dois comandos a seguir.

```
cd ${Env:ProgramFiles}\Amazon\AmazonCloudWatchAgent
```

```
.\amazon-cloudwatch-agent-ctl.ps1 -a fetch-config -m ec2 -c file:config.json -s
```

{Env:ProgramFiles} representa o local em que o diretório da Amazon pode ser encontrado, e que contém o agente do CloudWatch, normalmente `C:\Program Files`.

5. Na seção Targets (Destinos), escolha os nós gerenciados nos quais você quer executar essa operação, especificando as tags, selecionando as instâncias ou dispositivos de borda manualmente ou especificando um grupo de recursos.

> **Tip**
>
> Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

6. Para Rate control (Controle de taxa):

   - Em Concurrency (Concorrência), especifique um número ou uma porcentagem de nós gerenciados nos quais executar o comando ao mesmo tempo.

> **Note**
>
> Se você selecionou destinos especificando tags aplicadas a instâncias a nós gerenciados ou especificando grupos de recursos da AWS, e não tiver certeza de quantas instâncias são direcionadas, restrinja o número de instâncias que poderão executar o documento ao mesmo tempo, especificando uma porcentagem.

   - Em Error threshold (Limite de erro), especifique quando parar de executar o comando em outros nós depois de falhar em alguns ou em uma porcentagem de nós. Por exemplo, se você especificar três erros, o Systems Manager deixará de enviar o comando quando o 4° erro for recebido. Os nós gerenciados que continuam processando o comando também podem enviar erros.

7. (Opcional) Em Output options (Opções de saída), para salvar a saída do comando em um arquivo, selecione a caixa Write command output to an S3 bucket (Gravar saída do comando em um bucket do S3). Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

> **Note**
>
> As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil de instância (para instâncias do EC2) ou perfil de serviço do IAM (máquinas ativadas para ambientes híbridos) atribuído à instância, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço do IAM associado ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

8. Na seção SNS notifications (Notificações do SNS), se quiser enviar notificações sobre o status da execução do comando, marque a caixa de seleção Enable SNS notifications (Habilitar notificações do SNS).

   Para obter mais informações sobre a configuração de notificações do Amazon SNS para o Run Command, consulte Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS.

9. Escolha Executar.

Quatro: para desabilitar a coleta de logs no SSM Agent (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Run Command e Run command (Executar comando).
3. Na lista Command document (Documento do comando), escolha `AWS-ConfigureCloudWatch`.
4. Em Status, escolha Disabled (Desabilitado).
5. Na seção Targets (Destinos), escolha os nós gerenciados nos quais você quer executar essa operação, especificando as tags, selecionando as instâncias ou dispositivos de borda manualmente ou especificando um grupo de recursos.

> **Tip**
>
> Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

6. Em Status, escolha `Disabled`.
7. Para Rate control (Controle de taxa):
   - Em Concurrency (Concorrência), especifique um número ou uma porcentagem de nós gerenciados nos quais executar o comando ao mesmo tempo.

     > **Note**
     >
     > Se você selecionou destinos especificando tags aplicadas a instâncias a nós gerenciados ou especificando grupos de recursos da AWS, e não tiver certeza de quantas instâncias são direcionadas, restrinja o número de instâncias que poderão executar o documento ao mesmo tempo, especificando uma porcentagem.

   - Em Error threshold (Limite de erro), especifique quando parar de executar o comando em outros nós depois de falhar em alguns ou em uma porcentagem de nós. Por exemplo, se você especificar três erros, o Systems Manager deixará de enviar o comando quando o 4° erro for recebido. Os nós gerenciados que continuam processando o comando também podem enviar erros.
8. (Opcional) Em Output options (Opções de saída), para salvar a saída do comando em um arquivo, selecione a caixa Write command output to an S3 bucket (Gravar saída do comando em um bucket do S3). Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

> **Note**
>
> As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil de instância (para instâncias do EC2) ou perfil de serviço do IAM (máquinas ativadas para ambientes híbridos) atribuído à instância, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a

função de serviço do IAM associado ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

9. Na seção SNS notifications (Notificações do SNS), se quiser enviar notificações sobre o status da execução do comando, marque a caixa de seleção Enable SNS notifications (Habilitar notificações do SNS).

   Para obter mais informações sobre a configuração de notificações do Amazon SNS para o Run Command, consulte Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS.

10. Escolha Executar.

    Após a conclusão dessas etapas, verifique seus logs no CloudWatch para garantir que está recebendo as métricas, os logs, ou os logs de eventos do Windows que você espera. Se estiver satisfeito com os resultados, você poderá, opcionalmente, Armazenar as configurações do agente do CloudWatch no Parameter Store. Se a migração for malsucedida ou os resultados não forem os esperados, você poderá Reverter para a coleta de logs com o SSM Agent.

Armazenar as configurações do agente do CloudWatch no Parameter Store

Você pode armazenar o conteúdo de um arquivo de configuração do agente do CloudWatch da Amazon no Parameter Store. Ao manter esses dados de configuração em um parâmetro, vários nós poderão extrair suas definições de configuração desse parâmetro e você não precisará criar ou atualizar manualmente os arquivos de configuração em seus nós. Por exemplo, você pode usar o Run Command para gravar o conteúdo do parâmetro em arquivos de configuração em vários nós ou usar o State Manager, uma ferramenta do AWS Systems Manager, para ajudar a evitar oscilações de configuração no agente do CloudWatch em toda a frota de nós.

Ao executar o assistente de configuração do agente do CloudWatch, poderá optar por permitir que o assistente salve suas definições de configuração como um novo parâmetro no Parameter Store. Para obter informações sobre como executar o assistente de configuração do agente do CloudWatch, consulte Create the CloudWatch agent configuration file with the wizard (Criar o arquivo de configuração do agente do CloudWatch com o assistente) no Guia do usuário do Amazon CloudWatch.

Se você tiver executado o assistente, mas não tiver escolhido a opção para salvar as configurações como parâmetro, ou se tiver criado manualmente o arquivo de configuração do agente do

CloudWatch, poderá recuperar os dados para salvar como parâmetro em seu nó, neste arquivo a seguir.

```
${Env:ProgramFiles}\Amazon\AmazonCloudWatchAgent\config.json
```

{Env:ProgramFiles} representa o local em que o diretório da Amazon pode ser encontrado, e que contém o agente do CloudWatch, normalmente `C:\Program Files`.

Recomendamos manter um backup do JSON nesse arquivo em um local diferente do nó em si.

Para obter mais informações sobre como criar um parâmetro, consulte Criar parâmetros do Parameter Store no Systems Manager.

Para obter mais informações sobre o agente do CloudWatch, consulte Coletar métricas e logs de instâncias do Amazon EC2 e de servidores on-premises com o agente do CloudWatch no Guia do usuário do Amazon CloudWatch.

Reverter para a coleta de logs com o SSM Agent

Se quiser voltar a usar o SSM Agent para coletar logs, execute as etapas a seguir.

Primeira: Para recuperar dados de configuração do SSM Agent

1. Nesse nó no qual que você deseja voltar a coletar logs com o SSM Agent, localize o conteúdo do arquivo de configuração do SSM Agent. Esse arquivo JSON geralmente é encontrado no seguinte local:

   ```
   ${Env:ProgramFiles}\\Amazon\\SSM\\Plugins\\awsCloudWatch\
   \AWS.EC2.Windows.CloudWatch.json
   ```

   {Env:ProgramFiles} representa o local em que o diretório `Amazon` pode ser encontrado, normalmente em `C:\Program Files`.
2. Copie esses dados em um arquivo de texto para uso em uma etapa posterior.

   Recomendamos armazenar um backup do JSON em um local diferente do nó em si.

Dois: Para desinstalar o agente do CloudWatch (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.

2. No painel de navegação, escolha Run Command e Run command (Executar comando).
3. Na lista Command document (Documento do comando), escolha `AWS-ConfigureAWSPackage`.
4. Em Action (Ação), escolha Uninstall (Desinstalar).
5. Em Nome, digite **`AmazonCloudWatchAgent`**.
6. Na seção Targets (Destinos), escolha os nós gerenciados nos quais você quer executar essa operação, especificando as tags, selecionando as instâncias ou dispositivos de borda manualmente ou especificando um grupo de recursos.

   > **Tip**
   >
   > Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

7. Para Rate control (Controle de taxa):

   - Em Concurrency (Concorrência), especifique um número ou uma porcentagem de nós gerenciados nos quais executar o comando ao mesmo tempo.

     > **Note**
     >
     > Se você selecionou destinos especificando tags aplicadas a instâncias a nós gerenciados ou especificando grupos de recursos da AWS, e não tiver certeza de quantas instâncias são direcionadas, restrinja o número de instâncias que poderão executar o documento ao mesmo tempo, especificando uma porcentagem.

   - Em Error threshold (Limite de erro), especifique quando parar de executar o comando em outros nós depois de falhar em alguns ou em uma porcentagem de nós. Por exemplo, se você especificar três erros, o Systems Manager deixará de enviar o comando quando o 4° erro for recebido. Os nós gerenciados que continuam processando o comando também podem enviar erros.

8. (Opcional) Em Output options (Opções de saída), para salvar a saída do comando em um arquivo, selecione a caixa Write command output to an S3 bucket (Gravar saída do comando em um bucket do S3). Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

> **Note**
>
> As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil de instância (para instâncias do EC2) ou perfil de serviço do IAM (máquinas ativadas para ambientes híbridos) atribuído à instância, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço do IAM associado ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

9. Na seção SNS notifications (Notificações do SNS), se quiser enviar notificações sobre o status da execução do comando, marque a caixa de seleção Enable SNS notifications (Habilitar notificações do SNS).

   Para obter mais informações sobre a configuração de notificações do Amazon SNS para o Run Command, consulte Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS.

10. Escolha Executar.

Três: Para ativar novamente a coleção de logs no SSM Agent (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Run Command e Run command (Executar comando).
3. Na lista Command document (Documento do comando), escolha `AWS-ConfigureCloudWatch`.
4. Em Status, escolha `Enabled`.
5. Em Properties (Propriedades), cole o conteúdo dos dados da configuração antiga que você salvou no arquivo de texto.
6. Na seção Targets (Destinos), escolha os nós gerenciados nos quais você quer executar essa operação, especificando as tags, selecionando as instâncias ou dispositivos de borda manualmente ou especificando um grupo de recursos.

> **Tip**
>
> Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

7. Para Rate control (Controle de taxa):

   - Em Concurrency (Concorrência), especifique um número ou uma porcentagem de nós gerenciados nos quais executar o comando ao mesmo tempo.

     > **Note**
     >
     > Se você selecionou destinos especificando tags aplicadas a instâncias a nós gerenciados ou especificando grupos de recursos da AWS, e não tiver certeza de quantas instâncias são direcionadas, restrinja o número de instâncias que poderão executar o documento ao mesmo tempo, especificando uma porcentagem.

   - Em Error threshold (Limite de erro), especifique quando parar de executar o comando em outros nós depois de falhar em alguns ou em uma porcentagem de nós. Por exemplo, se você especificar três erros, o Systems Manager deixará de enviar o comando quando o 4° erro for recebido. Os nós gerenciados que continuam processando o comando também podem enviar erros.

8. (Opcional) Em Output options (Opções de saída), para salvar a saída do comando em um arquivo, selecione a caixa Write command output to an S3 bucket (Gravar saída do comando em um bucket do S3). Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

   > **Note**
   >
   > As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil de instância (para instâncias do EC2) ou perfil de serviço do IAM (máquinas ativadas para ambientes híbridos) atribuído à instância, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a

função de serviço do IAM associado ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

9. Na seção SNS notifications (Notificações do SNS), se quiser enviar notificações sobre o status da execução do comando, marque a caixa de seleção Enable SNS notifications (Habilitar notificações do SNS).

   Para obter mais informações sobre a configuração de notificações do Amazon SNS para o Run Command, consulte Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS.

10. Escolha Executar.

Enviar logs do SSM Agent ao CloudWatch Logs

O AWS Systems Manager Agent (SSM Agent) é um software da Amazon executado em suas instâncias, dispositivos de borda, servidores on-premises e máquinas virtuais (VMs) do EC2 configurados para o Systems Manager. O SSM Agent processa solicitações de serviços do Systems Manager na nuvem e configura sua máquina conforme especificado na solicitação. Para obter mais informações sobre o SSM Agent, consulte Trabalhar com o SSM Agent.

Além disso, seguindo as etapas abaixo, você pode configurar o SSM Agent para enviar dados de log para o Amazon CloudWatch Logs.

Antes de começar

Crie um grupo de logs no CloudWatch Logs. Para obter mais informações, consulte Conceitos básicos do CloudWatch Logs no Guia do usuário do Amazon CloudWatch Logs.

Para configurar o SSM Agent para enviar logs ao CloudWatch

1. Faça login em um nó e localize o seguinte arquivo:

   Linux

   Na maioria dos tipos de nó do Linux: `/etc/amazon/ssm/seelog.xml.template`.

   No Ubuntu Server 20,10 STR e 20.04, 18.04 e 16.04 LTS:`/snap/amazon-ssm-agent/current/seelog.xml.template`

   macOS

`/opt/aws/ssm/seelog.xml.template`

Windows

`%ProgramFiles%\Amazon\SSM\seelog.xml.template`

2. Altere o nome do arquivo de `seelog.xml.template` para `seelog.xml`.

> **Note**
>
> No Ubuntu Server 20.10 STR & 20.04, 18.04 e 16.04 LTS, o arquivo `seelog.xml` deve ser criado no diretório `/etc/amazon/ssm/`. Você pode criar esse diretório e arquivo executando os comandos a seguir.
>
> ```
> sudo mkdir -p /etc/amazon/ssm
> ```
>
> ```
> sudo cp -pr /snap/amazon-ssm-agent/current/* /etc/amazon/ssm
> ```
>
> ```
> sudo cp -p /etc/amazon/ssm/seelog.xml.template /etc/amazon/ssm/seelog.xml
> ```

3. Abra o arquivo `seelog.xml` com um editor de texto e localize a seguinte seção.

Linux and macOS

```
<outputs formatid="fmtinfo">
   <console formatid="fmtinfo"/>
   <rollingfile type="size" filename="/var/log/amazon/ssm/amazon-ssm-agent.log"
 maxsize="30000000" maxrolls="5"/>
   <filter levels="error,critical" formatid="fmterror">
      <rollingfile type="size" filename="/var/log/amazon/ssm/errors.log"
 maxsize="10000000" maxrolls="5"/>
   </filter>
</outputs>
```

Windows

```
<outputs formatid="fmtinfo">
   <console formatid="fmtinfo"/>
```

```
   <rollingfile type="size" maxrolls="5" maxsize="30000000"
 filename="{{LOCALAPPDATA}}\Amazon\SSM\Logs\amazon-ssm-agent.log"/>
   <filter formatid="fmterror" levels="error,critical">
      <rollingfile type="size" maxrolls="5" maxsize="10000000"
 filename="{{LOCALAPPDATA}}\Amazon\SSM\Logs\errors.log"/>
   </filter>
</outputs>
```

4. Edite o arquivo e adicione um elemento de nome personalizado após a tag de fechamento </filter>. No exemplo a seguir, o nome personalizado foi especificado como `cloudwatch_receiver`.

   Linux and macOS

```
<outputs formatid="fmtinfo">
   <console formatid="fmtinfo"/>
   <rollingfile type="size" filename="/var/log/amazon/ssm/amazon-ssm-agent.log"
 maxsize="30000000" maxrolls="5"/>
   <filter levels="error,critical" formatid="fmterror">
      <rollingfile type="size" filename="/var/log/amazon/ssm/errors.log"
 maxsize="10000000" maxrolls="5"/>
   </filter>
   <custom name="cloudwatch_receiver" formatid="fmtdebug" data-log-group="your-
CloudWatch-log-group-name"/>
</outputs>
```

   Windows

```
<outputs formatid="fmtinfo">
   <console formatid="fmtinfo"/>
   <rollingfile type="size" maxrolls="5" maxsize="30000000"
 filename="{{LOCALAPPDATA}}\Amazon\SSM\Logs\amazon-ssm-agent.log"/>
   <filter formatid="fmterror" levels="error,critical">
      <rollingfile type="size" maxrolls="5" maxsize="10000000"
 filename="{{LOCALAPPDATA}}\Amazon\SSM\Logs\errors.log"/>
   </filter>
   <custom name="cloudwatch_receiver" formatid="fmtdebug" data-log-group="your-
CloudWatch-log-group-name"/>
</outputs>
```

5. Salve as alterações e, em seguida, reinicie o SSM Agent ou o nó.
6. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.

7. No painel de navegação, escolha Log groups (Grupos de logs) e escolha o nome do grupo de logs.

> **Tip**
>
> O fluxo de log para dados do arquivo de log do SSM Agent são organizados por ID do nó.

Monitoramento dos seus eventos de solicitação de alteração

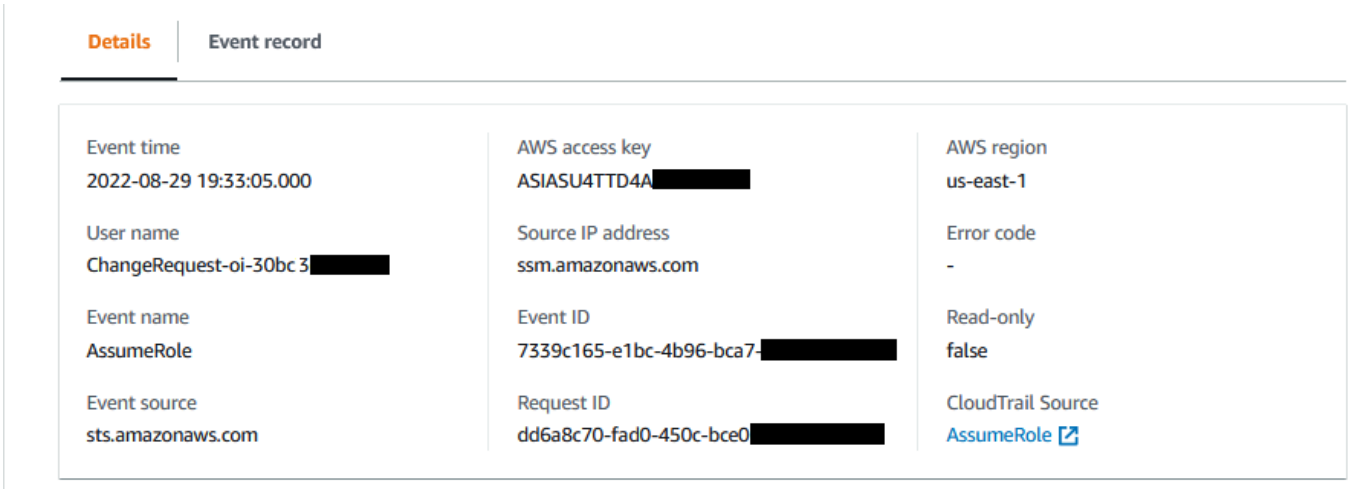
Após ativar a integração com o AWS CloudTrail Lake e criar um armazenamento de dados de eventos, você poderá visualizar detalhes auditáveis sobre as solicitações de alteração executadas em sua conta ou organização. Isso inclui detalhes como os seguintes:

- A identidade do usuário que iniciou a solicitação de alteração
- As Regiões da AWS onde as alterações foram feitas
- O endereço IP de origem da solicitação
- A chave de acesso da AWS usada para a solicitação
- As ações da API executadas para a solicitação de alteração
- Os parâmetros da solicitação incluídos para essas ações
- Os recursos atualizados durante o processo

A seguir há exemplos de detalhes do evento que você pode visualizar para uma solicitação de alteração depois de criar o armazenamento de dados de eventos no AWS CloudTrail Lake.

Details

A imagem a seguir mostra as informações de alto nível sobre uma solicitação de alteração disponíveis na guia Details (Detalhes). Esses detalhes incluem informações como a hora em que a operação da solicitação de alteração começou, o ID do usuário que iniciou a solicitação de alteração, o Região da AWS afetado e o ID do evento e o ID da solicitação associados à solicitação.



Event record

A imagem a seguir mostra a estrutura do conteúdo JSON fornecido pelo CloudTrail Lake para um evento de solicitação de alteração. Esses dados são fornecidos na guia Event record (Registro de eventos) em uma solicitação de alteração.



> **Important**
>
> Se você estiver usando o Change Manager para uma organização, será possível concluir o procedimento a seguir enquanto estiver conectado à conta de gerenciamento ou à conta de administrador delegado do Change Manager.
> No entanto, para usar a conta de administrador delegado para concluir essas etapas, a mesma conta de administrador delegado deve ser especificada tanto para o CloudTrail quanto para o Change Manager.
> Ao fazer login na conta de gerenciamento do Change Manager, será possível adicionar ou alterar a conta de administrador delegado do CloudTrail na página Settings (Configurações) do CloudTrail. Isso deve ser feito antes que a conta de administrador delegado possa criar um armazenamento de dados de eventos para uso por toda a organização.

Para ativar o rastreamento de eventos do CloudTrail Lake do Change Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Change Manager.
3. Selecione a guia Requests (Solicitações).
4. Escolha qualquer solicitação de alteração existente e, em seguida, escolha a guia Associated events (Eventos associados).
5. Escolha Enable CloudTrail Lake (Habilitar o CloudTrail Lake).
6. Siga as etapas descritas em Create an event data store for CloudTrail events no Guia do usuário do AWS CloudTrail.

   Para garantir que os dados de eventos de suas solicitações de alteração sejam armazenados, faça as seguintes seleções ao concluir o procedimento:

   - Em Tipo de evento, mantenha os padrões para Eventos da AWS e para Eventos do CloudTrail selecionados.
   - Se você estiver usando o Change Manager com uma organização, selecione Habilitar para todas as contas em minha organização.
   - Em Eventos de gerenciamento, não desmarque a caixa de seleção Gravar.

   Outras opções que você escolhe ao criar seu armazenamento de dados de eventos não afetam o armazenamento de dados de eventos para suas solicitações de alteração.

Monitoramento das automações

As métricas são um conceito fundamental do Amazon CloudWatch. Uma métrica representa um conjunto de pontos de dados ordenados ao longo do tempo que são publicados no CloudWatch. Considere uma métrica como variável a ser monitorada, e os pontos de dados representando os valores dessa variável ao longo do tempo.

O Automation é uma ferramenta do AWS Systems Manager. O Systems Manager publica métricas sobre o uso do Automation no CloudWatch. Isso permite definir alarmes com base nessas métricas.

Para exibir métricas do Automation no console do CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Métricas.
3. Escolha SSM.
4. Na guia Metrics (Métricas), escolha Usage (Uso) e By Resource AWS(Por recurso da).
5. Na caixa de pesquisa perto da lista de métricas, insira SSM.

Para exibir as métricas do Automation usando a AWS CLI

Abra um prompt de comando e use o comando a seguir.

```
aws cloudwatch list-metrics \
    --namespace "AWS/Usage"
```

Métricas do Automation

O Systems Manager envia as métricas do Automation a seguir ao CloudWatch.

Métrica	Descrição
`ConcurrentAutomationUsage`	O número de automações em execução ao mesmo tempo nas Conta da AWS e Região da AWS atuais.
`QueuedAutomationUsage`	O número de automações atualmente enfileiradas que não foram iniciadas e têm status de `Pending`.

Para obter mais informações sobre como trabalhar com as métricas do CloudWatch, consulte os seguintes tópicos no Manual do usuário do Amazon CloudWatch:

- Métricas
- Usar métricas do Amazon CloudWatch
- Usar alarmes do Amazon CloudWatch

Monitorar métricas do Run Command com o Amazon CloudWatch

As métricas são um conceito fundamental do Amazon CloudWatch. Uma métrica representa um conjunto de pontos de dados ordenados ao longo do tempo que são publicados no CloudWatch. Considere uma métrica como variável a ser monitorada, e os pontos de dados representando os valores dessa variável ao longo do tempo.

O AWS Systems Manager agora publica métricas sobre o status de comandos do Run Command no CloudWatch, permitindo definir alarmes com base nessas métricas. O Run Command é uma ferramenta do AWS Systems Manager. Essas estatísticas são registradas por um período prolongado para que seja possível acessar informações históricas e obter uma perspectiva melhor sobre a taxa de êxito dos comandos executados em sua Conta da AWS.

Os valores de status do terminal de comandos para os quais é possível rastrear métricas incluem `Success`, `Failed` e `Delivery Timed Out`. Por exemplo, para um documento de comando do SSM definido para ser executado a cada hora, será possível configurar um alarme para notificá-lo quando um status `Success` não for relatado para qualquer uma dessas horas. Para obter mais informações sobre valores de status de comando, consulte Noções básicas sobre status de comando.

Para exibir métricas no console do CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Métricas.
3. Na área Alarms service (Alarmes por serviço da AWS), em Services (Serviços), escolha SSM-Run Command.

Para visualizar métricas usando o AWS CLI

Abra um prompt de comando e use o comando a seguir.

```
aws cloudwatch list-metrics --namespace "AWS/SSM-RunCommand"
```

Para listar todas as métricas disponíveis, use o comando a seguir.

```
aws cloudwatch list-metrics
```

Métricas e dimensões de Run Command do Systems Manager

O Systems Manager envia métricas de comando do Run Command para o CloudWatch uma vez a cada minuto.

O Systems Manager envia as métricas de comandos a seguir ao CloudWatch.

> **Note**
>
> Essas métricas usam `Count` como a unidade, portanto, `Sum` e `SampleCount` são as estatísticas mais úteis.

Métrica	Descrição
`CommandsDeliveryTimedOut`	O número de comandos que têm um status do terminal `Delivery Timed Out`.
`CommandsFailed`	O número de comandos que têm um status do terminal `Failed`.
`CommandsSucceeded`	O número de comandos que têm um status do terminal `Success`.

Para obter mais informações sobre como trabalhar com as métricas do CloudWatch, consulte os seguintes tópicos no Manual do usuário do Amazon CloudWatch:

- Métricas
- Usar métricas do Amazon CloudWatch
- Usar alarmes do Amazon CloudWatch

Registrar em log chamadas de API do AWS Systems Manager com o AWS CloudTrail

O AWS Systems Manager é integrado ao AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, um perfil ou um AWS service (Serviço da AWS). O CloudTrail captura todas as chamadas API para Systems Manager como eventos. As chamadas capturadas incluem as aquelas do Systems Manager console e chamadas de código para operações API da Systems Manager. Ao fazer uso das informações coletadas pelo CloudTrail, é possível determinar a solicitação feita ao Systems Manager, o endereço IP no qual a solicitação foi feita, quando a solicitação foi feita e detalhes adicionais.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar o seguinte:

- Se a solicitação foi feita com credenciais de usuário raiz ou credenciais de usuário.
- Se a solicitação foi feita em nome de um usuário do Centro de Identidade do IAM.
- Se a solicitação foi feita com credenciais de segurança temporárias de um perfil ou de um usuário federado.
- Se a solicitação foi feita por outro AWS service (Serviço da AWS).

Quando a Conta da AWS é criada, o CloudTrail torna-se ativo nela e você tem acesso automático ao Histórico de eventos do CloudTrail. O Histórico de eventos do CloudTrail fornece um registro visualizável, pesquisável, baixável e imutável dos últimos 90 dias de eventos de gerenciamento gravados em uma Região da AWS. Para obter mais informações, consulte Trabalhar com histórico de eventos do CloudTrail no Guia do usuário do AWS CloudTrail. Não há cobranças do CloudTrail pela visualização do Histórico de eventos.

Para obter um registro contínuo de eventos em sua Conta da AWS nos últimos 90 dias, crie uma trilha ou um armazenamento de dados de eventos do CloudTrail Lake.

Trilhas do CloudTrail

Uma trilha permite que o CloudTrail entregue arquivos de log a um bucket do Amazon S3. As trilhas criadas usando o AWS Management Console são de várias regiões. Só é possível criar uma trilha de região única ou de várias regiões usando a AWS CLI. Criar uma trilha de várias regiões é uma prática recomendada, pois você captura atividades em todas as Regiões da AWS da conta. Ao criar uma trilha de região única, é possível visualizar somente os eventos

registrados na Região da AWS da trilha. Para obter mais informações sobre trilhas, consulte Criar uma trilha para a Conta da AWS e Criar uma trilha para uma organização no Guia do usuário do AWS CloudTrail.

Uma cópia dos eventos de gerenciamento em andamento pode ser entregue no bucket do Amazon S3 sem nenhum custo via CloudTrail com a criação de uma trilha; no entanto, há cobranças de armazenamento do Amazon S3. Para obter mais informações sobre os preços do CloudTrail, consulte Preços do AWS CloudTrail. Para receber informações sobre a definição de preços do Amazon S3, consulte Definição de preços do Amazon S3.

Armazenamentos de dados de eventos do CloudTrail Lake

O CloudTrail Lake permite executar consultas baseadas em SQL nos eventos. O CloudTrail Lake converte eventos existentes em formato JSON baseado em linhas para o formato Apache ORC. O ORC é um formato colunar de armazenamento otimizado para recuperação rápida de dados. Os eventos são agregados em armazenamentos de dados de eventos, que são coleções imutáveis de eventos baseados nos critérios selecionados com a aplicação de seletores de eventos avançados. Os seletores que aplicados a um armazenamento de dados de eventos controlam quais eventos persistem e estão disponíveis para consulta. Para obter mais informações sobre o CloudTrail Lake, consulte Trabalhar com o AWS CloudTrail Lake, no Guia do usuário do AWS CloudTrail.

Os armazenamentos de dados de eventos e consultas do CloudTrail Lake incorrem em custos. Ao criar um armazenamento de dados de eventos, você escolhe a opção de preço que deseja usar para ele. A opção de preço determina o custo para a ingestão e para o armazenamento de eventos, e o período de retenção padrão e máximo para o armazenamento de dados de eventos. Para obter mais informações sobre os preços do CloudTrail, consulte Preços do AWS CloudTrail.

Eventos de dados do Systems Manager no CloudTrail

Os eventos de dados fornecem informações sobre as operações de recursos realizadas em um recurso (por exemplo, criar ou abrir um canal de controle). Também são conhecidas como operações de plano de dados. Eventos de dados geralmente são atividades de alto volume. Por padrão, o CloudTrail não registra eventos de dados em log. O Histórico de eventos do CloudTrail não registra eventos de dados.

Há cobranças adicionais para eventos de dados. Para obter mais informações sobre os preços do CloudTrail, consulte Preços do AWS CloudTrail.

É possível registrar em log eventos de dados para os tipos de recurso do Systems Manager usando o console do CloudTrail, a AWS CLI ou operações da API do CloudTrail. Para obter mais informações sobre como registrar eventos de dados em log, consulte Registrar eventos de dados com o AWS Management Console e Registrar eventos de dados com a AWS Command Line Interface no Guia do usuário do AWS CloudTrail.

A tabela a seguir lista o tipo de recurso do Systems Manager para o qual é possível registrar eventos de dados em log. A coluna Tipo de evento de dados (console) mostra o valor a ser escolhido na lista Tipo de evento de dados no console do CloudTrail. A coluna do valor resources.type mostra o valor de `resources.type` que você especificaria ao configurar seletores de eventos avançados usando a AWS CLI ou as APIs do CloudTrail. A coluna APIs de dados registradas no CloudTrail mostra as chamadas de API registradas no CloudTrail para o tipo de recurso.

Tipo de evento de dados (console)	valor resources.type	APIs de dados registradas no CloudTrail
Systems Manager (Gerenciador de sistemas)	`AWS::SSMMessages::ControlChannel`	• `CreateControlChannel` • `OpenControlChannel` Para obter mais informações, consulte Ações definidas pelo Amazon Message Gateway Service na Referência de autorização do serviço.
Nó gerenciado pelo Systems Manager	`AWS::SSM::ManagedNode`	• `RequestManagedInstanceRoleToken` : este evento é gerado quando o AWS Systems Manager Agent (SSM Agent) executado em um nó gerenciado pelo Systems Manager solicita credenciais do serviço de credenciais do Systems Manager.

Tipo de evento de dados (console)	valor resources.type	APIs de dados registradas no CloudTrail
		Consulte mais informações sobre a operação `RequestManagedInstanceRoleToken` em Validar máquinas ativadas para ambiente híbrido usando uma impressão digital do hardware

É possível configurar seletores de eventos avançados para filtrar os campos `eventName`, `readOnly` e `resources.ARN` para registrar em log somente os eventos que são importantes para você. Para obter mais informações sobre esses campos, consulte AdvancedFieldSelector na Referência de API do AWS CloudTrail.

Eventos de gerenciamento do Systems Manager no CloudTrail

Os eventos de gerenciamento fornecem informações sobre operações de gerenciamento executadas em recursos na Conta da AWS. Elas também são conhecidas como operações de ambiente de gerenciamento. Por padrão, o CloudTrail registra eventos de gerenciamento em logs.

O Amazon CloudFront registra em log todas as operações do ambiente de gerenciamento no CloudTrail como eventos de gerenciamento. As operações de API do Systems Manager estão documentadas na Referência da API do AWS Systems Manager. Por exemplo, as chamadas para as ações `CreateMaintenanceWindows`, `PutInventory`, `SendCommand` e `StartSession` gerarão entradas nos arquivos de log do CloudTrail. Para obter um exemplo de como configurar o CloudTrail para monitorar uma chamada de API do Systems Manager, consulte Monitorar as atividades da sessão usando o Amazon EventBridge (console).

Exemplos de eventos do Systems Manager

Um evento representa uma única solicitação de qualquer origem e inclui informações sobre a operação solicitada, a data e a hora da operação da API, os parâmetros de solicitação etc. Os arquivos de log do CloudTrail não são um rastreamento de pilha ordenada de chamadas de API públicas, portanto não são exibidos em uma ordem específica.

Exemplos:

- Exemplos de eventos de gerenciamento
- Exemplos de dados de eventos

Exemplos de eventos de gerenciamento

Exemplo 1: **`DeleteDocument`**

O exemplo a seguir mostra um evento do CloudTrail que demonstra a operação `DeleteDocument` em um documento chamado `example-Document` na região Leste dos EUA (Ohio) (us-east-2).

```
{
    "eventVersion": "1.04",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAI44QH8DHBEXAMPLE:203.0.113.11",
        "arn": "arn:aws:sts::123456789012:assumed-role/example-role/203.0.113.11",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-03-06T20:19:16Z"
            },
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAI44QH8DHBEXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/example-role",
                "accountId": "123456789012",
                "userName": "example-role"
            }
        }
    },
    "eventTime": "2018-03-06T20:30:12Z",
    "eventSource": "ssm.amazonaws.com",
    "eventName": "DeleteDocument",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "203.0.113.11",
    "userAgent": "example-user-agent-string",
    "requestParameters": {
        "name": "example-Document"
    },
```

```
    "responseElements": null,
    "requestID": "86168559-75e9-11e4-8cf8-75d18EXAMPLE",
    "eventID": "832b82d5-d474-44e8-a51d-093ccEXAMPLE",
    "resources": [
        {
            "ARN": "arn:aws:ssm:us-east-2:123456789012:document/example-Document",
            "accountId": "123456789012"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```

Exemplo 2: **`StartConnection`**

O exemplo a seguir mostra um evento do CloudTrail para um usuário que inicia uma conexão RDP usando o Fleet Manager na região Leste dos EUA (Ohio) (us-east-2). A ação da API subjacente é `StartConnection`.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:sts::123456789012:assumed-role/exampleRole",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAI44QH8DHBEXAMPLE",
                "arn": "arn:aws:sts::123456789012:assumed-role/exampleRole",
                "accountId": "123456789012",
                "userName": "exampleRole"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2021-12-13T14:57:05Z",
                "mfaAuthenticated": "false"
            }
        }
    },
```

```
    "eventTime": "2021-12-13T16:50:41Z",
    "eventSource": "ssm-guiconnect.amazonaws.com",
    "eventName": "StartConnection",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "34.230.45.60",
    "userAgent": "example-user-agent-string",
    "requestParameters": {
        "AuthType": "Credentials",
        "Protocol": "RDP",
        "ConnectionType": "SessionManager",
        "InstanceId": "i-02573cafcfEXAMPLE"
    },
    "responseElements": {
        "ConnectionArn": "arn:aws:ssm-guiconnect:us-east-2:123456789012:connection/
fcb810cd-241f-4aae-9ee4-02d59EXAMPLE",
        "ConnectionKey": "71f9629f-0f9a-4b35-92f2-2d253EXAMPLE",
        "ClientToken": "49af0f92-d637-4d47-9c54-ea51aEXAMPLE",
        "requestId": "d466710f-2adf-4e87-9464-055b2EXAMPLE"
    },
    "requestID": "d466710f-2adf-4e87-9464-055b2EXAMPLE",
    "eventID": "fc514f57-ba19-4e8b-9079-c2913EXAMPLE",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```

Exemplos de dados de eventos

Exemplo 1: **`CreateControlChannel`**

O exemplo a seguir mostra um evento do CloudTrail que demonstra a operação `CreateControlChannel`.

```
{
  "eventVersion":"1.08",
  "userIdentity":{
    "type":"AssumedRole",
    "principalId":"AKIAI44QH8DHBEXAMPLE",
    "arn":"arn:aws:sts::123456789012:assumed-role/exampleRole",
    "accountId":"123456789012",
    "accessKeyId":"AKIAI44QH8DHBEXAMPLE",
```

```
    "sessionContext":{
      "sessionIssuer":{
        "type":"Role",
        "principalId":"AKIAI44QH8DHBEXAMPLE",
        "arn":"arn:aws:iam::123456789012:role/exampleRole",
        "accountId":"123456789012",
        "userName":"exampleRole"
      },
      "attributes":{
        "creationDate":"2023-05-04T23:14:50Z",
        "mfaAuthenticated":"false"
      }
    }
  },
  "eventTime":"2023-05-04T23:53:55Z",
  "eventSource":"ssm.amazonaws.com",
  "eventName":"CreateControlChannel",
  "awsRegion":"us-east-1",
  "sourceIPAddress":"192.0.2.0",
  "userAgent":"example-agent",
  "requestParameters":{
    "channelId":"44295c1f-49d2-48b6-b218-96823EXAMPLE",
    "messageSchemaVersion":"1.0",
    "requestId":"54993150-0e8f-4142-aa54-3438EXAMPLE",
    "userAgent":"example-agent"
  },
  "responseElements":{
    "messageSchemaVersion":"1.0",
    "tokenValue":"Value hidden due to security reasons.",
    "url":"example-url"
  },
  "requestID":"54993150-0e8f-4142-aa54-3438EXAMPLE",
  "eventID":"a48a28de-7996-4ca1-a3a0-a51fEXAMPLE",
  "readOnly":false,
  "resources":[
    {
      "accountId":"123456789012",
      "type":"AWS::SSMMessages::ControlChannel",
      "ARN":"arn:aws:ssmmessages:us-east-1:123456789012:control-
channel/44295c1f-49d2-48b6-b218-96823EXAMPLE"
    }
  ],
  "eventType":"AwsApiCall",
  "managementEvent":false,
```

```
  "recipientAccountId":"123456789012",
  "eventCategory":"Data"
}
```

Exemplo 2: **RequestManagedInstanceRoleToken**

O exemplo a seguir mostra um evento do CloudTrail que demonstra a operação `RequestManagedInstanceRoleToken`.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "123456789012:aws:ec2-instance:i-02854e4bEXAMPLE",
        "arn": "arn:aws:sts::123456789012:assumed-role/aws:ec2-instance/
i-02854e4bEXAMPLE",
        "accountId": "123456789012",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "123456789012:aws:ec2-instance",
                "arn": "arn:aws:iam::123456789012:role/aws:ec2-instance",
                "accountId": "123456789012",
                "userName": "aws:ec2-instance"
            },
            "attributes": {
                "creationDate": "2023-08-27T03:34:46Z",
                "mfaAuthenticated": "false"
            },
            "ec2RoleDelivery": "2.0"
        }
    },
    "eventTime": "2023-08-27T03:37:15Z",
    "eventSource": "ssm.amazonaws.com",
    "eventName": "RequestManagedInstanceRoleToken",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Apache-HttpClient/UNAVAILABLE (Java/1.8.0_362)",
    "requestParameters": {
        "fingerprint": "i-02854e4bf85EXAMPLE"
    },
    "responseElements": null,
```

```
    "requestID": "2582cced-455b-4189-9b82-7b48EXAMPLE",
    "eventID": "7f200508-e547-4c27-982d-4da0EXAMLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "123456789012",
            "type": "AWS::SSM::ManagedNode",
            "ARN": "arn:aws:ec2:us-east-1:123456789012:instance/i-02854e4bEXAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "123456789012",
    "eventCategory": "Data"
}
```

Para obter informações sobre o conteúdo dos registros do CloudTrail, consulte Conteúdo dos registros do CloudTrail no Guia do usuário do AWS CloudTrail.

Registro de saída de ações do Automation em log com o CloudWatch Logs

O Automation, uma ferramenta do AWS Systems Manager, integra-se ao Amazon CloudWatch Logs. Você pode enviar o resultado das ações do `aws:executeScript` nos runbooks para o grupo de logs especificado. O Systems Manager não cria um grupo de logs ou quaisquer fluxos de log para documentos que não usam ações `aws:executeScript`. Se o documento usar `aws:executeScript`, a saída enviada ao CloudWatch Logs pertence somente a essas ações. Você pode usar a ação `aws:executeScript` armazenada no grupo de logs do CloudWatch Logs para fins de depuração e solução de problemas. Se você escolher um grupo de logs criptografado, a ação `aws:executeScript` também é criptografada. O registro em log de saída de ações `aws:executeScript` é uma configuração no nível da conta.

Para enviar saídas de ação ao CloudWatch Logs para runbooks de propriedade da Amazon, o usuário ou o perfil que executa a automação deve ter permissões para as operações a seguir:

- `logs:CreateLogGroup`
- `logs:CreateLogStream`
- `logs:DescribeLogGroups`
- `logs:DescribeLogStreams`

- `logs:PutLogEvents`

Para runbooks de sua propriedade, permissões semelhantes devem ser adicionadas ao perfil de serviço do IAM (ou AssumeRole) utilizado para executar o runbook.

Para enviar a saída de ação para o CloudWatch Logs (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação à esquerda, escolha Automation (Automação).
3. Escolha a guia Preferences (Preferências) e, em seguida, escolha Edit (Editar).
4. Marque a caixa de seleção ao lado de Send output to CloudWatch Logs (Enviar saída ao CloudWatch Logs).
5. (Recomendado) Marque a caixa de seleção ao lado de Encrypt log data (Criptografar dados de log). Com essa opção ativada, os dados de log serão criptografados usando a chave de criptografia no lado do servidor especificada para o grupo de logs. Se você não quiser criptografar os dados de log que são enviados ao CloudWatch Logs, desmarque a caixa de seleção. Desmarque a caixa de seleção se a criptografia não for permitida no grupo de logs.
6. Em CloudWatch Logs, para especificar o grupo de logs existente do CloudWatch Logs no Conta da AWS, para o qual você quer enviar o resultado da ação, selecione uma das seguintes opções:
   - Send output to the default log group (Enviar saída para o grupo de logs padrão): se o grupo de logs padrão não existir (`/aws/ssm/automation/executeScript`), o Automation o criará para você.
   - Escolha um grupo de logs na lista: selecione um grupo de logs que já tenha sido criado na sua conta para armazenar resultados práticos.
   - Insira um nome de grupo de logs na caixa de texto: insira o nome de um grupo de logs na caixa de texto que já tenha sido criado na conta para armazenar os resultados da ação.
7. Escolha Salvar.

Para enviar a saída de ação para o CloudWatch Logs (linha de comando)

1. Abra sua ferramenta da linha de comando preferencial e execute o comando a fim de atualizar o destino de saída da ação.

Linux & macOS

```
aws ssm update-service-setting \
    --setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/automation/
customer-script-log-destination \
    --setting-value CloudWatch
```

Windows

```
aws ssm update-service-setting ^
    --setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/automation/
customer-script-log-destination ^
    --setting-value CloudWatch
```

PowerShell

```
Update-SSMServiceSetting `
    -SettingId "arn:aws:ssm:region:account-id:servicesetting/ssm/automation/
customer-script-log-destination" `
    -SettingValue "CloudWatch"
```

Não haverá saída se o comando for bem-sucedido.

2. Execute o comando a seguir a fim de especificar o grupo de logs para o qual você deseja enviar a saída de ação.

Linux & macOS

```
aws ssm update-service-setting \
    --setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/automation/
customer-script-log-group-name \
    --setting-value my-log-group
```

Windows

```
aws ssm update-service-setting ^
    --setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/automation/
customer-script-log-group-name ^
    --setting-value my-log-group
```

PowerShell

```
Update-SSMServiceSetting `
    -SettingId "arn:aws:ssm:region:account-id:servicesetting/ssm/automation/
customer-script-log-group-name" `
    -SettingValue "my-log-group"
```

Não haverá saída se o comando for bem-sucedido.

3. Execute o comando a seguir para visualizar as configurações do serviço para as preferências de registro em log de ações do Automation na Conta da AWS e Região da AWS atuais.

Linux & macOS

```
aws ssm get-service-setting \
    --setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/automation/
customer-script-log-destination
```

Windows

```
aws ssm get-service-setting ^
    --setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/automation/
customer-script-log-destination
```

PowerShell

```
Get-SSMServiceSetting `
    -SettingId "arn:aws:ssm:region:account-id:servicesetting/ssm/automation/
customer-script-log-destination"
```

O comando retorna informações como as seguintes.

```
{
    "ServiceSetting": {
        "Status": "Customized",
        "LastModifiedDate": 1613758617.036,
        "SettingId": "/ssm/automation/customer-script-log-destination",
        "LastModifiedUser": "arn:aws:sts::123456789012:assumed-role/Administrator/
User_1",
```

```
        "SettingValue": "CloudWatch",
        "ARN": "arn:aws:ssm:us-east-2:123456789012:servicesetting/ssm/automation/
customer-script-log-destination"
    }
}
```

Configurar o Amazon CloudWatch Logs para Run Command

Quando você envia um comando usando o Run Command, uma ferramenta do AWS Systems Manager, é possível especificar para onde deseja enviar a saída do comando. Por padrão, o Systems Manager retorna apenas os primeiros 24.000 caracteres do resultado do comando. Se quiser visualizar os detalhes completos da saída do comando, você pode especificar um bucket do Amazon Simple Storage Service (Amazon S3). Ou você pode especificar o Amazon CloudWatch Logs. Se você especificar o CloudWatch Logs, o Run Command envia periodicamente todos os logs de erros e saídas de comandos para o CloudWatch Logs. Você pode monitorar os logs de saída quase em tempo real, pesquisar frases, valores ou padrões específicos e criar alarmes com base na pesquisa.

Se você configurou seu nó gerenciado para usar as políticas gerenciadas do AWS Identity and Access Management (IAM) `AmazonSSMManagedInstanceCore` e `CloudWatchAgentServerPolicy`, seu nó não requer configuração adicional para enviar saída ao CloudWatch Logs. Selecione essa opção se estiver enviando comandos do console ou adicione a seção `cloud-watch-output-config` e parâmetro `CloudWatchOutputEnabled`, se estiver usando a AWS Command Line Interface (AWS CLI), o AWS Tools for Windows PowerShell ou uma ação de API. A seção `cloud-watch-output-config` e o parâmetro `CloudWatchOutputEnabled` são descritos em mais detalhes posteriormente neste tópico.

Para obter informações sobre como adicionar políticas a um perfil de instância para instâncias do EC2, consulte Configurar permissões de instância obrigatórias para o Systems Manager. Para obter informações sobre como adicionar políticas a um perfil de serviço para servidores on-premises e máquinas virtuais que você planeja usar como nós gerenciados, consulte Criar o perfil de serviço do IAM obrigatório para o Systems Manager em ambientes híbridos e multinuvem.

Se você estiver usando uma política personalizada em seus nós, atualize a política em cada nó para permitir que o Systems Manager envie a saída e os logs para o CloudWatch Logs. Adicione os seguintes objetos de política para a sua política personalizada. Para obter mais informações sobre como atualizar uma política do IAM, consulte Editar políticas do IAM no Manual do usuário do IAM.

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

Especificar o CloudWatch Logs ao enviar comandos

Para especificar o CloudWatch Logs como a saída quando você enviar um comando a partir do AWS Management Console, escolha CloudWatch Output (Saída do CloudWatch) na seção Output options (Opções de saída). Opcionalmente, você pode especificar o nome do grupo do CloudWatch Logs para onde você deseja enviar uma saída de comando. Se você não especificar um nome de grupo, o Systems Manager cria automaticamente um grupo de logs para você. O grupo de logs usa o seguinte formato de nomenclatura: `/aws/ssm/`*`SystemsManagerDocumentName`*.

Se você executar comandos usando a AWS CLI, você deve especificar a seção `cloud-watch-output-config` no seu comando. Esta seção permite que você especifique o parâmetro `CloudWatchOutputEnabled` e, opcionalmente, o parâmetro `CloudWatchLogGroupName`. Aqui está um exemplo.

Linux & macOS

```
aws ssm send-command \
    --instance-ids "instance ID" \
    --document-name "AWS-RunShellScript" \
    --parameters "commands=echo helloWorld" \
    --cloud-watch-output-config
 "CloudWatchOutputEnabled=true,CloudWatchLogGroupName=log group name"
```

Windows

```
aws ssm send-command ^
    --document-name "AWS-RunPowerShellScript" ^
    --parameters commands=["echo helloWorld"] ^
    --targets "Key=instanceids,Values=an instance ID" ^
    --cloud-watch-output-config '{"CloudWatchLogGroupName":"log group
 name","CloudWatchOutputEnabled":true}'
```

Visualizar a saída de comandos no CloudWatch Logs

Assim que o comando começar a ser executado, o Systems Manager enviará o resultado para o CloudWatch Logs praticamente em tempo real. A saída no CloudWatch Logs usa o seguinte formato:

CommandID/*InstanceID*/*PluginID*/stdout

CommandID/*InstanceID*/*PluginID*/stderr

O resultado da execução é carregado a cada 30 segundos ou quando o buffer exceder 200 KB, o que acontecer primeiro.

> **Note**
>
> Fluxos de log são criados somente quando os dados de saída são disponibilizados. Por exemplo, se não houver dados de erro para uma execução, o stream stderr não é criado.

Veja a seguir um exemplo de saída do comando como ela é exibida no CloudWatch Logs.

```
Group - /aws/ssm/AWS-RunShellScript
Streams –
1234-567-8910/i-abcd-efg-hijk/AWS-RunPowerShellScript/stdout
24/1234-567-8910/i-abcd-efg-hijk/AWS-RunPowerShellScript/stderr
```

Monitorar eventos do Systems Manager com o Amazon EventBridge

Amazon EventBridge: é um serviço de barramento de eventos sem servidor que permite conectar as aplicações a dados de diversas fontes. O EventBridge fornece um fluxo de dados em tempo

real de suas próprias aplicações, de aplicações de software como serviço (SaaS) e de Serviços da AWS, roteando esses dados para destinos como o AWS Lambda. É possível configurar regras de roteamento que determinam o destino dos dados para criar arquiteturas de aplicações que reagem em tempo real a todas as fontes de dados. O EventBridge permite que você crie arquiteturas orientadas a eventos, que são vagamente acopladas e distribuídas.

Anteriormente, o EventBridge era chamado de Amazon CloudWatch Events. O EventBridge inclui novos recursos que permitem que você receba eventos de parceiros de SaaS e suas próprias aplicações. Os usuários do CloudWatch Events existentes do podem acessar o barramento, as regras e os eventos padrão existentes no novo console do EventBridge e no console do CloudWatch Events. O EventBridge usa a mesma API do CloudWatch Events, portanto, todo o uso existente da API do CloudWatch Events permanece o mesmo.

O EventBridge pode adicionar eventos de dezenas de Serviços da AWS às suas regras e destinos de mais de 20 Serviços da AWS.

O EventBridge fornece suporte para eventos do AWS Systems Manager e destinos do Systems Manager.

Tipos de evento do Systems Manager compatíveis

Entre os muitos tipos de eventos do Systems Manager que o EventBridge pode detectar estão:

- Uma atualização do status da solicitação de acesso a nós just-in-time para aprovações manuais.
- Uma solicitação de acesso a nós just-in-time que falhou.
- Uma janela de manutenção sendo desligada.
- Um fluxo de trabalho do Automation concluído com êxito. O Automation é uma ferramenta do AWS Systems Manager.
- Um nó gerenciado que não está em conformidade com os patches.
- O valor de um parâmetro sendo atualizado.

O EventBridge oferece suporte a eventos das seguintes ferramentas do AWS Systems Manager:

- Acesso a nós just-in-time (Os eventos são emitidos com base no melhor esforço.)
- Automation (Eventos são emitidos com base no melhor esforço).
- Change Calendar (Eventos são emitidos com base no melhor esforço).
- Conformidade

- Inventário (Eventos são emitidos com base no melhor esforço).
- Maintenance Windows (Eventos são emitidos com base no melhor esforço).
- Parameter Store (Eventos são emitidos com base no melhor esforço).
- Run Command (Eventos são emitidos com base no melhor esforço).
- State Manager (Eventos são emitidos com base no melhor esforço).

Para obter detalhes completos sobre os tipos de evento do Systems Manager suportados, consulte Referência: padrões e tipos de eventos do Amazon EventBridge para o Systems Manager e Exemplos de eventos do Amazon EventBridge para Systems Manager.

Tipos de destino do Systems Manager compatíveis

O EventBridge oferece suporte às seguintes três ferramentas do Systems Manager como destinos de uma regra de evento:

- Executar um fluxo de trabalho do Automation
- Executar um documento de comando do Run Command (Os eventos são emitidos com base no melhor esforço).
- Criar um OpsCenter OpsItem

Para obter sugestões de como usar esses destinos, consulte Cenários de exemplo: destinos do Systems Manager em regras do Amazon EventBridge.

Para obter mais informações sobre o EventBridge e configurar regras, consulte Conceitos básicos do Amazon EventBridge no Guia do usuário do Amazon EventBridge. Para obter informações completas sobre como trabalhar com o EventBridge, consulte o Manual do usuário do Amazon EventBridge.

Tópicos

- Configurar o EventBridge para eventos do Systems Manager
- Exemplos de eventos do Amazon EventBridge para Systems Manager
- Cenários de exemplo: destinos do Systems Manager em regras do Amazon EventBridge

Configurar o EventBridge para eventos do Systems Manager

Você pode usar o Amazon EventBridge para executar um evento de destino quando o status de AWS Systems Manager compatíveis for alterado, o estado mudar ou outras condições ocorrem.

Você pode criar uma regra que será executada sempre que houver uma transição de estado ou status, ou quando houver uma transição para um ou mais estados que sejam de interesse.

O procedimento a seguir fornece etapas gerais para criar uma regra do EventBridge que se envolve quando um evento especificado é emitido pelo Systems Manager. Para obter uma lista de procedimentos neste manual do usuário que aborda cenários específicos, consulte Mais informações, no final deste tópico.

> **Note**
>
> Quando um serviço da Conta da AWS emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta. Para escrever uma regra que responde a eventos de Serviços da AWS na sua conta, associe-a ao barramento de eventos padrão. Você pode criar uma regra em um barramento de eventos personalizado que procura eventos de Serviços da AWS, mas essa regra será acionada somente quando você receber esse evento de outra conta por meio da entrega de eventos entre contas. Para obter mais informações, consulte Enviar e receber eventos do Amazon EventBridge entre Contas da AWS no Guia do usuário do Amazon EventBridge.

Para configurar o EventBridge para eventos do Systems Manager

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Regras.
3. Escolha Create rule.
4. Insira um nome e uma descrição para a regra.

   Uma regra não pode ter o mesmo nome que outra regra na mesma Região da AWS e no mesmo barramento de eventos.
5. Em Barramento de Eventos, escolha o barramento de eventos que você deseja associar a essa regra. Se você quiser que essa regra responda a eventos correspondentes provenientes da sua Conta da AWS, selecione default (padrão). Quando um AWS service (Serviço da AWS) na sua conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta.
6. Em Tipo de Regra, escolha Regra com Padrão de Evento.
7. Escolha Próximo.
8. Em Event source, escolha Eventos da AWS ou eventos de parceiro do EventBridge.

9. Na seção Event patter (Padrão de evento), selecione Event pattern form (Formulário de padrão de evento).
10. Em Fonte do evento, selecione Serviços da AWS.
11. Em Serviço da AWS, escolha Systems Manager.
12. Em Event type (Tipo de evento), siga um destes procedimentos:

    - Escolha All Events (Todos os eventos).

      Se você selecionar All Events (Todos os eventos), todos os eventos emitidos pelo Systems Manager corresponderão à regra. Observe que essa opção pode resultar em muitas ações direcionadas ao evento.

    - Selecione o tipo de evento do Systems Manager a ser usado para essa regra. O EventBridge oferece suporte a eventos das seguintes ferramentas do AWS Systems Manager:
      - Automação
      - Change Calendar
      - Compliance
      - Inventory
      - Maintenance Windows
      - Parameter Store
      - Run Command
      - State Manager

      > **Note**
      >
      > Para ações do Systems Manager, que não sejam compatíveis com o EventBridge, escolha uma chamada de API da AWS por meio do CloudTrail para criar uma regra de evento baseada em uma chamada de API, que é registrada pelo CloudTrail. Para obter um exemplo, consulte Monitorar as atividades da sessão usando o Amazon EventBridge (console).

13. (Opcional) Para tornar a regra mais específica, adicione valores de filtros. Por exemplo, se você escolheu State Manager e deseja limitar a regra ao estado de uma instância única gerenciada que é direcionada por uma associação, para Specific type(s) (Tipos específicos), escolha EC2 State Manager Instance Association State Change (Alteração de estado da associação de instância do EC2 State Manager).

Para obter detalhes completos sobre os tipos de detalhes compatíveis, consulte Referência: padrões e tipos de eventos do Amazon EventBridge para o Systems Manager.

Alguns tipos de detalhes têm outras opções compatíveis, como status. As opções disponíveis dependem da ferramenta que você selecionou.

14. Escolha Next (Avançar).
15. Em Target types (Tipos de destinos), escolha AWS service (Serviço da).
16. Em Select a target (Selecione um destino), escolha um destino, como um tópico do Amazon SNS ou uma função do AWS Lambda. O destino é acionado quando é recebido um evento que corresponde ao padrão de evento definido na regra.
17. Para muitos tipos de destino, o Eventbridge precisa de permissões para enviar eventos ao destino. Nesses casos, o Eventbridge pode criar a função do AWS Identity and Access Management (IAM) necessária para que sua regra seja executada.
    - Para criar um perfil do IAM automaticamente, escolha Criar novo perfil para este recurso específico.
    - Para usar um perfil do IAM criado anteriormente, escolha Usar função existente
18. (Opcional) Selecione Adicionar outro destino para adicionar outro destino a essa regra.
19. Escolha Next (Próximo).
20. (Opcional) Insira uma ou mais tags para a regra. Para mais informações, consulte Tags Amazon EventBridge em Guia de Usuário Amazon EventBridge.
21. Escolha Próximo.
22. Analise os detalhes da regra e selecione Criar regra.

Mais informações

- Criar um evento do EventBridge que use um runbook (console)
- Transferência de dados para o Automation usando transformadores de entrada
- Corrija problemas de conformidade usando o EventBridge
- Visualizar ações de exclusão de inventário no EventBridge
- Configurar regras do EventBridge para criar OpsItems
- Configurar regras do EventBridge para parâmetros e políticas de parâmetros

Exemplos de eventos do Amazon EventBridge para Systems Manager

Veja a seguir exemplos, no formato JSON, de eventos EventBridge com suporte para o AWS Systems Manager.

Tipos de evento do Systems Manager

- Eventos do Automation do AWS Systems Manager
- AWS Systems ManagerEventosChange Calendar
- AWS Systems ManagerEventosChange Manager
- Eventos do Compliance do AWS Systems Manager
- AWS Systems ManagerEventosMaintenance Windows
- AWS Systems ManagerEventosParameter Store
- AWS Systems ManagerEventosOpsCenter
- AWS Systems ManagerEventosRun Command
- AWS Systems ManagerEventosState Manager

Eventos do Automation do AWS Systems Manager

Notificação de alteração do status da etapa de automação


```
{
  "version": "0",
  "id": "eeca120b-a321-433e-9635-dab369006a6b",
  "detail-type": "EC2 Automation Step Status-change Notification",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2024-11-29T19:43:35Z",
  "region": "us-east-1",
  "resources": ["arn:aws:ssm:us-east-2:123456789012:automation-
execution/333ba70b-2333-48db-b17e-a5e69c6f4d1c",
    "arn:aws:ssm:us-east-2:123456789012:automation-definition/runcommand1:1"],
  "detail": {
    "ExecutionId": "333ba70b-2333-48db-b17e-a5e69c6f4d1c",
    "Definition": "runcommand1",
    "DefinitionVersion": 1.0,
    "Status": "Success",
    "EndTime": "Nov 29, 2024 7:43:25 PM",
```

```
    "StartTime": "Nov 29, 2024 7:43:23 PM",
    "Time": 2630.0,
    "StepName": "runFixedCmds",
    "Action": "aws:runCommand"
  }
}
```

Notificação de alteração de status de execução de automação

```
{
  "version": "0",
  "id": "d290ece9-1088-4383-9df6-cd5b4ac42b99",
  "detail-type": "EC2 Automation Execution Status-change Notification",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2024-11-29T19:43:35Z",
  "region": "us-east-2",
  "resources": ["arn:aws:ssm:us-east-2:123456789012:automation-
execution/333ba70b-2333-48db-b17e-a5e69c6f4d1c",
    "arn:aws:ssm:us-east-2:123456789012:automation-definition/runcommand1:1"],
  "detail": {
    "ExecutionId": "333ba70b-2333-48db-b17e-a5e69c6f4d1c",
    "Definition": "runcommand1",
    "DefinitionVersion": 1.0,
    "Status": "Success",
    "StartTime": "Nov 29, 2024 7:43:20 PM",
    "EndTime": "Nov 29, 2024 7:43:26 PM",
    "Time": 5753.0,
    "ExecutedBy": "arn:aws:iam::123456789012:user/userName"
  }
}
```

AWS Systems ManagerEventosChange Calendar

Use as informações deste tópico para planejar e entender o comportamento dos eventos do EventBridge para o AWS Systems Manager Change Calendar.

> **Note**
>
> No momento, não são permitidas alterações de estado para calendários compartilhados de outras Contas da AWS.

Integração do Change Calendar com o Amazon EventBridge

O AWS Systems Manager Change Calendar se integra ao Amazon EventBridge para notificar você sobre mudanças no estado do calendário. Esteja ciente dos seguintes comportamentos relacionados à arquitetura de agendamento subjacente:

Tempo e confiabilidade dos eventos

- O EventBridge entrega notificações com base no melhor esforço possível, com uma tolerância de programação de até 15 minutos.
- Os eventos de mudança de estado refletem as transições gerais do status do calendário, e não eventos individuais do calendário.
- Quando vários eventos do calendário ocorrem simultaneamente, o EventBridge gera somente um evento por mudança real de estado do calendário.
- O EventBridge aciona eventos somente quando o estado geral do calendário muda (por exemplo, de FECHADO para ABERTO), e não para eventos individuais do calendário que não resultam em uma mudança de estado.
- Eventos consultivos que não modificam o estado do calendário não acionam as notificações do EventBridge.

Considerações sobre modificações e tempo dos eventos

- Se você modificar os eventos do calendário em até 15 minutos do horário de início ou término programado, o EventBridge poderá gerar notificações duplicadas ou notificações perdidas.
- Esse comportamento ocorre porque o sistema de programação pode não ter tempo suficiente para atualizar ou cancelar adequadamente as notificações programadas anteriormente.
- Para eventos recorrentes, esse comportamento geralmente afeta somente a primeira ocorrência após a modificação.

Eventos adjacentes e sobrepostos

- Quando os eventos do calendário são programados com 5 minutos de diferença um do outro, os eventos de transição de estado podem ou não ocorrer, dependendo da mudança de estado real.
- A criação de eventos sobrepostos em determinadas ordens pode gerar eventos adicionais do EventBridge mesmo quando nenhuma mudança de estado real ocorre.
- Para garantir um comportamento previsível, evite criar ou modificar eventos do calendário próximos aos horários de execução.

Práticas recomendadas

- Crie suas regras do EventBridge e a automação downstream para lidar com possíveis eventos duplicados.
- Implemente a idempotência em seus fluxos de trabalho de automação para evitar problemas causados por notificações duplicadas.
- Reserve antecedência suficiente (pelo menos 15 minutos) ao criar ou modificar eventos do calendário.
- Teste suas integrações do EventBridge minuciosamente com os padrões de eventos do calendário específico.

Calendário ABERTO

```
{
    "version": "0",
    "id": "47a3f03a-f30d-1011-ac9a-du3bdEXAMPLE",
    "detail-type": "Calendar State Change",
    "source": "aws.ssm",
    "account": "123456789012",
    "time": "2024-09-19T18:00:07Z",
    "region": "us-east-2",
    "resources": [
        "arn:aws:ssm:us-east-2:123456789012:document/MyCalendar"
    ],
    "detail": {
        "state": "OPEN",
        "atTime": "2024-09-19T18:00:07Z",
        "nextTransitionTime": "2024-10-11T18:00:07Z"
    }
}
```

Calendário FECHADO

```
{
    "version": "0",
    "id": "f30df03a-1011-ac9a-47a3-f761eEXAMPLE",
    "detail-type": "Calendar State Change",
    "source": "aws.ssm",
    "account": "123456789012",
    "time": "2024-09-17T21:40:02Z",
    "region": "us-east-2",
```

```
    "resources": [
        "arn:aws:ssm:us-east-2:123456789012:document/MyCalendar"
    ],
    "detail": {
        "state": "CLOSED",
        "atTime": "2024-08-17T21:40:00Z",
        "nextTransitionTime": "2024-09-19T18:00:07Z"
    }
}
```

AWS Systems ManagerEventosChange Manager

Notificação de atualização do status da solicitação de alteração: exemplo 1

```
{
  "version": "0",
  "id": "feab80c1-a8ff-c721-b8b1-96ce70939696",
  "detail-type": "Change Request Status Update",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2024-10-24T10:51:52Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ssm:us-west-2:123456789012:opsitem/oi-12345abcdef",
    "arn:aws:ssm:us-west-2:123456789012:document/MyRunbook1"
  ],
  "detail": {
    "change-request-id": "d0585556-80f6-4522-8dad-dada6d45b67d",
    "change-request-title": "A change request title",
    "ops-item-id": "oi-12345abcdef",
    "ops-item-created-by": "arn:aws:iam::123456789012:user/JohnDoe",
    "ops-item-created-time": "2024-10-24T10:50:33.180334Z",
    "ops-item-modified-by": "arn:aws:iam::123456789012:user/JohnDoe",
    "ops-item-modified-time": "2024-10-24T10:50:33.180340Z",
    "ops-item-status": "InProgress",
    "change-template-document-name": "MyChangeTemplate",
    "runbook-document-arn": "arn:aws:ssm:us-west-2:123456789012:document/MyRunbook1",
    "runbook-document-version": "1",
    "auto-approve": true,
    "approvers": [
      "arn:aws:iam::123456789012:user/JaneDoe"
    ]
  }
```

```
}
```

Notificação de atualização do status da solicitação de alteração: exemplo 2

```
{
  "version": "0",
  "id": "25ce6b03-2e4e-1a2b-2a8f-6c9de8d278d2",
  "detail-type": "Change Request Status Update",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2024-10-24T10:51:52Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ssm:us-west-2:123456789012:opsitem/oi-abcdef12345",
    "arn:aws:ssm:us-west-2:123456789012:document/MyRunbook1"
  ],
  "detail": {
    "change-request-id": "d0585556-80f6-4522-8dad-dada6d45b67d",
    "change-request-title": "A change request title",
    "ops-item-id": "oi-abcdef12345",
    "ops-item-created-by": "arn:aws:iam::123456789012:user/JohnDoe",
    "ops-item-created-time": "2024-10-24T10:50:33.180334Z",
    "ops-item-modified-by": "arn:aws:iam::123456789012:user/JohnDoe",
    "ops-item-modified-time": "2024-10-24T10:50:33.997163Z",
    "ops-item-status": "Rejected",
    "change-template-document-name": "MyChangeTemplate",
    "runbook-document-arn": "arn:aws:ssm:us-west-2:123456789012:document/MyRunbook1",
    "runbook-document-version": "1",
    "auto-approve": true,
    "approvers": [
      "arn:aws:iam::123456789012:user/JaneDoe"
    ]
  }
}
```

Eventos do Compliance do AWS Systems Manager

Veja a seguir exemplos dos eventos do Compliance do AWS Systems Manager.

Conformidade de associação

```
{
  "version": "0",
```

```
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2024-07-17T19:03:26Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:ssm:us-east-2:123456789012:managed-instance/i-01234567890abcdef"
  ],
  "detail": {
    "last-runtime": "2024-01-01T10:10:10Z",
    "compliance-status": "compliant",
    "resource-type": "managed-instance",
    "resource-id": "i-01234567890abcdef",
    "compliance-type": "Association"
  }
}
```

Não conformidade de associação

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2024-07-17T19:02:31Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:ssm:us-east-2:123456789012:managed-instance/i-01234567890abcdef"
  ],
  "detail": {
    "last-runtime": "2024-01-01T10:10:10Z",
    "compliance-status": "non_compliant",
    "resource-type": "managed-instance",
    "resource-id": "i-01234567890abcdef",
    "compliance-type": "Association"
  }
}
```

Conformidade de patch

```
{
```

```
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.123456789012",
  "account": "123456789012",
  "time": "2024-07-17T19:03:26Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:ssm:us-east-2:123456789012:managed-instance/i-01234567890abcdef"
  ],
  "detail": {
    "resource-type": "managed-instance",
    "resource-id": "i-01234567890abcdef",
    "compliance-status": "compliant",
    "compliance-type": "Patch",
    "patch-baseline-id": "PB789",
    "severity": "critical"
  }
}
```

Não conformidade de patch

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2024-07-17T19:02:31Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:ssm:us-east-2:123456789012:managed-instance/i-01234567890abcdef"
  ],
  "detail": {
    "resource-type": "managed-instance",
    "resource-id": "i-01234567890abcdef",
    "compliance-status": "non_compliant",
    "compliance-type": "Patch",
    "patch-baseline-id": "PB789",
    "severity": "critical"
  }
}
```

AWS Systems ManagerEventosMaintenance Windows

Veja a seguir exemplos dos eventos do Maintenance Windows para o Systems Manager.

Registrar um destino

Os valores de status válidos incluem `REGISTERED` e `DEREGISTERED`.

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Target Registration Notification",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2024-11-16T00:58:37Z",
   "region":"us-east-2",
   "resources":[
      "arn:aws:ssm:us-east-2:123456789012:maintenancewindow/mw-0ed7251d3fcf6e0c2",
      "arn:aws:ssm:us-east-2:123456789012:windowtarget/
e7265f13-3cc5-4f2f-97a9-7d3ca86c32a6"
   ],
   "detail":{
      "window-target-id":"e7265f13-3cc5-4f2f-97a9-7d3ca86c32a6",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "status":"REGISTERED"
   }
}
```

Tipo de execução da janela

Confira os valores de status válidos:

- `CANCELLED`
- `CANCELLING`
- `FAILED`
- `IN_PROGRESS`
- `PENDING`
- `SKIPPED_OVERLAPPING`
- `SUCCESS TIMED_OUT`

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Execution State-change Notification",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2025-06-02T14:52:18Z",
   "region":"us-east-2",
   "resources":[
      "arn:aws:ssm:us-east-2:123456789012:maintenancewindow/mw-0c50858d01EXAMPLE"
   ],
   "detail":{
      "start-time":"2025-06-02T14:48:28.039273Z",
      "end-time":"2025-06-02T14:52:18.083773Z",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "window-execution-id":"14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE",
      "status":"SUCCESS"
   }
}
```

Tipo de execução de tarefa

Os valores de status válidos incluem `IN_PROGRESS`, `SUCCESS`, `FAILED` e `TIMED_OUT`.

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Task Execution State-change Notification",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2025-06-02T14:52:18Z",
   "region":"us-east-2",
   "resources":[
      "arn:aws:ssm:us-east-2:123456789012:maintenancewindow/mw-0c50858d01EXAMPLE"
   ],
   "detail":{
      "start-time":"2025-06-02T14:48:28.039273Z",
      "task-execution-id":"6417e808-7f35-4d1a-843f-123456789012",
      "end-time":"2025-06-02T14:52:18.083773Z",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "window-execution-id":"14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE",
      "status":"SUCCESS"
   }
```

```
}
```

Destino de tarefa processada

Os valores de status válidos incluem `IN_PROGRESS`, `SUCCESS`, `FAILED` e `TIMED_OUT`.

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Task Target Invocation State-change Notification",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2025-06-02T14:52:18Z",
   "region":"us-east-2",
   "resources":[
      "arn:aws:ssm:us-east-2:123456789012:maintenancewindow/mw-123456789012345678"
   ],
   "detail":{
      "start-time":"2025-06-02T14:48:28.039273Z",
      "end-time":"2025-06-02T14:52:18.083773Z",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "window-execution-id":"791b72e0-f0da-4021-8b35-f95dfEXAMPLE",
      "task-execution-id":"c9b05aba-197f-4d8d-be34-e73fbEXAMPLE",
      "window-target-id":"e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE",
      "status":"SUCCESS",
      "owner-information":"Owner"
   }
}
```

Alteração do estado da janela

Os valores de status válidos incluem `ENABLED` e `DISABLED`.

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window State-change Notification",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2024-11-16T00:58:37Z",
   "region":"us-east-2",
   "resources":[
      "arn:aws:ssm:us-east-2:123456789012:maintenancewindow/mw-0c50858d01EXAMPLE"
```

```
   ],
   "detail":{
      "window-id":"mw-0c50858d01EXAMPLE",
      "status":"DISABLED"
   }
}
```

AWS Systems ManagerEventosParameter Store

Veja a seguir exemplos dos eventos do Parameter Store para o Systems Manager.

Criar parâmetro

```
{
  "version": "0",
  "id": "6a7e4feb-b491-4cf7-a9f1-bf3703497718",
  "detail-type": "Parameter Store Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2024-05-22T16:43:48Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:ssm:us-east-2:123456789012:parameter/MyExampleParameter"
  ],
  "detail": {
    "operation": "Create",
    "name": "MyExampleParameter",
    "type": "String",
    "description": "Sample Parameter"
  }
}
```

Atualizar parâmetro

```
{
  "version": "0",
  "id": "9547ef2d-3b7e-4057-b6cb-5fdf09ee7c8f",
  "detail-type": "Parameter Store Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2024-05-22T16:44:48Z",
```

```
  "region": "us-east-2",
  "resources": [
    "arn:aws:ssm:us-east-2:123456789012:parameter/MyExampleParameter"
  ],
  "detail": {
    "operation": "Update",
    "name": "MyExampleParameter",
    "type": "String",
    "description": "Sample Parameter"
  }
}
```

Excluir parâmetro

```
{
  "version": "0",
  "id": "80e9b391-6a9b-413c-839a-453b528053af",
  "detail-type": "Parameter Store Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2024-05-22T16:45:48Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:ssm:us-east-2:123456789012:parameter/MyExampleParameter"
  ],
  "detail": {
    "operation": "Delete",
    "name": "MyExampleParameter",
    "type": "String",
    "description": "Sample Parameter"
  }
}
```

AWS Systems ManagerEventosOpsCenter

Criar notificações do OpsCenter OpsItem

```
{
  "version": "0",
  "id": "aae66adc-7aac-f0c0-7854-7691e8c079b8",
  "detail-type": "OpsItem Create",
  "source": "aws.ssm",
  "account": "123456789012",
```

```
  "time": "2024-10-19T02:48:11Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ssm:us-west-2:123456789012:opsitem/oi-123456abcdef"
  ],
  "detail": {
    "created-by": "arn:aws:iam::123456789012:user/JohnDoe",
    "created-time": "2024-10-19T02:46:53.629361Z",
    "source": "aws.ssm",
    "status": "Open",
    "ops-item-id": "oi-123456abcdef",
    "title": "An issue title",
    "ops-item-type": "/aws/issue",
    "description": "A long description may appear here"
  }
}
```

Atualizar notificação do OpsCenter OpsItem

```
{
  "version": "0",
  "id": "2fb5b168-b725-41dd-a890-29311200089c",
  "detail-type": "OpsItem Update",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2024-10-19T02:48:11Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ssm:us-west-2:123456789012:opsitem/oi-123456abcdef"
  ],
  "detail": {
    "created-by": "arn:aws:iam::123456789012:user/JohnDoe",
    "created-time": "2024-10-19T02:46:54.049271Z",
    "modified-by": "arn:aws:iam::123456789012:user/JohnDoe",
    "modified-time": "2024-10-19T02:46:54.337354Z",
    "source": "aws.ssm",
    "status": "Open",
    "ops-item-id": "oi-123456abcdef",
    "title": "An issue title",
    "ops-item-type": "/aws/issue",
    "description": "A long description may appear here"
  }
}
```

AWS Systems ManagerEventosRun Command

Run Command Notificação de alteração de status

```
{
    "version": "0",
    "id": "51c0891d-0e34-45b1-83d6-95db273d1602",
    "detail-type": "EC2 Command Status-change Notification",
    "source": "aws.ssm",
    "account": "123456789012",
    "time": "2024-07-10T21:51:32Z",
    "region": "us-east-2",
    "resources": ["arn:aws:ec2:us-east-2:123456789012:instance/i-02573cafcfEXAMPLE"],
    "detail": {
        "command-id": "e8d3c0e4-71f7-4491-898f-c9b35bee5f3b",
        "document-name": "AWS-RunPowerShellScript",
        "expire-after": "2024-07-14T22:01:30.049Z",
        "parameters": {
            "executionTimeout": ["3600"],
            "commands": ["date"]
        },
        "requested-date-time": "2024-07-10T21:51:30.049Z",
        "status": "Success"
    }
}
```

Notificação de alteração de status de invocação do Run Command

```
{
    "version": "0",
    "id": "4780e1b8-f56b-4de5-95f2-95dbEXAMPLE",
    "detail-type": "EC2 Command Invocation Status-change Notification",
    "source": "aws.ssm",
    "account": "123456789012",
    "time": "2024-07-10T21:51:32Z",
    "region": "us-east-2",
    "resources": ["arn:aws:ec2:us-east-2:123456789012:instance/i-02573cafcfEXAMPLE"],
    "detail": {
        "command-id": "e8d3c0e4-71f7-4491-898f-c9b35bee5f3b",
        "document-name": "AWS-RunPowerShellScript",
        "instance-id": "i-02573cafcfEXAMPLE",
        "requested-date-time": "2024-07-10T21:51:30.049Z",
        "status": "Success"
```

```
    }
}
```

AWS Systems ManagerEventosState Manager

Alteração do estado do State Manager

```
{
   "version":"0",
   "id":"db839caf-6f6c-40af-9a48-25b2ae2b7774",
   "detail-type":"EC2 State Manager Association State Change",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2024-05-16T23:01:10Z",
   "region":"us-east-2",
   "resources":[
      "arn:aws:ssm:us-east-2::document/AWS-RunPowerShellScript"
   ],
   "detail":{
      "association-id":"6e37940a-23ba-4ab0-9b96-5d0a1a05464f",
      "document-name":"AWS-RunPowerShellScript",
      "association-version":"1",
      "document-version":"Optional.empty",
      "targets":"[{\"key\":\"InstanceIds\",\"values\":[\"i-12345678\"]}]",
      "creation-date":"2024-02-13T17:22:54.458Z",
      "last-successful-execution-date":"2024-05-16T23:00:01Z",
      "last-execution-date":"2024-05-16T23:00:01Z",
      "last-updated-date":"2024-02-13T17:22:54.458Z",
      "status":"Success",
      "association-status-aggregated-count":"{\"Success\":1}",
      "schedule-expression":"cron(0 */30 * * * ? *)",
      "association-cwe-version":"1.0"
   }
}
```

Alteração do estado da associação da instância do State Manager

```
{
   "version":"0",
   "id":"6a7e8feb-b491-4cf7-a9f1-bf3703467718",
   "detail-type":"EC2 State Manager Instance Association State Change",
   "source":"aws.ssm",
```

```
   "account":"123456789012",
   "time":"2024-02-23T15:23:48Z",
   "region":"us-east-2",
   "resources":[
      "arn:aws:ec2:us-east-2:123456789012:instance/i-12345678",
      "arn:aws:ssm:us-east-2:123456789012:document/my-custom-document"
   ],
   "detail":{
      "association-id":"34fcb7e0-9a14-4984-9989-0e04e3f60bd8",
      "instance-id":"i-02573cafcfEXAMPLE",
      "document-name":"my-custom-document",
      "document-version":"1",
      "targets":"[{\"key\":\"instanceids\",\"values\":[\"i-02573cafcfEXAMPLE\"]}]",
      "creation-date":"2024-02-23T15:23:48Z",
      "last-successful-execution-date":"2024-02-23T16:23:48Z",
      "last-execution-date":"2024-02-23T16:23:48Z",
      "status":"Success",
      "detailed-status":"",
      "error-code":"testErrorCode",
      "execution-summary":"testExecutionSummary",
      "output-url":"sampleurl",
      "instance-association-cwe-version":"1"
   }
}
```

Cenários de exemplo: destinos do Systems Manager em regras do Amazon EventBridge

Ao especificar o destino a ser invocado em uma regra do Amazon EventBridge, você pode escolher entre mais de 20 tipos de destinos e adicionar até cinco destinos a cada regra.

Entre os vários destinos, é possível escolher entre Automation, OpsCenter e Run Command, que são ferramentas do AWS Systems Manager, como ações de destino quando um evento do EventBridge ocorre.

Veja a seguir vários exemplos de maneiras como você pode usar essas ferramentas como o destino de uma regra do EventBridge.

Exemplos de automação

Você pode configurar uma regra do EventBridge para iniciar fluxos de trabalho do Automation quando ocorrerem eventos como os seguintes:

- Quando um alarme do Amazon CloudWatch relatar que um nó gerenciado mostrou falha em uma verificação de status (`StatusCheckFailed_Instance=1`), execute o runbook do Automation `AWSSupport-ExecuteEC2Rescue` nesse nó.
- Quando um evento `EC2 Instance State-change Notification` ocorrer porque uma nova instância do Amazon Elastic Compute Cloud (Amazon EC2) está em execução, execute o runbook `AWS-AttachEBSVolume` do Automation na instância.
- Quando um volume do Amazon Elastic Block Store (Amazon EBS) for criado e disponibilizado, execute o runbook `AWS-CreateSnapshot` do Automation no volume.

Exemplos do OpsCenter

Você pode configurar uma regra do EventBridge para criar um novo OpsItem quando ocorrerem incidentes como os seguintes:

- Um evento de controle de utilização está ocorrendo para o Amazon DynamoDB ou a performance do volume do Amazon EBS foi reduzida.
- Um grupo do Amazon EC2 Auto Scaling falhará ao executar um nó gerenciado ou se um fluxo de trabalho do Systems Manager Automation falhar.
- Uma instância do EC2 altera o estado de `Running` para `Stopped`.

Exemplos do Run Command

Você pode configurar uma regra do EventBridge para executar um documento de comando do Systems Manager no Run Command quando ocorrerem eventos como os seguintes:

- Quando um grupo do Auto Scaling estiver prestes a terminar, um script Run Command poderá capturar os arquivos de log do nó, antes que ele seja encerrado.
- Quando um novo nó for criado em um grupo do Auto Scaling, uma ação de destino do Run Command pode ativar a função de servidor Web ou instalar o software nesse nó.
- Quando um nó gerenciado estiver fora da conformidade, uma ação de destino Run Command pode atualizar os patches nesse nó executando o documento `AWS-RunPatchBaseline`.

Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS

É possível configurar o Amazon Simple Notification Service (Amazon SNS) para enviar notificações sobre o status dos comandos que você envia usando o Run Command ou Maintenance Windows, ambos ferramentas do AWS Systems Manager. O Amazon SNS coordena e gerencia a entrega e o envio de notificações a clientes e endpoints que assinam tópicos do Amazon SNS. Você pode receber uma notificação sempre que um comando muda para um novo estado ou atinge um estado específico, como Failed (Falha) ou Timed out (Tempo limite). Nos casos em que você envia um comando para vários nós, você pode receber uma notificação para cada cópia do comando enviado para um nó específico. Cada cópia é chamada de uma invocação.

O Amazon SNS pode entregar notificações como HTTP ou HTTPS POST, e-mail (SMTP, texto sem formatação ou no formato JSON) ou como uma mensagem postada em uma fila do Amazon Simple Queue Service (Amazon SQS). Para obter mais informações, consulte O que é o Amazon SNS no Guia do desenvolvedor do Amazon Simple Notification Service. Para obter exemplos da estrutura dos dados JSON incluídos na notificação do Amazon SNS fornecida pelo Run Command e Maintenance Windows, consulte Exemplo de notificações do Amazon SNS para AWS Systems Manager.

Important

Observe as seguintes informações importantes:

- Não há suporte para tópicos FIFO do Amazon Simple Notification Service.
- O Amazon Q Developer em aplicações de chat não é compatível com o monitoramento do Systems Manager com o Amazon SNS. Caso deseje usar o Amazon Q Developer para monitorar o Systems Manager, será necessário usá-lo com o Amazon EventBridge. Para obter informações sobre como monitorar o Systems Manager usando o EventBridge, consulte Monitorar eventos do Systems Manager com o Amazon EventBridge. Para obter informações sobre o Amazon EventBridge e o Amazon Q Developer em aplicações de chat, consulte Tutorial: criação de uma regra do EventBridge que envia notificações para o Amazon Q Developer em aplicações de chat no Guia do administrador do Amazon Q Developer em aplicações de chat.

Configurar notificações do Amazon SNS para o AWS Systems Manager

As tarefas Run Command e Maintenance Windows, registradas para uma janela de manutenção, podem enviar notificações do Amazon SNS para tarefas de comando que entram nos seguintes status:

- Em andamento
- Bem-sucedida
- Falha
- Timed Out
- Cancelado

Para obter informações sobre as condições que fazem com que um comando entre em um desses status, consulte Noções básicas sobre status de comando.

> **Note**
>
> Os comandos enviados usando o Run Command também relatam o status Cancelando e Pendente. Esses status não são capturados pelas notificações do Amazon SNS.

Notificações do Amazon SNS de resumo de comandos

Se você configurar o Run Command ou uma tarefa do Run Command na janela de manutenção para notificações do Amazon SNS, o Amazon SNS enviará mensagens de resumo que incluem as informações a seguir.

Campo	Tipo	Descrição
eventTime	String	A hora em que o evento foi iniciado. O carimbo de hora é importante, pois o Amazon SNS não garante a ordem de entrega das mensagens. Exemplo: 2016-04-26T13:15:30Z

Campo	Tipo	Descrição
documentName	String	O nome do documento do SSM usado para executar esse comando.
commandId	String	O ID gerado pelo Run Command após o envio do comando.
expiresAfter	Data	Se esse tempo for atingido e o comando ainda não tiver iniciado a execução, ele não executará.
outputS3BucketName	String	O bucket do Amazon Simple Storage Service (Amazon S3) em que respostas para a execução do comando devem ser armazenadas.
outputS3KeyPrefix	String	O caminho de diretório do Amazon S3 dentro do bucket no qual as respostas para a execução do comando devem ser armazenadas.
requestedDateTime	String	A data e a hora em que a solicitação foi enviada para esse nó específico.

Campo	Tipo	Descrição
instanceIds	StringList	Os nós que foram seleciona dos pelo comando. **Note** Os IDs de instância só serão incluídos na mensagem de resumo se a tarefa do Run Command direcionar IDs de instância diretamen te. Os IDs de instância não serão incluídos na mensagem de resumo se a tarefa Run Command tiver sido emitida usando o direcionamento com base em tags.
status	String	Status do comando para o comando.

Notificações do Amazon SNS com base em invocação

Se você enviar um comando para vários nós, o Amazon SNS poderá enviar mensagens sobre cada cópia ou invocação desse comando. As mensagens incluem as seguintes informações.

Campo	Tipo	Descrição
eventTime	String	A hora em que o evento foi iniciado. O carimbo de hora é importante, pois o Amazon

Campo	Tipo	Descrição
		SNS não garante a ordem de entrega das mensagens. Exemplo: 2016-04-26T13:15:30Z
documentName	String	O nome do documento SSM usado para executar esse comando.
requestedDateTime	String	A data e a hora em que a solicitação foi enviada para esse nó específico.
commandId	String	O ID gerado pelo Run Command após o envio do comando.
instanceId	String	A instância que foi direcionada pelo comando.
status	String	Status do comando para essa invocação.

Para configurar notificações do Amazon SNS quando um comando mudar de status, conclua as tarefas a seguir.

> **Note**
>
> Se você não estiver configurando notificações do Amazon SNS para sua janela de manutenção, poderá ignorar a Tarefa 5 apresentada posteriormente neste tópico.

Tópicos

- Tarefa 1: Criar e assinar um tópico do Amazon SNS
- Tarefa 2: Criar uma política do IAM para notificações do Amazon SNS
- Tarefa 3: Criar uma função do IAM para notificações do Amazon SNS

- Tarefa 4: Configurar o acesso do usuário
- Tarefa 5: Anexar a política iam:PassRole à função da janela de manutenção

Tarefa 1: Criar e assinar um tópico do Amazon SNS

Um tópico do Amazon SNS é um canal de comunicação que o Run Command e as tarefas do Run Command registradas em uma janela de manutenção usam para enviar notificações sobre o status dos comandos. O Amazon SNS é compatível com diferentes protocolos de comunicação, incluindo HTTP/S, e-mail e outros Serviços da AWS, como o Amazon Simple Queue Service (Amazon SQS). Para começar rapidamente, recomendamos que você comece com o protocolo de e-mail. Para obter informações sobre como criar um tópico, consulte Criar um tópico do Amazon SNS no Guia do desenvolvedor do Amazon Simple Notification Service.

> **Note**
>
> Depois de criar o tópico, copie ou anote o ARN do tópico. Você especifica esse ARN ao enviar um comando que esteja configurado para retornar notificações de status.

Após criar o tópico, inscreva-se nele especificando um Endpoint. Se você selecionou o protocolo de e-mail, o endpoint é o endereço de e-mail no qual você deseja receber notificações. Para obter mais informações sobre como se inscrever em um tópico, consulte Inscrever-se em um tópico do Amazon SNS no Guia do desenvolvedor do Amazon Simple Notification Service.

O Amazon SNS envia um e-mail de confirmação de Notificações AWS para o endereço de e-mail que você especificar. Abra o e-mail e selecione o link Confirm subscription (Confirmar assinatura).

Você receberá uma mensagem de confirmação da AWS. O Amazon SNS agora está configurado para receber notificações e enviar a notificação como um e-mail para o endereço de e-mail que você especificou.

Tarefa 2: Criar uma política do IAM para notificações do Amazon SNS

Use o procedimento a seguir para criar uma política personalizada do AWS Identity and Access Management (IAM) que forneça permissões para acionar notificações do Amazon SNS.

Para criar uma política do IAM personalizada SNS para notificações do Amazon SNS

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.

2. No painel de navegação, escolha Policies e, em seguida, Create Policy. (Se aparecer um botão Get Started, selecione-o e, em seguida, clique em Create Policy).
3. Selecione a guia JSON.
4. Substitua o conteúdo padrão por um dos seguintes, dependendo se o tópico do Amazon SNS usa a criptografia do AWS KMS:

   SNS topic not encrypted

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sns:Publish"
            ],
            "Resource": "arn:aws:sns:us-east-1:111122223333:sns-topic-
name"
        }
    ]
}
```

   A *região* representa o identificador da região para uma região da Região da AWS compatível com o AWS Systems Manager, como `us-east-2` para a região Leste dos EUA (Ohio). Para ver uma lista dos valores de *região* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

   account-id representa o identificador de 12 dígitos para a Conta da AWS, no formato `123456789012`.

   sns-topic-name representa o nome do tópico do Amazon SNS que você deseja usar para publicar notificações.

SNS topic encrypted

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sns:Publish"
            ],
            "Resource": "arn:aws:sns:us-east-1:111122223333:sns-topic-
name"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/kms-key-
id"
        }
    ]
}
```

A *região* representa o identificador da região para uma região da Região da AWS compatível com o AWS Systems Manager, como `us-east-2` para a região Leste dos EUA (Ohio). Para ver uma lista dos valores de *região* com suporte, consulte a coluna Region em Systems Manager service endpoints no Referência geral da Amazon Web Services.

account-id representa o identificador de 12 dígitos para a Conta da AWS, no formato `123456789012`.

sns-topic-name representa o nome do tópico do Amazon SNS que você deseja usar para publicar notificações.

kms-key-id representa o ID da chave do KMS de criptografia simétrica no AWS KMS a ser usada para criptografar e descriptografar o tópico, no formato `1234abcd-12ab-34cd-56ef-12345EXAMPLE`.

> **Note**
>
> Há uma taxa pelo uso da criptografia do AWS KMS. Para obter mais informações, consulte Managing Amazon SNS encryption keys and costs no Guia do desenvolvedor do AWS Key Management Service.

5. Escolha Próximo: tags.
6. (Opcional) Adicione um ou mais pares de chave-valor de etiqueta para organizar, monitorar ou controlar acesso para essa política.
7. Escolha Próximo: revisar.
8. Na página Revisar política, em Nome, digite um nome para a política em linha. Por exemplo: **`my-sns-publish-permissions`**.
9. (Opcional) Em Descrição, digite uma descrição para a política.
10. Escolha Criar política.

Tarefa 3: Criar uma função do IAM para notificações do Amazon SNS

Use o procedimento a seguir para criar uma função de serviço do IAM para notificações do Amazon SNS. Essa função de serviço é usada pelo Systems Manager para iniciar notificações do Amazon SNS. Em todos os procedimentos subsequentes, essa função é chamada de função do IAM do Amazon SNS.

Para criar uma função de serviço do IAM para notificações do Amazon SNS

1. Faça login no AWS Management Console e abra o console do IAM, em https://console.aws.amazon.com/iam/.
2. No painel de navegação do console do IAM, escolha Perfis e, em seguida, Criar perfil.
3. Escolha o tipo de perfil de AWS service (Serviço da AWS) e, em seguida, selecione o Systems Manager.
4. Escolha o caso de uso do Systems Manager. Em seguida, escolha Próximo.

5. Na página Attach permissions policies (Anexar políticas de permissões), marque a caixa à esquerda do nome da política personalizada criada na Tarefa 2. Por exemplo: **`my-sns-publish-permissions`**.
6. (Opcional) Defina um limite de permissões. Esse é um atributo avançado que está disponível para perfis de serviço, mas não para perfis vinculados ao serviço.

   Expanda a seção Limite de permissões e escolha Usar um limite de permissões para controlar o número máximo de permissões de funções. O IAM inclui uma lista das políticas gerenciadas pela AWS e pelo cliente em sua conta. Selecione a política a ser usada para o limite de permissões ou escolha Criar política para abrir uma nova guia no navegador e criar uma nova política a partir do zero. Para obter mais informações, consulte Criar políticas do IAM no Guia do usuário do IAM. Depois de criar a política, feche essa guia e retorne à guia original para selecionar a política a ser usada para o limite de permissões.
7. Escolha Próximo.
8. Se possível, insira um nome de função ou sufixo de nome de função para ajudar a identificar o propósito desta função. Os nomes de função devem ser exclusivos em sua Conta da AWS. Eles não são diferenciados por letras maiúsculas e minúsculas. Por exemplo, não é possível criar perfis denominados **`PRODROLE`** e **`prodrole`**. Como várias entidades podem fazer referência à função, não é possível editar o nome da função depois que ela é criada.
9. (Opcional) Em Descrição da função, insira uma descrição para a nova função.
10. Selecione Editar nas seções Etapa 1: selecionar entidades confiáveis ou Etapa 2: selecionar permissões para editar os casos de uso e as permissões para a função.
11. (Opcional) Adicione metadados ao usuário anexando tags como pares de chave-valor. Para obter mais informações sobre o uso de tags no IAM, consulte Marcar recursos do IAM no Guia do usuário do IAM.
12. Revise a função e escolha Criar perfil.
13. Escolha o nome da função e depois copie ou anote o valor do Role ARN (ARN da função). Esse nome do recurso da Amazon (ARN) para a função é usado quando você envia um comando que foi configurado para retornar notificações do Amazon SNS.
14. A página Summary (Resumo) é aberta.

Tarefa 4: Configurar o acesso do usuário

Se uma entidade do IAM (usuário, perfil ou grupo) receber permissões de administrador, o usuário ou o perfil terá acesso a Run Command e Maintenance Windows, que são ferramentas do AWS Systems Manager.

Para entidades que não têm permissões de administrador, um administrador deve conceder as permissões a seguir à entidade do IAM:

- A política gerenciada `AmazonSSMFullAccess` ou uma política que forneça permissões comparáveis.
- As permissões `iam:PassRole` para o perfil criado em Tarefa 3: Criar uma função do IAM para notificações do Amazon SNS. Por exemplo:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/sns-role-name",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ssm.amazonaws.com"
                }
            }
        }
    ]
}
```

Para conceder acesso, adicione as permissões aos seus usuários, grupos ou perfis:

- Usuários e grupos no AWS IAM Identity Center:

  Crie um conjunto de permissões. Siga as instruções em Criação de um conjunto de permissões no Guia do usuário do AWS IAM Identity Center.
- Usuários gerenciados no IAM com provedor de identidades:

Crie um perfil para a federação de identidades. Siga as instruções em Criando um perfil para um provedor de identidades de terceiros (federação) no Guia do Usuário do IAM.

- Usuários do IAM:
  - Crie um perfil que seu usuário possa assumir. Siga as instruções em Criação de um perfil para um usuário do IAM no Guia do usuário do IAM.
  - (Não recomendado) Vincule uma política diretamente a um usuário ou adicione um usuário a um grupo de usuários. Siga as instruções em Adição de permissões a um usuário (console) no Guia do usuário do IAM.

Para configurar acesso de usuário e anexar a política **`iam:PassRole`** a uma conta de usuário

1. No painel de navegação do IAM, escolha Users (Usuários) e selecione a conta de usuário que deseja configurar.
2. Na guia Permissions (Permissões), na lista de políticas, verifique se a política **`AmazonSSMFullAccess`** está listada ou se existe uma política comparável que conceda à conta permissão para acessar o Systems Manager.
3. Escolha Add inline policy (Adicionar política em linha).
4. Na página Create policy (Criar política), selecione a guia Visual editor (Editor visual).
5. Selecione Choose a service (Escolher um serviço) e, em seguida, IAM.
6. Para Actions (Ações), na caixa de texto Filter actions (Filtrar ações), insira **`PassRole`** e selecione a caixa de verificação ao lado de PassRole.
7. Em Resources (Recursos), verifique se Specific (Específico) está selecionado e, em seguida, selecione Add ARN (Adicionar ARN).
8. No campo Specify ARN for role (Especificar ARN para função), cole o ARN da função do IAM do Amazon SNS que você copiou no final da Tarefa 3. O sistema preenche automaticamente os campos Account (Conta) e Role name with path (Nome de função com caminho).
9. Escolha Add (Adicionar).
10. Escolha Review policy (Revisar política).
11. Na página Review Policy (Revisar política), insira um nome e depois escolha Create Policy (Criar política).

Tarefa 5: Anexar a política iam:PassRole à função da janela de manutenção

Ao registrar uma tarefa do Run Command em uma janela de manutenção, você especifica o nome de recurso da Amazon (ARN) de uma função de serviço. Essa função de serviço é usada pelo Systems Manager para executar tarefas registradas para a janela de manutenção. Para configurar notificações do Amazon SNS para uma tarefa registrada do Run Command, anexe uma política `iam:PassRole` à função de serviço da janela de manutenção especificada. Se você não pretende configurar a tarefa registrada para notificações do Amazon SNS, essa tarefa pode ser ignorada.

A política `iam:PassRole` permite que a função de serviço do Maintenance Windows transmita a função do IAM do Amazon SNS criada na Tarefa 3 ao serviço Amazon SNS. O procedimento a seguir mostra como anexar a política `iam:PassRole` à função de serviço do Maintenance Windows.

> **Note**
>
> Use uma função de serviço personalizada para sua janela de manutenção para enviar notificações relacionadas às tarefas do Run Command registradas. Para mais informações, consulte Configurar o Maintenance Windows.
> Se for necessário criar um perfil de serviço personalizado para tarefas de janela de manutenção, consulte Configurar o Maintenance Windows.

Para anexar sua política **`iam:PassRole`** à função do Maintenance Windows

1. Abra o console do IAM, em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Roles (Funções) e selecione a função do IAM do Amazon SNS criada na Tarefa 3.
3. Copie ou anote o Role ARN (ARN da função) e retorne à seção Roles (Funções) do console do IAM.
4. Selecione a função de serviço personalizada Maintenance Windows que você criou na lista Role name (Nome da função).
5. Na guia Permissions (Permissões), verifique se a política `AmazonSSMMaintenanceWindowRole` está listada ou se existe uma política comparável que dê permissão de janelas de manutenção à API do Systems Manager. Se não estiver, escolha Adicionar permissões, Anexar políticas para anexá-la.
6. Escolha Add permissions, Create inline policy (Adicionar permissões, Criar política em linha).
7. Selecione a guia Visual Editor (Editor visual).

8. Para Service (Serviço), selecione IAM.
9. Para Actions (Ações), na caixa de texto Filter actions (Filtrar ações), insira **`PassRole`** e selecione a caixa de verificação ao lado de PassRole.
10. Em Resources (Recursos), escolha Specific (Específico) e Add ARN (Adicionar ARN).
11. Na caixa Specify ARN for role (Especificar ARN para função), cole o ARN da função do IAM do Amazon SNS criada na Tarefa 3 e escolha Add (Adicionar).
12. Selecione Revisar política.
13. Na página Revisar política, especifique um nome para a política `PassRole` e, em seguida, escolha Criar política.

Exemplo de notificações do Amazon SNS para AWS Systems Manager

É possível configurar o Amazon Simple Notification Service (Amazon SNS) para enviar notificações sobre o status dos comandos que você envia usando o Run Command ou Maintenance Windows, ambos ferramentas do AWS Systems Manager.

> **Note**
>
> Este guia não aborda como configurar notificações usando o Run Command ou a Maintenance Windows. Para obter informações sobre como configurar o Run Command ou o Maintenance Windows para enviar notificações do Amazon SNS sobre o status dos comandos, consulte Configurar notificações do Amazon SNS para o AWS Systems Manager.

Os exemplos a seguir mostram a estrutura da saída JSON retornada por notificações do Amazon SNS, quando estas estão configuradas para o Run Command ou o Maintenance Windows.

Exemplo de saída JSON para mensagens de resumo de comandos usando o direcionamento de IDs de instância

```
{
    "commandId": "a8c7e76f-15f1-4c33-9052-0123456789ab",
    "documentName": "AWS-RunPowerShellScript",
    "instanceIds": [
        "i-1234567890abcdef0",
        "i-9876543210abcdef0"
    ],
```

```
    "requestedDateTime": "2019-04-25T17:57:09.17Z",
    "expiresAfter": "2019-04-25T19:07:09.17Z",
    "outputS3BucketName": "amzn-s3-demo-bucket",
    "outputS3KeyPrefix": "runcommand",
    "status": "InProgress",
    "eventTime": "2019-04-25T17:57:09.236Z"
}
```

Exemplo de saída JSON para mensagens de resumo de comandos usando o direcionamento com base em tags

```
{
    "commandId": "9e92c686-ddc7-4827-b040-0123456789ab",
    "documentName": "AWS-RunPowerShellScript",
    "instanceIds": [],
    "requestedDateTime": "2019-04-25T18:01:03.888Z",
    "expiresAfter": "2019-04-25T19:11:03.888Z",
    "outputS3BucketName": "",
    "outputS3KeyPrefix": "",
    "status": "InProgress",
    "eventTime": "2019-04-25T18:01:05.825Z"
}
```

Exemplo de saída JSON para mensagens de invocação

```
{
    "commandId": "ceb96b84-16aa-4540-91e3-925a9a278b8c",
    "documentName": "AWS-RunPowerShellScript",
    "instanceId": "i-1234567890abcdef0",
    "requestedDateTime": "2019-04-25T18:06:05.032Z",
    "status": "InProgress",
    "eventTime": "2019-04-25T18:06:05.099Z"
}
```

Usar o Run Command para enviar um comando que retorna notificações de status

Os procedimentos a seguir mostram como usar a AWS Command Line Interface (AWS CLI) ou o console do AWS Systems Manager para enviar um comando por meio do Run Command, uma ferramenta do AWS Systems Manager, que é configurado para retornar notificações de status.

Enviar um Run Command que retorna notificações (console)

Use o procedimento a seguir para enviar um comando por meio do Run Command que esteja configurado para retornar notificações de status usando o console do Systems Manager.

Como enviar um comando que retorne notificações (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Run Command.
3. Selecione Run command.
4. Na lista Command document (Documento de comando), escolha um documento do Systems Manager.
5. Na seção Command parameters, especifique valores para os parâmetros necessários.
6. Na seção Targets (Destinos), escolha os nós gerenciados nos quais você quer executar essa operação, especificando as etiquetas, selecionando as instâncias ou dispositivos de borda manualmente ou especificando um grupo de recursos.

   > **Tip**
   >
   > Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

7. Para Other parameters (Outros parâmetros):
   - Em Comment (Comentário), digite as informações sobre esse comando.
   - Em Timeout (seconds) (Tempo limite [segundos]), especifique o número de segundos para o sistema aguardar até a falha de execução do comando total.
8. Para Rate control (Controle de taxa):
   - Em Concurrency (Concorrência), especifique um número ou uma porcentagem de nós gerenciados nos quais executar o comando ao mesmo tempo.

     > **Note**
     >
     > Se você selecionou destinos especificando tags aplicadas a instâncias a nós gerenciados ou especificando grupos de recursos da AWS, e não tiver certeza de

quantas instâncias são direcionadas, restrinja o número de instâncias que poderão executar o documento ao mesmo tempo, especificando uma porcentagem.

- Em Error threshold (Limite de erro), especifique quando parar de executar o comando em outros nós depois de falhar em alguns ou em uma porcentagem de nós. Por exemplo, se você especificar três erros, o Systems Manager deixará de enviar o comando quando o 4° erro for recebido. Os nós gerenciados que continuam processando o comando também podem enviar erros.

9. (Opcional) Em Output options (Opções de saída), para salvar a saída do comando em um arquivo, selecione a caixa Write command output to an S3 bucket (Gravar saída do comando em um bucket do S3). Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

> **Note**
>
> As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil de instância (para instâncias do EC2) ou perfil de serviço do IAM (máquinas ativadas para ambientes híbridos) atribuído à instância, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço do IAM associado ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

10. Na seção SNS Notifications, escolha Enable SNS notifications.
11. Em IAM role (Função do IAM), escolha o ARN da função do IAM do Amazon SNS que você criou na Tarefa 3 em Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS.
12. Para SNS topic (Tópico do SNS), insira o ARN do tópico do Amazon SNS a ser usado.
13. Em Event notifications (Notificações de eventos), escolha os eventos para os quais deseja receber notificações.
14. Em Change notifications (Alterar notificações), escolha receber notificações apenas para o resumo do comando (Command status changes) (Alterações de status do comando) ou para cada cópia de um comando enviado a vários nós (Command status on each instance changes) (O status do comando em cada instância é alterado).
15. Escolha Executar.

16. Verifique se há uma mensagem do Amazon SNS em seu e-mail e abra o e-mail. O Amazon SNS pode levar alguns minutos para enviar a mensagem de e-mail.

Enviar um Run Command que retorna notificações (CLI)

Use o procedimento a seguir para enviar um comando por meio do Run Command que esteja configurado para retornar notificações de status usando a AWS CLI.

Para enviar um comando que retorne notificações (CLI)

1. Abra a AWS CLI.
2. Especifique parâmetros no comando a seguir para definir destinos com base nos IDs dos nós gerenciados.

```
aws ssm send-command --instance-ids "ID-1, ID-2" --document-name "Name"
 --parameters '{"commands":["input"]}' --service-role "SNSRoleARN" --
notification-config '{"NotificationArn":"SNSTopicName","NotificationEvents":
["All"],"NotificationType":"Command"}'
```

Veja um exemplo a seguir.

```
aws ssm send-command --instance-ids "i-02573cafcfEXAMPLE, i-0471e04240EXAMPLE"
 --document-name "AWS-RunPowerShellScript" --parameters '{"commands":
["Get-Process"]}' --service-role "arn:aws:iam::111122223333:role/
SNS_Role" --notification-config '{"NotificationArn":"arn:aws:sns:us-
east-1:111122223333:SNSTopic","NotificationEvents":
["All"],"NotificationType":"Command"}'
```

Comandos alternativos

Especifique parâmetros no comando a seguir para direcionar instâncias gerenciadas usando tags.

```
aws ssm send-command --targets "Key=tag:TagName,Values=TagKey" --document-name
 "Name" --parameters '{"commands":["input"]}' --service-role "SNSRoleARN" --
notification-config '{"NotificationArn":"SNSTopicName","NotificationEvents":
["All"],"NotificationType":"Command"}'
```

Veja um exemplo a seguir.

```
aws ssm send-command --targets "Key=tag:Environment,Values=Dev" --
document-name "AWS-RunPowerShellScript" --parameters '{"commands":
["Get-Process"]}' --service-role "arn:aws:iam::111122223333:role/
SNS_Role" --notification-config '{"NotificationArn":"arn:aws:sns:us-
east-1:111122223333:SNSTopic","NotificationEvents":
["All"],"NotificationType":"Command"}'
```

3. Pressione Enter.
4. Verifique se há uma mensagem do Amazon SNS em seu e-mail e abra o e-mail. O Amazon SNS pode levar alguns minutos para enviar a mensagem de e-mail.

Para obter mais informações, consulte send-command na Referência de comandos da AWS CLI.

Usar uma janela de manutenção para enviar um comando que retorna notificações de status

Os procedimentos a seguir mostram como registrar uma tarefa do Run Command com sua janela de manutenção usando o console do AWS Systems Manager ou a AWS Command Line Interface (AWS CLI). O Run Command é uma ferramenta do AWS Systems Manager. Os procedimentos também descrevem como configurar a tarefa do Run Command para retornar notificações de status.

Antes de começar

Se você não tiver criado uma janela de manutenção ou destinos registrados, consulte Crie e gerencie janelas de manutenção usando o console para obter instruções sobre como criar uma janela de manutenção e registrar destinos.

Para receber notificações do serviço Amazon Simple Notification Service (Amazon SNS), anexe uma política `iam:PassRole` à função de serviço do Maintenance Windows especificada na tarefa registrada. Se você não tiver adicionado as permissões `iam:PassRole` à sua função de serviço do Maintenance Windows, consulte Tarefa 5: Anexar a política iam:PassRole à função da janela de manutenção.

Registrar uma tarefa do Run Command em uma janela de manutenção que retorna notificações (console)

Use o procedimento a seguir para registrar uma tarefa do Run Command que esteja configurada para retornar notificações de status à janela de manutenção usando o console do Systems Manager.

Como registrar uma tarefa do Run Command na janela de manutenção que retorna notificações (console)

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Maintenance Windows.
3. Selecione a janela de manutenção para a qual você deseja registrar uma tarefa do Run Command configurada para enviar notificações do Amazon Simple Notification Service (Amazon SNS).
4. Selecione Actions (Ações) e depois Register run command task (Registrar tarefa de comando de execução).
5. No campo Name (Nome), insira um nome para a tarefa.
6. (Opcional) No campo Description (Descrição), insira uma descrição.
7. Em Command document (Documento do comando), escolha um documento do comando.
8. Em Task priority (Prioridade da tarefa), especifique uma prioridade para essa tarefa. Zero (`0`) é a prioridade mais alta. As tarefas em uma janela de manutenção são agendadas em ordem de prioridade. Tarefas que têm a mesma prioridade estão agendadas em paralelo.
9. Na seção Targets (Destinos), selecione um grupo de destino registrado ou selecione destinos não registrados.
10. Para Rate control (Controle de taxa):

    - Em Concurrency (Concorrência), especifique um número ou uma porcentagem de nós gerenciados nos quais executar o comando ao mesmo tempo.

      > **Note**
      >
      > Se você selecionou destinos especificando tags aplicadas a instâncias a nós gerenciados ou especificando grupos de recursos da AWS, e não tiver certeza de quantas instâncias são direcionadas, restrinja o número de instâncias que poderão executar o documento ao mesmo tempo, especificando uma porcentagem.

    - Em Error threshold (Limite de erro), especifique quando parar de executar o comando em outros nós depois de falhar em alguns ou em uma porcentagem de nós. Por exemplo, se você especificar três erros, o Systems Manager deixará de enviar o comando quando o 4° erro for recebido. Os nós gerenciados que continuam processando o comando também podem enviar erros.

11. Na área IAM service role (Função de serviço do IAM), escolha a função de serviço do Maintenance Windows que tenha permissões `iam:PassRole` para a função do SNS.

> **Note**
>
> Adicione permissões do `iam:PassRole` a função Maintenance Windows para permitir que o Systems Manager passe a função do SNS ao Amazon SNS. Se você não tiver adicionado permissões `iam:PassRole`, consulte a Tarefa 5, no tópico Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS.

12. (Opcional) Em Opções de saída, para salvar a saída de comando em um arquivo, selecione a caixa Habilitar a gravação da saída no S3. Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

> **Note**
>
> As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil da instância atribuído ao nó gerenciado, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço IAM associada ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

13. Na seção SNS notifications (Notificações do SNS), faça o seguinte:

- Selecione Enable SNS Notifications (Habilitar notificações do SNS).
- Para IAM role (Função do IAM), escolha o nome do recurso da Amazon (ARN) da função do IAM do Amazon SNS que você criou na Tarefa 3 em Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS para iniciar o Amazon SNS.
- Para SNS topic (Tópico do SNS), insira o ARN do tópico do Amazon SNS a ser usado.
- Em Event type (Tipo de evento) escolha os eventos para os quais deseja receber notificações.
- Em Notification type (Tipo de notificação), escolha receber notificações para cada cópia de um comando enviado a vários nós (invocações) ou o resumo de comandos.

14. Na seção Parameters (Parâmetros), insira os parâmetros necessários com base no documento do Command que você escolheu.

15. Selecione Register Run command task (Registrar tarefa do Run Command).
16. Depois da próxima vez que a janela de manutenção for executada, verifique se recebeu um e-mail do Amazon SNS e abra a mensagem de e-mail. O Amazon SNS pode levar alguns minutos para enviar o e-mail.

Registrar uma tarefa do Run Command em uma janela de manutenção que retorna notificações (CLI)

Use o procedimento a seguir para registrar uma tarefa do Run Command que esteja configurada para retornar notificações de status à janela de manutenção usando a AWS CLI.

Para registrar uma tarefa do Run Command com a janela de manutenção que retorna notificações (CLI)

> **Note**
>
> Para gerenciar melhor suas opções de tarefas, esse procedimento usa a opção de comando `--cli-input-json`, com valores de opção armazenados em um arquivo JSON.

1. Na sua máquina local, crie um arquivo chamado `RunCommandTask.json`.
2. Cole o conteúdo a seguir no arquivo .

```
{
    "Name": "Name",
    "Description": "Description",
    "WindowId": "mw-0c50858d01EXAMPLE",
    "ServiceRoleArn": "arn:aws:iam::account-id:role/MaintenanceWindowIAMRole",
    "MaxConcurrency": "1",
    "MaxErrors": "1",
    "Priority": 3,
    "Targets": [
        {
            "Key": "WindowTargetIds",
            "Values": [
                "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
            ]
        }
    ],
    "TaskType": "RUN_COMMAND",
```

```
    "TaskArn": "CommandDocumentName",
    "TaskInvocationParameters": {
        "RunCommand": {
            "Comment": "Comment",
            "TimeoutSeconds": 3600,
            "NotificationConfig": {
                "NotificationArn": "arn:aws:sns:region:account-id:SNSTopicName",
                "NotificationEvents": [
                    "All"
                ],
                "NotificationType": "Command"
            },
            "ServiceRoleArn": "arn:aws:iam::account-id:role/SNSIAMRole"
        }
    }
}
```

3. Substitua os valores de exemplo por informações sobre seus próprios recursos.

   Você também pode restaurar opções omitidas deste exemplo se quiser usá-las. Por exemplo, você pode salvar a saída do comando em um bucket do S3.

   Para obter mais informações, consulte register-task-with-maintenance-window na Referência de comandos da AWS CLI.

4. Salve o arquivo.

5. No diretório da sua máquina local em que você salvou o arquivo, execute o comando a seguir.

```
aws ssm register-task-with-maintenance-window --cli-input-json file://
RunCommandTask.json
```

> **Important**
>
> Não se esqueça de incluir `file://` antes do nome de arquivo. Ele é obrigatório nesse comando.

Se houver êxito, o comando retornará informações semelhantes às mostradas a seguir.

```
{
    "WindowTaskId": "j2l8d5b5c-mw66-tk4d-r3g9-1d4d1EXAMPLE"
```

```
}
```

6. Após a próxima execução da janela de manutenção, verifique se recebeu um e-mail do Amazon SNS e abra a mensagem do e-mail. O Amazon SNS pode levar alguns minutos para enviar o e-mail.

Para obter mais informações sobre como registrar tarefas para uma janela de manutenção na linha de comando, consulte Registrar tarefas com a janela de manutenção.

Integrações de produtos e serviços com o Systems Manager

Por padrão, o AWS Systems Manager é integrado a vários Serviços da AWS e a outros produtos e serviços. A informações a seguir podem ajudar você a configurar o Systems Manager para integrar-se aos produtos e os serviços que você usa.

- Integração com Serviços da AWS
- Integração com outros produtos e serviços

Integração com Serviços da AWS

Por meio do uso de documentos do Systems Manager Command (documentos SSM) e de runbooks do Automation, você pode usar o AWS Systems Manager para fazer uma integração com Serviços da AWS. Para obter mais informações sobre esses recursos, consulte Documentos do AWS Systems Manager.

O Systems Manager está integrado aos Serviços da AWS.

Computação

Amazon Elastic Compute Cloud (Amazon EC2)	O Amazon EC2 fornece capacidade de computação escalável na Nuvem AWS. O uso do Amazon EC2 elimina a necessidade de investir em hardware inicialmente, portanto, você pode desenvolver e implantar aplicativos com mais rapidez. É possível usar o Amazon EC2 para executar quantos servidores virtuais forem necessários, configurar a segurança e as redes e gerenciar o armazenamento. O Systems Manager permite que você execute várias tarefas nas instâncias do EC2. Por exemplo, você pode executar, configurar, gerenciar, manter, solucionar problemas e conectar-se com segurança às instâncias do EC2. Você também pode usar o Systems

	Manager para implantar software, determinar o status da conformidade e coletar inventário de suas instâncias do EC2. Saiba mais • Trabalhar com nós gerenciados • AWS Systems Manager State Manager • AWS Systems Manager Run Command • AWS Systems Manager Patch Manager • AWS Systems Manager Session Manager • AWS Systems Manager Distributor • AWS Systems Manager Compliance • AWS Systems Manager Inventory
Amazon EC2 Auto Scaling	O Auto Scaling ajuda a garantir que você tenha o número correto de instâncias do EC2 disponíveis para processar a carga da sua aplicação. Você cria coleções de instâncias EC2, chamadas de grupos de Auto Scaling. O Systems Manager permite que você automatize procedimentos comuns, como aplicar patches na Amazon Machine Image (AMI), usados no modelo do Auto Scaling para o grupo do Auto Scaling. Saiba mais Atualização de AMIs para grupos do Auto Scaling

Amazon Elastic Container Service (Amazon ECS)	O Amazon ECS é um serviço de gerenciamento de contêineres altamente escalável e rápido que permite a execução, a interrupção e o gerenciamento de contêineres do Docker em um cluster. O Systems Manager permite a você gerenciar instâncias de contêineres remotamente e injete dados confidenciais em seus contêineres armazenando seus dados confidenciais em parâmetros do Parameter Store, uma ferramenta do Systems Manager, fazendo referência a eles na definição de contêiner. Saiba mais • Gerencie remotamente as instâncias de contêiner usando o AWS Systems Manager • Especificar dados sigilosos usando o Systems Manager Parameter Store

AWS Lambda	O Lambda é um serviço de computação que permite que você execute o código sem provisionar ou gerenciar servidores. O Lambda executa o código somente quando necessári o e dimensiona automaticamente, desde algumas solicitações por dia até milhares por segundo. O Systems Manager permite que você use funções do Lambda no conteúdo do runbook do Automation, usando a ação `aws:invok eLambdaFunction` . Para usar parâmetros do Parameter Store em funções do AWS Lambda, você pode usar a extensão do Lambda para parâmetros e segredos da AWS para recuperar os valores dos parâmetros e armazená-los em cache para uso futuro. Saiba mais Atualize uma AMI dourada usando Automation, AWS Lambda e Parameter Store Usar parâmetros do Parameter Store em funções do AWS Lambda

Internet das Coisas (IoT)

Dispositivos principais do AWS IoT Greengrass	AWS IoT Greengrass é um serviço de nuvem e runtime de borda da IoT de código aberto que ajuda você a criar, implantar e gerenciar aplicações de IoT em seus dispositivos. O Systems Manager oferece suporte nativo para os dispositivos principais do AWS IoT Greengrass.

	Saiba mais Gerenciar dispositivos de borda com o Systems Manager
Dispositivos principais do AWS IoT	A AWS IoT fornece os serviços de nuvem que conectam seus dispositivos IoT a outros dispositivos e aos serviços de nuvem da AWS. A AWS IoT fornece software para dispositivos que pode ajudar você a integrar seus dispositivos IoT a soluções baseadas em AWS IoT. Se seus dispositivos puderem se conectar ao AWS IoT, o AWS IoT poderá conectá-los aos serviços em nuvem que a AWS fornece. O Systems Manager é compatível com os dispositivos principais de AWS IoT, desde que esses dispositivos estejam configurados como nós gerenciados em um ambiente híbrido e multinuvem. Saiba mais Gerenciar nós em ambientes híbridos e multinuvem com o Systems Manager

Armazenamento

Amazon Simple Storage Service (Amazon S3)	O Amazon S3 é o armazenamento para a Internet. Ele foi projetado para facilitar a computação de escala na Web para os desenvolvedores. O Amazon S3 tem uma interface de serviços da web simples que você pode usar para armazenar e recuperar qualquer quantidade de dados, a qualquer momento, em qualquer lugar da web.

	O Systems Manager permite que você execute scripts remotos e documentos do SSM armazenados no Amazon S3. O Distributor, uma ferramenta do AWS Systems Manager, usa o Amazon S3 para armazenar pacotes. Também é possível enviar resultados para o Amazon S3 para o Run Command e o Session Manager, ferramentas do AWS Systems Manager. Saiba mais • Executar scripts no Amazon S3 • Executar documentos do em locais remotos • AWS Systems Manager Distributor • Registrar dados da sessão em log usando o Amazon S3 (console)

Developer Tools

AWS CodeBuild	O CodeBuild é um serviço de compilação na nuvem, completamente gerenciado. O CodeBuild compila o código-fonte, executa testes de unidade e produz artefatos prontos para implantação. O CodeBuild elimina a necessidade de provisionar, gerenciar e escalar seus próprios servidores de compilaçã o. O Parameter Store permite que você armazene informações confidenciais para suas especific ações de compilação e projetos.

	Saiba mais • Crie a referência de especificação para o CodeBuild • Criar um projeto de compilação no AWS CodeBuild
AWS CDK	O AWS Cloud Development Kit (AWS CDK) é um framework para definir a infraestrutura de nuvem como código, com linguagens de programação, e implantá-la pelo AWS CloudFormation. Com o Application Manager, você pode visualizar suas estruturas de CDK agrupadas como aplicações, visualizar a estrutura da aplicação, inclusive os recursos subjacent es, visualizar alertas, investigar e corrigir problemas operacionais e acompanhar os custos no console do Application Manager. Saiba mais • Visualizar informações da visão geral sobre uma aplicação • Visualizar recursos da aplicação

Segurança, identidade e conformidade

AWS Identity and Access Management (IAM)	O IAM é um serviço da Web que ajuda você a controlar o acesso aos recursos da AWS de forma segura. Você usa o IAM para controlar quem é autenticado (fez login) e autorizado (tem permissões) a usar os recursos. O Systems Manager permite controlar o acesso aos serviços usando o IAM.

	Saiba mais • Como o AWS Systems Manager funciona com o IAM • Ações, recursos e chaves de condição para o AWS Systems Manager • Configurar permissões de instância obrigatórias para o Systems Manager
AWS Secrets Manager	Secrets Manager fornece um gerenciamento mais fácil de segredos. Os segredos podem ser credenciais de banco de dados, senhas, chaves de API de terceiros e até mesmo texto arbitrário. O Parameter Store permite recuperar segredos do Secrets Manager quando você utiliza outros serviços da Serviços da AWS que já oferecem suporte para referências a parâmetros do Parameter Store. Saiba mais Fazer referência a segredos do AWS Secrets Manager em parâmetros do Parameter Store

AWS Security Hub	O Security Hub oferece uma visão abrangente dos alertas de segurança de alta prioridade e do status de conformidade em todas as Contas da AWS. O Security Hub agrega, organiza e prioriza alertas de segurança ou descobertas de vários Serviços da AWS. Quando você ativa a integração entre o Security Hub e o Patch Manager, uma ferramenta do AWS Systems Manager, o Security Hub monitora o status de patches das suas frotas do ponto de vista da segurança. Os detalhes de conformidade de patches são exportados automaticamente para o Security Hub. Isso permite que você use uma única exibição para monitorar centralmente o status de conformidade de patches e rastrear outras descobertas de segurança. Você pode receber alertas quando os nós da sua frota saírem da conformidade de patches e revisar as descobertas de conformidade de patches no console do Security Hub. Você também pode integrar o Security Hub ao Explorer e OpsCenter, ferramentas do AWS Systems Manager. A integração com o Security Hub permite que você receba descobertas do Security Hub no Explorer e no OpsCenter . As descobertas do Security Hub fornecem informações de segurança que você pode usar no Explorer e OpsCenter para agregar e tomar medidas sobre problemas operacionais, de segurança e de performance no AWS Systems Manager.

	Será cobrada uma taxa para o uso do Security Hub. Para obter mais informações, consulte Preço do Security Hub. Saiba mais • Receber descobertas do AWS Security Hub no Explorer • Noções básicas sobre a integração do OpsCenter com o AWS Security Hub • Integrar o Patch Manager ao AWS Security Hub

Criptografia e PKI

AWS Key Management Service (AWS KMS)	O AWS KMS é um serviço gerenciado que permite a criação e o controle de chaves de criptografia gerenciadas usadas para criptografar seus dados. O Systems Manager permite que você use o AWS KMS para criar parâmetros `SecureString` e criptografar dados de sessões do Session Manager. Saiba mais • Criptografia do AWS KMS para parâmetros SecureString do AWS Systems Manager Parameter Store • Ativar a criptografia de chaves do KMS de dados de sessão (console)

Gerenciamento e governança

AWS CloudFormation	O AWS CloudFormation é um serviço que ajuda você a modelar e configurar os recursos da Amazon Web Services, para que passe menos tempo gerenciando esses recursos e mais tempo se concentrando nas aplicações executadas na AWS. O Parameter Store é uma fonte para referênci as dinâmicas. Referências dinâmicas fornecem uma maneira concisa e poderosa para você especificar valores externos armazenados e gerenciados em outros serviços nos modelos de pilha do AWS CloudFormation. Saiba mais Usar referências dinâmicas para especificar valores de modelos
AWS CloudTrail	O CloudTrail é um AWS service (Serviço da AWS) que ajuda você a autorizar a governanç a, a conformidade e as auditorias operacion ais e de risco na sua Conta da AWS. As ações realizadas por um usuário, uma função ou um AWS service (Serviço da AWS) são registrad as como eventos no CloudTrail. Os eventos incluem ações realizadas no AWS Managemen t Console, AWS Command Line Interface (AWS CLI) e nos SDKs e APIs da AWS. O Systems Manager se integra ao CloudTrai l, que captura a maioria das chamadas da API do Systems Manager como eventos. Isso inclui chamadas de API iniciadas pelo console do Systems Manager e chamadas feitas para as APIs do Systems Manager.

	Saiba mais Registrar em log chamadas de API do AWS Systems Manager com o AWS CloudTrail
Amazon CloudWatch Logs	O Amazon CloudWatch Logs permite centralizar os logs de todos os sistemas, aplicações e Serviços da AWS que você usa. Você pode visualizá-los, pesquisá-los em busca de códigos de erro ou padrões específicos, filtrá-los com base em campos específicos ou arquivá-los com segurança para análise futura. O Systems Manager suporta o envio de logs para o SSM Agent, Run Command e Session Manager para o CloudWatch Logs. Saiba mais • Enviar logs de nós para o CloudWatch Logs unificado (agente do CloudWatch) • Configurar o Amazon CloudWatch Logs para Run Command • Registrar dados da sessão em log usando o Amazon CloudWatch Logs (console)

Amazon EventBridge	O EventBridge oferece um fluxo quase em tempo quase real dos eventos do sistema que descrevem as alterações nos recursos da Amazon Web Services. Com regras simples que você pode configurar rapidamente, é possível corresponder eventos e roteá-los para um ou mais streams ou funções de destino. O EventBridge se torna ciente das alterações operacionais no momento em que elas ocorrem. O EventBridge responde a essas alterações operacionais e toma medidas corretivas conforme necessário. Essas ações incluem o envio de mensagens para responder ao ambiente, ativar funções e capturar informações sobre o estado. O Systems Manager tem vários eventos suportados pelo EventBridge, permitindo que você execute ações com base no conteúdo desses eventos. Saiba mais Monitorar eventos do Systems Manager com o Amazon EventBridge **Note** O Amazon EventBridge é a maneira preferencial de gerenciar seus eventos. O CloudWatch Events e o EventBrid ge são o mesmo serviço subjacent e e API, mas o EventBridge oferece mais recursos. Alterações feitas no CloudWatch ou no EventBridge são refletidas em cada console. Para obter

	mais informações, consulte o Guia do Usuário do Amazon EventBridge.
AWS Config	AWS Config: fornece uma visão detalhada da configuração dos recursos da AWS em sua Conta da AWS. Isso inclui como os recursos estão relacionados entre si e como eles foram configurados. Isso permite que você veja como as configurações e relacionamentos mudam ao longo do tempo. O Systems Manager é integrado ao AWS Config, fornecendo várias regras que ajudam você a obter visibilidade das instâncias do EC2. Essas regras ajudam a identificar quais instâncias do EC2 são gerenciadas pelo Systems Manager, configurações do sistema operacional, atualizações no nível do sistema, aplicações instaladas, configurações de rede e muito mais. Saiba mais • Tipos de recursos da AWS Config com suporte • Registrar a configuração de software para instâncias gerenciadas • Visualizar o histórico do inventário e o controle de alterações

AWS Trusted Advisor	O Trusted Advisor é uma ferramenta online que fornece orientação em tempo real para ajudar você a provisionar seus recursos seguindo as práticas recomendadas da AWS. O Systems Manager hospeda o Trusted Advisor e você pode visualizar os dados do Trusted Advisor no Explorer. Saiba mais • AWS Systems Manager Explorer • Conceitos básicos da AWS Trusted Advisor

AWS Organizations	O Organizations é um serviço de gerenciamento de contas que permite consolidar várias Contas da AWS em uma organização que você cria e gerencia de maneira centralizada. O Organizations inclui os recursos de faturamento consolidado e gerenciamento de contas, que permitem que você atenda melhor às necessidades orçamentárias, de segurança e de conformidade do seu negócio. A integração entre o Change Manager, uma ferramenta do AWS Systems Manager, com o Organizations, possibilita usar uma conta de administrador delegado para gerenciar solicitações de alteração, modelos de alteração e aprovações para toda a sua organização por meio dessa única conta. A integração do Organizations com o Inventory, uma ferramenta do AWS Systems Manager, e o Explorer permite agregar dados de inventário e de operações (OpsData) de várias Regiões da AWS e Contas da AWS. A integração entre o Quick Setup, uma ferramenta do AWS Systems Manager, e o Organizations automatiza tarefas comuns de configuração de serviço e implanta configurações de serviço com base nas práticas recomendadas em todas as unidades organizacionais (UOs).

Rede e entrega de conteúdo

AWS PrivateLink	Um AWS PrivateLink permite que você conecte de forma privada sua VPC aos Serviços da

	AWS compatíveis e aos serviços do endpoint da VPC, sem exigir um gateway da Internet, um dispositivo NAT, uma conexão VPN ou uma conexão do AWS Direct Connect. O Systems Manager é compatível com nós gerenciados conectados a APIs do Systems Manager que usam o AWS PrivateLink. Isso melhora o procedimento de segurança de seus nós gerenciados porque o AWS PrivateLink restringe todo o tráfego de rede entre esses nós, o Systems Manager e o Amazon EC2 para a rede da Amazon. Isso significa que os nós gerenciados não precisam ter acesso à Internet. Saiba mais Melhorar a segurança das instâncias do EC2 usando endpoints da VPC para o Systems Manager

Analytics

Amazon Athena	O Athena é um serviço de consultas interativ as que permite a análise de dados diretamente no Amazon Simple Storage Service (Amazon S3) usando SQL padrão. Com algumas ações no AWS Management Console, você pode direcionar o Athena para os dados armazenad os no Amazon S3 e começar a usar o SQL padrão para executar consultas ad hoc e obter resultados em segundos. O Systems Manager Inventory integra-se ao Amazon Athena para ajudar você a consultar dados de inventário de várias Regiões da AWS

	e Contas da AWS. A integração com o Athena usa a sincronização de dados dos recursos para que você possa visualizar os dados do inventário de todas os nós gerenciados na página Detailed View (Visualização detalhada) no console do Systems Manager Inventory. Saiba mais • Consultar dados de inventário de várias regiões e contas • Demonstração: usar a sincronização de dados de recursos para agregar dados do inventário
AWS Glue	O AWS Glue é um serviço de ETL (extração , transformação e carregamento) totalment e gerenciado que torna fácil e econômico categorizar os dados, limpá-los, aprimorá-los e movê-los de modo confiável entre vários armazenamentos e fluxos de dados. O Systems Manager usa o AWS Glue para rastrear os dados do Inventory no bucket do S3. Saiba mais Consultar dados de inventário de várias regiões e contas

Amazon QuickSight	O QuickSight é um serviço de analytics de negócios que você pode usar para criar visualizações, realizar uma análise única e obter insights comerciais dos seus dados. Ele pode detectar automaticamente fontes de dados da AWS, além de funcionar também com suas fontes de dados. A sincronização de dados de recursos do Systems Manager envia dados de inventári o, coletados de todos os nós gerenciados, para um único bucket do S3. Você pode usar o QuickSight para consultar e analisar os dados agregados. Saiba mais • Creating a resource data sync for Inventory • Demonstração: usar a sincronização de dados de recursos para agregar dados do inventário

Integração de aplicativo

Amazon Simple Notification Service (Amazon SNS)	O Amazon SNS é um serviço da Web que coordena e gerencia a entrega ou o envio de mensagens para endpoints ou clientes inscritos . O Systems Manager gera status para vários serviços que podem ser capturados pelas notificações do Amazon SNS.

	Saiba mais • Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS • Configurar notificações ou acionar ações com base nos eventos do Parameter Store

AWS Management Console

AWS Resource Groups	Grupos de recursos organizam seus recursos da AWS. Os grupos de recursos facilitam o gerenciamento, o monitoramento e a automatiz ação de tarefas em grandes números de recursos de uma vez. Os tipos de recursos do Systems Manager, como nós gerenciados, documentos SSM, janelas de manutenção, parâmetros do Parameter Store e listas de referência de patches podem ser adicionados aos grupos de recursos. Saiba mais O que é o AWS Resource Groups?

Tópicos

- Executar scripts no Amazon S3
- Fazer referência a segredos do AWS Secrets Manager em parâmetros do Parameter Store
- Criptografia do AWS KMS para parâmetros SecureString do AWS Systems Manager Parameter Store
- Use parâmetros do Parameter Store no Amazon Elastic Kubernetes Service.
- Usar parâmetros do Parameter Store em funções do AWS Lambda

Executar scripts no Amazon S3

Esta seção descreve como baixar e executar scripts do Amazon Simple Storage Service (Amazon S3). O tópico a seguir inclui informações e terminologia relacionadas ao Amazon S3. Para saber mais sobre o Amazon S3, consulte O que é o Amazon S3? Você pode executar tipos diferentes de script, inclusive scripts do Ansible Playbooks, Python, Ruby, Shell e PowerShell.

Você pode também fazer download de um diretório que inclua vários scripts. Ao executar o script principal no diretório, o AWS Systems Manager executa também qualquer script referenciado que estiver incluído no diretório.

Observe os detalhes essenciais a seguir sobre a execução de scripts do Amazon S3:

- O Systems Manager não verifica se o script pode ser executado em um nó. Antes de baixar e executar o script, verifique se o software necessário está instalado em seu nó. Ou você pode criar um documento composto que instala o software usando o Run Command ou o State Manager, ambos ferramentas do AWS Systems Manager, e, em seguida, baixa e executa o script.
- Verifique se seu usuário, perfil ou grupo recebeu as permissões do AWS Identity and Access Management (IAM) necessárias para realizar leitura do bucket do S3.
- Verifique se o perfil de instância nas instâncias do Amazon Elastic Compute Cloud (Amazon EC2) tem as permissões `s3:ListBucket` e `s3:GetObject`. Se o perfil de instância não tiver essas permissões, o sistema falhará ao fazer download do script do bucket do S3. Para obter mais informações, consulte Usar perfis de instâncias no Manual do usuário do IAM.

Execute scripts shell no Amazon S3

As informações a seguir incluem procedimentos para ajudar você a executar scripts do Amazon Simple Storage Service (Amazon S3) usando o console do AWS Systems Manager ou a AWS Command Line Interface (AWS CLI). Embora scripts shell sejam usados nos exemplos, outros tipos de scripts podem ser substituídos.

Execute um script shell no Amazon S3 (console)

Execute um script shell no Amazon S3

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Run Command.
3. Selecione Run command.

4. Na lista Command document (Documento do comando), escolha **`AWS-RunRemoteScript`**.
5. Em Command parameters, faça o seguinte:
   - Em Source Type, selecione S3.
   - Na caixa de texto Source Info (Informações da origem) digite as informações necessárias para acessar a origem no seguinte formato: Substitua cada *`espaço reservado para recurso de exemplo`* por suas próprias informações.

     > **Note**
     >
     > Substitua https://s3.*`aws-api-domain`* pela URL do bucket. Você pode copiar a URL do bucket no Amazon S3 na guia Objects (Objetos).

     ```
     {"path":"https://s3.aws-api-domain/path to script"}
     ```

     Veja um exemplo a seguir.

     ```
     {"path":"https://amzn-s3-demo-bucket.s3.us-east-2.amazonaws.com/scripts/shell/
     helloWorld.sh"}
     ```

   - No campo Command Line (Linha de comando), digite os parâmetros para a execução do script. Aqui está um exemplo.

     ```
     helloWorld.sh argument-1 argument-2
     ```

   - (Opcional) No campo Working Directory (Diretório de trabalho), insira o nome de um diretório do nó em que você deseja baixar e executar o script.
   - (Opcional) Em Execution Timeout, especifique o número de segundos para o sistema aguardar antes de a execução do comando de script falhar.
6. Na seção Targets (Destinos), escolha os nós gerenciados nos quais você quer executar essa operação, especificando as etiquetas, selecionando as instâncias ou dispositivos de borda manualmente ou especificando um grupo de recursos.

> **Tip**
>
> Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

7. Para Other parameters (Outros parâmetros):

   - Em Comment (Comentário), digite as informações sobre esse comando.
   - Em Timeout (seconds) (Tempo limite [segundos]), especifique o número de segundos para o sistema aguardar até a falha de execução do comando total.

8. Para Rate control (Controle de taxa):

   - Em Concurrency (Concorrência), especifique um número ou uma porcentagem de nós gerenciados nos quais executar o comando ao mesmo tempo.

     > **Note**
     >
     > Se você selecionou destinos especificando tags aplicadas a instâncias a nós gerenciados ou especificando grupos de recursos da AWS, e não tiver certeza de quantas instâncias são direcionadas, restrinja o número de instâncias que poderão executar o documento ao mesmo tempo, especificando uma porcentagem.

   - Em Error threshold (Limite de erro), especifique quando parar de executar o comando em outros nós depois de falhar em alguns ou em uma porcentagem de nós. Por exemplo, se você especificar três erros, o Systems Manager deixará de enviar o comando quando o 4° erro for recebido. Os nós gerenciados que continuam processando o comando também podem enviar erros.

9. (Opcional) Em Output options (Opções de saída), para salvar a saída do comando em um arquivo, selecione a caixa Write command output to an S3 bucket (Gravar saída do comando em um bucket do S3). Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

> **Note**
>
> As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil de instância (para instâncias do EC2) ou perfil de serviço do IAM (máquinas ativadas para ambientes híbridos) atribuído à instância, e não as do usuário

do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço do IAM associado ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

10. Na seção SNS notifications (Notificações do SNS), se quiser enviar notificações sobre o status da execução do comando, marque a caixa de seleção Enable SNS notifications (Habilitar notificações do SNS).

    Para obter mais informações sobre a configuração de notificações do Amazon SNS para o Run Command, consulte Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS.

11. Escolha Executar.

Executar um script shell no Amazon S3 (linha de comando)

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Execute o seguinte comando: Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

   > **Note**
   >
   > Substitua o *caminho de script* https://s3.*aws-api-domain*/ pela URL do seu bucket. Você pode copiar a URL do bucket no Amazon S3 na guia Objects (Objetos).

   Linux & macOS

   ```
   aws ssm send-command \
       --document-name "AWS-RunRemoteScript" \
       --output-s3-bucket-name "amzn-s3-demo-bucket" \
       --output-s3-key-prefix "key-prefix" \
       --targets "Key=InstanceIds,Values=instance-id" \
   ```

```
    --parameters '{"sourceType":["S3"],"sourceInfo":["{\"path\":\"https://
s3.aws-api-domain/script path\"}"],"commandLine":["script name and arguments"]}'
```

Windows

```
aws ssm send-command ^
    --document-name "AWS-RunRemoteScript" ^
    --output-s3-bucket-name "amzn-s3-demo-bucket" ^
    --output-s3-key-prefix "key-prefix" ^
    --targets "Key=InstanceIds,Values=instance-id" ^
    --parameters "sourceType"="S3",sourceInfo='{\"path\":\"https://s3.aws-api-
domain/script path\"}',"commandLine"="script name and arguments"
```

PowerShell

```
Send-SSMCommand `
    -DocumentName "AWS-RunRemoteScript" `
    -OutputS3BucketName "amzn-s3-demo-bucket" `
    -OutputS3KeyPrefix "key-prefix" `
    -Target @{Key="InstanceIds";Values=@("instance-id")} `
    -Parameter @{
        sourceType = "S3";
        sourceInfo = '{"path": "s3://bucket-name/path/to/script"}';
        commandLine = "script name and arguments"
    }
```

Fazer referência a segredos do AWS Secrets Manager em parâmetros do Parameter Store

O AWS Secrets Manager ajuda você a organizar e gerenciar dados de configuração importantes, como credenciais, senhas e chaves de licença. O Parameter Store, uma ferramenta do AWS Systems Manager, agora está integrado ao Secrets Manager para que você possa recuperar segredos do Secrets Manager ao usar outros Serviços da AWS que já oferecem suporte para referências a parâmetros do Parameter Store. Esses serviços incluem o Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic Container Service (Amazon ECS), AWS Lambda, AWS CloudFormation, AWS CodeBuild, AWS CodeDeploy e outras ferramentas do Systems Manager. Usando o Parameter Store para referenciar segredos do Secrets Manager, você cria um processo consistente e seguro para chamar e usar segredos e referenciar dados no código e nos scripts de configuração.

Para obter mais informações sobre o Secrets Manager, consulte O que é o AWS Secrets Manager no Manual do usuário do AWS Secrets Manager.

Restrições

Observe as seguintes restrições ao usar o Parameter Store para fazer referência a segredos do Secrets Manager:

- Você só pode recuperar segredos do Secrets Manager usando as operações de API GetParameter e GetParameters. As operações de modificação e de APIs de consulta avançada, como DescribeParameters ou GetParametersByPath, não são compatíveis com o Secrets Manager.
- Você pode usar a AWS Command Line Interface (AWS CLI), AWS Tools for Windows PowerShell e os SDKs para recuperar um segredo usando o Parameter Store.
- Os segredos do Secrets Manager no Parameter Store devem ter um prefixo de `/aws/reference/secretsmanager`. Veja os exemplos a seguir:

  ```
  /aws/reference/secretsmanager/CFCreds1
  ```

  ```
  /aws/reference/secretsmanager/myapp/db/password
  ```
- O Parameter Store honra as políticas (IAM) do AWS Identity and Access Management anexadas aos segredos do Secrets Manager. Por exemplo, se o Usuário 1 não tiver acesso ao Segredo A, o Usuário 1 não poderá recuperar o Segredo A usando o Parameter Store.
- Os parâmetros que fazem referência a segredos do Secrets Manager não podem usar os recursos de versionamento ou de histórico do Parameter Store.
- O Parameter Store honra os estágios de versão do Secrets Manager. Se você fizer referência a um estágio da versão, ele usará letras, números, um ponto (.), um hífen (-) ou um sublinhado (_). Todos os outros símbolos especificados no estágio da versão provocam falha na referência.

Como referenciar um segredo do Secrets Manager usando o Parameter Store

O procedimento a seguir descreve como fazer referência a um segredo do Secrets Manager usando APIs do Parameter Store. O procedimento faz referência a outros procedimentos no Manual do usuário do AWS Secrets Manager.

> **Note**
>
> Antes de começar, verifique se você tem permissão para fazer referência aos segredos do Secrets Manager em parâmetros do Parameter Store. Se você tiver permissões de

administrador no Secrets Manager e no Systems Manager, poderá fazer referência ou recuperar segredos usando as APIs do Parameter Store. Se você fizer referência a um segredo do Secrets Manager em um parâmetro do Parameter Store, mas não tiver permissão para acessar o segredo, a referência não funcionará. Para obter mais informações, consulte Controle de acesso e autenticação para o AWS Secrets Manager no Manual do usuário do AWS Secrets Manager.

Important

O Parameter Store funciona como um serviço de passagem para referências a segredos do Secrets Manager. O Parameter Store não retém dados ou metadados sobre os segredos. A referência é stateless.

Para referenciar um segredo do Secrets Manager usando o Parameter Store

1. Crie um segredo no Secrets Manager. Para obter mais informações, consulte Criar e gerenciar segredos com o AWS Secrets Manager.
2. Faça referência a um segredo usando a AWS CLI, o AWS Tools for Windows PowerShell ou o SDK. Ao referenciar um segredo do Secrets Manager, o nome deve começar com o seguinte caminho reservado: `/aws/reference/secretsmanager/`. Especificando esse caminho, o Systems Manager saberá que precisa recuperar o segredo do Secrets Manager em vez do Parameter Store. Aqui estão alguns nomes de exemplo que referenciam corretamente os segredos do Gerenciador de Segredos, `CFCreds1` e `DBPass`, usando o Parameter Store.

   - `/aws/reference/secretsmanager/CFCreds1`
   - `/aws/reference/secretsmanager/DBPass`

   Este é um exemplo de código Java que faz referência a uma chave de acesso e a uma chave secreta armazenadas no Secrets Manager. Esse exemplo de código define um cliente do Amazon DynamoDB. O código recupera dados da configuração e credenciais do Parameter Store. Os dados da configuração são armazenados como um parâmetro de string no Parameter Store, e as credenciais são armazenadas no Secrets Manager. Embora os dados da configuração e as credenciais sejam armazenados em serviços separados, os dois conjuntos de dados podem ser acessados no Parameter Store usando a API `GetParameter`.

```
/**
* Initialize Systems Manager client with default credentials
*/
AWSSimpleSystemsManagement ssm =
 AWSSimpleSystemsManagementClientBuilder.defaultClient();

...

/**
* Example method to launch DynamoDB client with credentials different from default
* @return DynamoDB client
*/
AmazonDynamoDB getDynamoDbClient() {
    //Getting AWS credentials from Secrets Manager using GetParameter
    BasicAWSCredentials differentAWSCreds = new BasicAWSCredentials(
            getParameter("/aws/reference/secretsmanager/access-key"),
            getParameter("/aws/reference/secretsmanager/secret-key"));

    //Initialize the DynamoDB client with different credentials
    final AmazonDynamoDB client = AmazonDynamoDBClient.builder()
            .withCredentials(new AWSStaticCredentialsProvider(differentAWSCreds))
            .withRegion(getParameter("region")) //Getting configuration from
 Parameter Store
            .build();
    return client;
}

/**
* Helper method to retrieve parameter value
* @param parameterName identifier of the parameter
* @return decrypted parameter value
*/
public GetParameterResult getParameter(String parameterName) {
    GetParameterRequest request = new GetParameterRequest();
    request.setName(parameterName);
    request.setWithDecryption(true);
    return ssm.newGetParameterCall().call(request).getParameter().getValue();
}
```

Estes são alguns exemplos da AWS CLI. Use o comando `aws secretsmanager list-secrets` para encontrar os nomes dos segredos.

Exemplo 1 da AWS CLI: referência usando o nome do segredo

Linux & macOS

```
aws ssm get-parameter \
    --name /aws/reference/secretsmanager/s1-secret \
    --with-decryption
```

Windows

```
aws ssm get-parameter ^
    --name /aws/reference/secretsmanager/s1-secret ^
    --with-decryption
```

O comando retorna informações como as seguintes.

```
{
    "Parameter": {
        "Name": "/aws/reference/secretsmanager/s1-secret",
        "Type": "SecureString",
        "Value": "Fl*MEishm!al875",
        "Version": 0,
        "SourceResult":
              "{
                   \"CreatedDate\": 1526334434.743,
                   \"Name\": \"s1-secret\",
                   \"VersionId\": \"aaabbbccc-1111-222-333-123456789\",
                   \"SecretString\": \"Fl*MEishm!al875\",
                   \"VersionStages\": [\"AWSCURRENT\"],
                   \"ARN\": \"arn:aws:secretsmanager:us-
east-2:123456789012:secret:s1-secret-E18LRP\"
               }"
        "LastModifiedDate": 2018-05-14T21:47:14.743Z,
        "ARN": "arn:aws:secretsmanager:us-east-2:123456789012:secret:s1-secret-
               E18LRP",
      }
}
```

Exemplo 2 da AWS CLI: referência que inclui o ID da versão

Linux & macOS

```
aws ssm get-parameter \
    --name /aws/reference/secretsmanager/s1-secret:11111-aaa-bbb-ccc-123456789 \
    --with-decryption
```

Windows

```
aws ssm get-parameter ^
    --name /aws/reference/secretsmanager/s1-secret:11111-aaa-bbb-ccc-123456789 ^
    --with-decryption
```

O comando retorna informações como as seguintes.

```
{
    "Parameter": {
        "Name": "/aws/reference/secretsmanager/s1-secret",
        "Type": "SecureString",
        "Value": "Fl*MEishm!al875",
        "Version": 0,
        "SourceResult":
              "{
                   \"CreatedDate\": 1526334434.743,
                   \"Name\": \"s1-secret\",
                   \"VersionId\": \"11111-aaa-bbb-ccc-123456789\",
                   \"SecretString\": \"Fl*MEishm!al875\",
                   \"VersionStages\": [\"AWSCURRENT\"],
                   \"ARN\": \"arn:aws:secretsmanager:us-
east-2:123456789012:secret:s1-secret-E18LRP\"
               }"
        "Selector": ":11111-aaa-bbb-ccc-123456789"
      }
        "LastModifiedDate": 2018-05-14T21:47:14.743Z,
        "ARN": "arn:aws:secretsmanager:us-east-2:123456789012:secret:s1-secret-
               E18LRP",
}
```

Exemplo 3 da AWS CLI: referência que inclui o estágio da versão

Linux & macOS

```
aws ssm get-parameter \
    --name /aws/reference/secretsmanager/s1-secret:AWSCURRENT \
    --with-decryption
```

Windows

```
aws ssm get-parameter ^
    --name /aws/reference/secretsmanager/s1-secret:AWSCURRENT ^
    --with-decryption
```

O comando retorna informações como as seguintes.

```
{
    "Parameter": {
        "Name": "/aws/reference/secretsmanager/s1-secret",
        "Type": "SecureString",
        "Value": "Fl*MEishm!al875",
        "Version": 0,
        "SourceResult":
              "{
                   \"CreatedDate\": 1526334434.743,
                   \"Name\": \"s1-secret\",
                   \"VersionId\": \"11111-aaa-bbb-ccc-123456789\",
                   \"SecretString\": \"Fl*MEishm!al875\",
                   \"VersionStages\": [\"AWSCURRENT\"],
                   \"ARN\": \"arn:aws:secretsmanager:us-
east-2:123456789012:secret:s1-secret-E18LRP\"
                }"
        "Selector": ":AWSCURRENT"
      }
      "LastModifiedDate": 2018-05-14T21:47:14.743Z,
      "ARN": "arn:aws:secretsmanager:us-east-2:123456789012:secret:s1-secret-
                E18LRP",
}
```

Criptografia do AWS KMS para parâmetros SecureString do AWS Systems Manager Parameter Store

Com o AWS Systems Manager Parameter Store, você pode criar parâmetros SecureString, que são parâmetros cujos nomes estão em texto sem formatação e os valores são criptografados. O Parameter Store usa o AWS KMS para criptografar e descriptografar os valores de parâmetro de parâmetros `SecureString`.

Com o Parameter Store, é possível criar, armazenar e gerenciar dados como parâmetros com valores. Você pode criar um parâmetro no Parameter Store e usá-lo em vários aplicativos e serviços, de acordo com políticas e permissões que você define. Quando você precisa alterar um valor de parâmetro, altere uma instância, em vez de gerenciar uma alteração sujeita a erros em várias origens. O Parameter Store oferece suporte a uma estrutura hierárquica para nomes de parâmetros, para que você possa qualificar um parâmetro para usos específicos.

Para gerenciar dados confidenciais, é possível criar parâmetros `SecureString`. O Parameter Store usa o AWS KMS keys para criptografar os valores de parâmetros `SecureString` quando você os cria ou altera. Ele também usa chaves do KMS para descriptografar os valores do parâmetro quando você os acessa. É possível usar a Chave gerenciada pela AWS gerenciada que o Parameter Store cria para sua conta ou especificar sua própria chave gerenciada pelo cliente.

> **Important**
>
> O Parameter Store oferece suporte somente a chaves do KMS simétricas. Não é possível usar uma chave do KMS assimétrica para criptografar os parâmetros. Para obter ajuda para determinar se uma chave do KMS é simétrica ou assimétrica, consulte Identificar tipos de chaves diferentes no Guia do desenvolvedor do AWS Key Management Service.

O Parameter Store oferece suporte a dois níveis de parâmetros de `SecureString`: padrão e avançado. Parâmetros padrão, que não podem exceder 4.096 bytes, são criptografados e descriptografados diretamente na chave do KMS especificada. Para criptografar e descriptografar parâmetros de `SecureString` avançados, o Parameter Store usa a criptografia de envelope com o AWS Encryption SDK. Você pode converter um parâmetro de `SecureString` padrão para um parâmetro avançado, mas não pode converter um parâmetro avançado para um padrão. Para obter mais informações sobre a diferença entre parâmetros `SecureString` padrão e avançados, consulte Gerenciar camadas de parâmetros.

Tópicos

- Proteger parâmetros SecureString padrão
- Proteger parâmetros SecureString avançados
- Definir permissões para criptografar e descriptografar valores de parâmetro
- Contexto de criptografia do Parameter Store
- Solucionar problemas de chaves do KMS no Parameter Store

Proteger parâmetros SecureString padrão

O Parameter Store não executa nenhuma operação de criptografia. Em vez disso, ele se baseia no AWS KMS para criptografar e descriptografar valores de parâmetro `SecureString`. Quando você cria ou altera um valor de parâmetro `SecureString` padrão, o Parameter Store chama a operação do Criptografar do AWS KMS. Essa operação usa uma chave do KMS de criptografia simétrica diretamente para criptografar o valor do parâmetro em vez de usar a chave do KMS para gerar uma chave de dados.

É possível selecionar a chave do KMS que o Parameter Store usa para criptografar o valor do parâmetro. Se você não especificar uma chave do KMS, o Parameter Store usará a Chave gerenciada pela AWS criada automaticamente pelo Systems Manager na sua conta. Essa chave do KMS tem o alias `aws/ssm`.

Para visualizar a chave do KMS padrão do `aws/ssm` da sua conta, use a operação DescribeKey na API do AWS KMS. O exemplo a seguir usa o comando `describe-key` na AWS Command Line Interface (AWS CLI) com o nome de alias `aws/ssm`.

```
aws kms describe-key \
    --key-id alias/aws/ssm
```

Para criar um parâmetro `SecureString` padrão, use a operação PutParameter na API do Systems Manager. Omita o parâmetro de `Tier` ou especifique um valor de `Standard`, que é o padrão. Inclua um parâmetro `Type` com um valor de `SecureString`. Para especificar uma chave do KMS, use o parâmetro `KeyId`. O padrão é a Chave gerenciada pela AWS para sua conta, `aws/ssm`.

O Parameter Store então chama a operação `Encrypt` do AWS KMS com a chave do KMS e o valor do parâmetro em texto não criptografado. O AWS KMS retorna o valor do parâmetro criptografado, que o Parameter Store armazena com o nome do parâmetro.

O exemplo a seguir usa o comando put-parameter do Systems Manager e seu parâmetro `--type` na AWS CLI para criar um parâmetro `SecureString`. Como o comando omite os parâmetros `--tier` e `--key-id` opcionais, o Parameter Store cria um parâmetro `SecureString` padrão e o criptografa com a Chave gerenciada pela AWS.

```
aws ssm put-parameter \
    --name MyParameter \
    --value "secret_value" \
    --type SecureString
```

O exemplo semelhante a seguir usa o parâmetro `--key-id` para especificar uma chave gerenciada pelo cliente. O exemplo usa um ID de chave do KMS para identificar a chave do KMS, mas você pode usar qualquer identificador de chave do KMS válido. Como o comando omite o parâmetro `Tier` (`--tier`), o Parameter Store cria um parâmetro de `SecureString` padrão, não um avançado.

```
aws ssm put-parameter \
    --name param1 \
    --value "secret" \
    --type SecureString \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Quando você obtém um parâmetro de `SecureString` do Parameter Store, seu valor é criptografado. Para obter um parâmetro, use a operação GetParameter na API do Systems Manager.

O exemplo a seguir usa o comando get-parameter do Systems Manager na AWS CLI para obter o parâmetro `MyParameter` do Parameter Store sem descriptografar seu valor.

```
aws ssm get-parameter --name MyParameter
```

```
{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value": "AQECAHgnOkMROh5LaLXkA4j0+vYi6tmM17Lg"
    }
}
```

Para descriptografar o valor de parâmetro antes de retorná-lo, defina o parâmetro `WithDecryption` de `GetParameter` como `true`. Quando você usa `WithDecryption`, o Parameter Store chama

a operação Descriptografar do AWS KMS em seu nome para descriptografar o valor do parâmetro. Como resultado, a solicitação de `GetParameter` retorna o parâmetro com um valor de parâmetro de texto simples, como mostrado no exemplo a seguir.

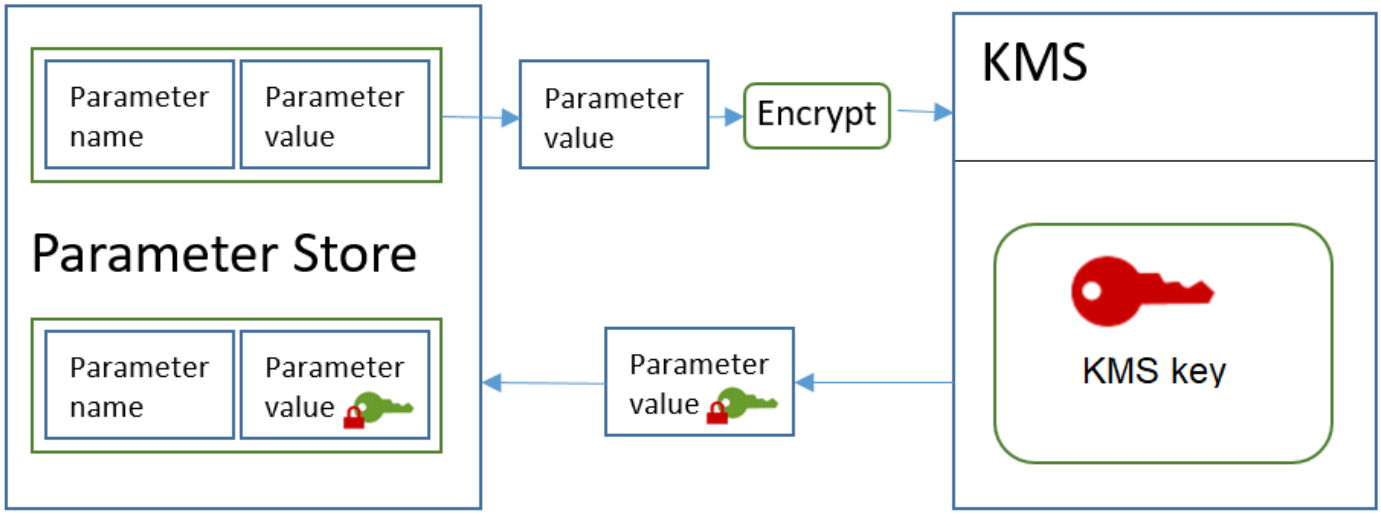
```
aws ssm get-parameter \
    --name MyParameter \
    --with-decryption
```

```
{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value": "secret_value"
    }
}
```

O fluxo de trabalho a seguir mostra como o Parameter Store usa uma chave do KMS para criptografar e descriptografar um parâmetro `SecureString` padrão.

Criptografar um parâmetro padrão

1. Quando você usa `PutParameter` para criar um parâmetro de `SecureString`, o Parameter Store envia uma solicitação `Encrypt` para o AWS KMS. Essa solicitação inclui o valor de parâmetro de texto sem formatação, a chave do KMS que você escolheu e o contexto de criptografia do Parameter Store. Durante a transmissão ao AWS KMS, o valor em texto simples no parâmetro `SecureString` é protegido por Transport Layer Security (TLS).
2. O AWS KMS criptografa o valor de parâmetro com a chave do KMS especificada e o contexto de criptografia. Ele retorna o texto cifrado para o Parameter Store, que armazena o nome de parâmetro e seu valor criptografado.



Descriptografar um parâmetro padrão

1. Quando você inclui o parâmetro `WithDecryption` em uma solicitação `GetParameter`, o Parameter Store envia uma solicitação `Decrypt` para o AWS KMS com o valor criptografado de parâmetro de `SecureString` e o contexto de criptografia do Parameter Store.
2. O AWS KMS usa a mesma chave do KMS e o contexto de criptografia fornecido para descriptografar o valor criptografado. Ele retorna o valor do parâmetro em texto não criptografado (descriptografado) ao Parameter Store. Durante a transmissão, os dados em texto simples são protegidos por TLS.
3. O Parameter Store retorna o valor do parâmetro em texto não criptografado na resposta de `GetParameter`.

Proteger parâmetros SecureString avançados

Quando você usa o `PutParameter` para criar um parâmetro `SecureString` avançado, o Parameter Store usa a criptografia de envelope com o AWS Encryption SDK e uma AWS KMS key de criptografia simétrica para proteger o valor de parâmetro. Cada valor de parâmetro avançado é criptografado com uma chave de dados exclusiva, e a chave de dados é criptografada em uma chave do KMS. É possível usar a Chave gerenciada pela AWS para a conta (`aws/ssm`) ou qualquer chave gerenciada pelo cliente.

O AWS Encryption SDK é uma biblioteca no lado do cliente de software livre que ajuda você a criptografar e descriptografar dados usando padrões do setor e práticas recomendadas. Ele é compatível com várias plataformas e várias linguagens de programação, incluindo uma interface

de linha de comando. Você pode visualizar o código-fonte e contribuir para o desenvolvimento no GitHub.

Para cada valor de parâmetro de `SecureString`, o Parameter Store chama o AWS Encryption SDK para criptografar o valor do parâmetro usando uma chave de dados exclusiva que o AWS KMS gera (GenerateDataKey). O AWS Encryption SDK retorna para o Parameter Store uma mensagem criptografada que inclui o valor de parâmetro criptografado e uma cópia criptografada da chave de dados exclusiva. O Parameter Store armazena toda a mensagem criptografada no valor de parâmetro de `SecureString`. Quando você obtém um parâmetro de `SecureString` avançado, Parameter Store usa o valor AWS Encryption SDK para descriptografar o valor de parâmetro. Isso requer uma chamada para o AWS KMS descriptografar a chave de dados criptografada.

Para criar um parâmetro `SecureString` avançado, use a operação PutParameter na API do Systems Manager. Defina o valor do parâmetro `Tier` como `Advanced`. Inclua um parâmetro `Type` com um valor de `SecureString`. Para especificar uma chave do KMS, use o parâmetro `KeyId`. O padrão é a Chave gerenciada pela AWS para sua conta, `aws/ssm`.

```
aws ssm put-parameter \
    --name MyParameter \
    --value "secret_value" \
    --type SecureString \
    --tier Advanced
```

O exemplo semelhante a seguir usa o parâmetro `--key-id` para especificar uma chave gerenciada pelo cliente. O exemplo usa o Amazon Resource Name (ARN) da chave do KMS, mas você pode usar qualquer identificador válido de chave do KMS.

```
aws ssm put-parameter \
    --name MyParameter \
    --value "secret_value" \
    --type SecureString \
    --tier Advanced \
    --key-id arn:aws:kms:us-
east-2:987654321098:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

Quando você obtém um parâmetro de `SecureString` o Parameter Store, seu valor é a mensagem criptografada que o AWS Encryption SDK retornou. Para obter um parâmetro, use a operação GetParameter na API do Systems Manager.

O exemplo a seguir usa a operação `GetParameter` do Systems Manager para obter o parâmetro `MyParameter` do Parameter Store sem descriptografar seu valor.

```
aws ssm get-parameter --name MyParameter
```

```
{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value": "AQECAHgnOkMROh5LaLXkA4j0+vYi6tmM17Lg"
    }
}
```

Para descriptografar o valor de parâmetro antes de retorná-lo, defina o parâmetro `WithDecryption` de `GetParameter` como `true`. Quando você usa `WithDecryption`, o Parameter Store chama a operação Descriptografar do AWS KMS em seu nome para descriptografar o valor do parâmetro. Como resultado, a solicitação de `GetParameter` retorna o parâmetro com um valor de parâmetro de texto simples, como mostrado no exemplo a seguir.

```
aws ssm get-parameter \
    --name MyParameter \
    --with-decryption
```

```
{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value": "secret_value"
    }
}
```

Você não pode converter um parâmetro de `SecureString` avançado para um padrão, mas pode converter um `SecureString` padrão para um avançado. Para converter um parâmetro de `SecureString` padrão para um de `SecureString` avançado, use a operação `PutParameter` com o parâmetro `Overwrite`. O `Type` deve ser `SecureString`, e o valor `Tier` deve ser `Advanced`. O parâmetro `KeyId`, que identifica uma chave gerenciada pelo cliente, é opcional. Se você o omitir, o Parameter Store usará a Chave gerenciada pela AWS para a conta. É possível especificar qualquer chave do KMS que a entidade principal tenha permissão para usar, mesmo se você tiver usado uma chave do KMS diferente para criptografar o parâmetro padrão.

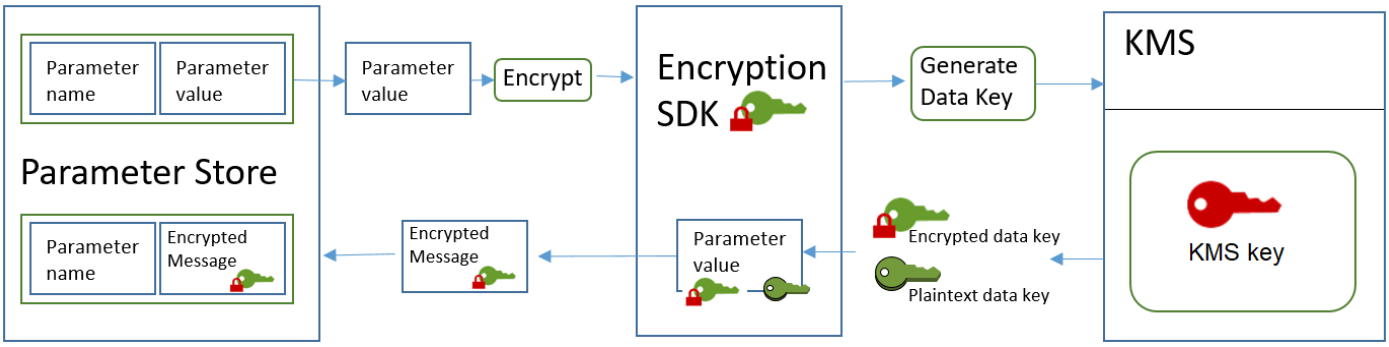
Quando você usa o parâmetro `Overwrite`, o Parameter Store usa o AWS Encryption SDK para criptografar o valor de parâmetro. Ele armazena a mensagem recém-criptografada no Parameter Store.

```
aws ssm put-parameter \
    --name myStdParameter \
    --value "secret_value"  \
    --type SecureString \
    --tier Advanced \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --overwrite
```

O fluxo de trabalho a seguir mostra como o Parameter Store usa uma chave do KMS para criptografar e descriptografar um parâmetro `SecureString` avançado.

Criptografar um parâmetro avançado

1. Quando o `PutParameter` é usado para criar um parâmetro `SecureString` avançado, o Parameter Store usa o AWS Encryption SDK e o AWS KMS para criptografar o valor do parâmetro. O Parameter Store chama o AWS Encryption SDK com o valor do parâmetro, a chave do KMS especificada e o contexto de criptografia do Parameter Store.
2. O AWS Encryption SDK envia uma solicitação GenerateDataKey ao AWS KMS com o identificador da chave do KMS especificada e o contexto de criptografia do Parameter Store. O AWS KMS retorna duas cópias da chave de dados exclusiva: uma em texto simples e uma criptografada na chave do KMS. (O contexto de criptografia é usado ao criptografar a chave de dados.)
3. O AWS Encryption SDK usa a chave de dados em texto simples para criptografar valor de parâmetro. Ele retorna uma mensagem criptografada que inclui o valor de parâmetro criptografado, a chave de dados criptografada e outros dados, incluindo o contexto criptografia do Parameter Store.
4. O Parameter Store armazena a mensagem criptografada como valor de parâmetro.



Descriptografar um parâmetro avançado

1. Você pode incluir o parâmetro `WithDecryption` em uma solicitação `GetParameter` para obter um parâmetro de `SecureString` avançado. Quando você fizer isso, o Parameter Store repassará a mensagem criptografada do valor de parâmetro para um método de descriptografia do AWS Encryption SDK.
2. O AWS Encryption SDK chama a operação do AWS KMS Decrypt. Ele repassa a chave de dados criptografada e o contexto de criptografia do Parameter Store da mensagem criptografada.
3. O AWS KMS usa a chave do KMS e o contexto de criptografia do Parameter Store para descriptografar a chave de dados criptografada. Ele retorna a chave de dados de texto simples (descriptografada) ao AWS Encryption SDK.
4. O AWS Encryption SDK usa a chave de dados de texto simples para descriptografar o valor de parâmetro. Ele retorna o valor de parâmetro de texto simples ao Parameter Store.
5. O Parameter Store verifica o contexto de criptografia e retorna a você o valor de parâmetro de texto simples na resposta `GetParameter`.

Definir permissões para criptografar e descriptografar valores de parâmetro

Para criptografar um valor de parâmetro de `SecureString` padrão, o usuário precisa da permissão `kms:Encrypt`. Para criptografar um valor de parâmetro de `SecureString` avançado, o usuário precisa da permissão `kms:GenerateDataKey`. Para descriptografar qualquer tipo de valor de parâmetro de `SecureString`, o usuário precisa da permissão `kms:Decrypt`.

É possível usar políticas do AWS Identity and Access Management (IAM) para permitir ou negar permissão para um usuário chamar as operações `PutParameter` e `GetParameter` do Systems Manager.

Além disso, se estiver usando chaves gerenciadas pelo cliente para criptografar seus valores de parâmetros `SecureString`, você poderá usar políticas do IAM e políticas de chaves para gerenciar permissões de criptografia e descriptografia. No entanto, você não pode estabelecer políticas de controle de acesso para a chave do KMS `aws/ssm` padrão. Para obter informações sobre como controlar o acesso a chaves gerenciadas pelo cliente, consulte Acesso e permissões de chaves do KMS no Guia do desenvolvedor do AWS Key Management Service.

O exemplo a seguir mostra uma política do IAM criada para parâmetros `SecureString` padrão. Ela permite que o usuário chame a operação `PutParameter` do Systems Manager em todos os parâmetros no caminho `FinancialParameters`. A política também permite que o usuário chame a operação do AWS KMS `Encrypt` em um exemplo de chave gerenciada pelo cliente.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:us-east-2:111122223333:parameter/
FinancialParameters/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt"
            ],
            "Resource": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ]
}
```

O exemplo seguinte mostra uma política do IAM criada para parâmetros `SecureString` avançados. Ela permite que o usuário chame a operação `PutParameter` do Systems Manager em todos os

parâmetros no caminho `ReservedParameters`. A política também permite que o usuário chame a operação do AWS KMS `GenerateDataKey` em um exemplo de chave gerenciada pelo cliente.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:us-east-2:111122223333:parameter/
ReservedParameters/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey"
            ],
            "Resource": "arn:aws:kms:us-east-2:111122223333:key/key-id"
        }
    ]
}
```

O último exemplo também mostra uma política do IAM que pode ser usada para parâmetros de `SecureString` padrão ou avançados. Ela permite que o usuário chame as operações `GetParameter` do Systems Manager (e operações relacionadas) em todos os parâmetros no caminho de `ITParameters`. A política também permite que o usuário chame a operação do AWS KMS `Decrypt` em um exemplo de chave gerenciada pelo cliente.

JSON

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

```
            "Action": [
                "ssm:GetParameter*"
            ],
            "Resource": "arn:aws:ssm:us-east-2:111122223333:parameter/
ITParameters/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ]
}
```

Contexto de criptografia do Parameter Store

Um contexto de criptografia é um conjunto de pares de chave-valor que contêm dados arbitrários não secretos. Quando você inclui um contexto de criptografia em uma solicitação para criptografar dados, o AWS KMS vincula de forma criptográfica o contexto de criptografia aos dados criptografados. Para descriptografar os dados, você deve passar o mesmo contexto de criptografia.

Você também pode usar o contexto de criptografia para identificar uma operação de criptografia em registros de auditoria e logs. O contexto de criptografia aparece em texto simples em logs, como logs do AWS CloudTrail.

O AWS Encryption SDK também obtém um contexto de criptografia, embora ele lide com ele de forma diferente. O Parameter Store fornece o contexto de criptografia para o método de criptografia. O AWS Encryption SDK vincula forma criptográfica o contexto de criptografia aos dados criptografados. Isso inclui o contexto de criptografia em texto simples no cabeçalho da mensagem criptografada retornada por ele. No entanto, ao contrário do AWS KMS, os métodos de descriptografia do AWS Encryption SDK não têm um contexto de criptografia como entrada. Em vez disso, ao descriptografar dados, o AWS Encryption SDK obtém o contexto de criptografia na mensagem criptografada. O Parameter Store verifica se o contexto de criptografia inclui o valor que ele espera antes de retornar o valor de parâmetro de texto simples para você.

O Parameter Store usa o seguinte contexto de criptografia em suas operações de criptografia:

- Chave: `PARAMETER_ARN`
- Valor: O Amazon Resource Name (ARN) do parâmetro que está sendo criptografado.

O formato do contexto de criptografia é o seguinte:

```
"PARAMETER_ARN":"arn:aws:ssm:region-id:account-id:parameter/parameter-name"
```

Por exemplo, o Parameter Store inclui esse contexto de criptografia em chamadas para criptografar e descriptografar o parâmetro `MyParameter` em um exemplo de região e Conta da AWS.

```
"PARAMETER_ARN":"arn:aws:ssm:us-east-2:111122223333:parameter/MyParameter"
```

Se o parâmetro estiver em um caminho hierárquico do Parameter Store, o caminho e o nome serão incluídos no contexto de criptografia. Por exemplo, esse contexto de criptografia é usado ao criptografar ou descriptografar o parâmetro `MyParameter` no caminho `/ReadableParameters` em um exemplo de Conta da AWS e região.

```
"PARAMETER_ARN":"arn:aws:ssm:us-east-2:111122223333:parameter/ReadableParameters/
MyParameter"
```

Você pode descriptografar um valor de parâmetro `SecureString` criptografado chamando a operação do AWS KMS `Decrypt` com o contexto de criptografia correto e o valor de parâmetro criptografado retornado pela operação `GetParameter` do Systems Manager. No entanto, recomendamos que você descriptografe valores de parâmetro do Parameter Store usando a operação `GetParameter` com o parâmetro `WithDecryption`.

Você também pode incluir o contexto de criptografia em uma política do IAM. Por exemplo, você pode permitir que um usuário descriptografe apenas determinado valor de parâmetro ou conjunto de valores de parâmetro.

O exemplo a seguir de declaração de política do IAM permite que o usuário obtenha o valor do parâmetro `MyParameter` e descriptografe o valor usando a chave do KMS especificada. No entanto, as permissões são aplicáveis somente quando o contexto de criptografia corresponde à string especificada. Essas permissões não são aplicáveis a nenhum outro parâmetro ou chave do KMS, e ocorrerá falha na chamada para `GetParameter` se o contexto de criptografia não corresponder à string.

Antes de usar uma declaração de política como essa, substitua os *ARNs de exemplo* por valores válidos.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter*"
            ],
            "Resource": "arn:aws:ssm:us-
east-2:111122223333:parameter/MyParameter"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-east-2:111122223333:key/key-id",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:PARAMETER_ARN":"arn:aws:ssm:us-
east-2:111122223333:parameter/MyParameter"
                }
            }
        }
    ]
}
```

Solucionar problemas de chaves do KMS no Parameter Store

Para executar qualquer operação em um parâmetro `SecureString`, o Parameter Store deve ser capaz de usar a chave do AWS KMS que você especificou para a operação desejada. A maioria das falhas do Parameter Store relacionadas a chaves do KMS é causada pelos seguintes problemas:

- As credenciais que uma aplicação está usando não têm permissão para executar a ação especificada na chave do KMS.

Para corrigir este erro, execute a aplicação com credenciais diferentes ou revise a política da IAM ou de chaves que está impedindo a operação. Para obter ajuda com o IAM e as políticas de chave do AWS KMS, consulte Acesso e permissões de chave do KMS no Guia do desenvolvedor do AWS Key Management Service.

- A chave do KMS não foi encontrada.

  Isso geralmente acontece quando você usa um identificador incorreto para a chave do KMS. Encontre os identificadores corretos para a chave do KMS e tente o comando novamente.

- A chave do KMS não está habilitada. Quando isso ocorre, o Parameter Store retorna uma exceção InvalidKeyId com uma mensagem de erro detalhada do AWS KMS. Se o estado da chave do KMS for `Disabled`, habilite-a. Se for `Pending Import`, conclua o procedimento de importação. Se o estado da chave for `Pending Deletion`, cancele a exclusão da chave ou use uma chave do KMS diferente.

  Para encontrar o estado de chave de uma chave do KMS, use a operação DescribeKey.

Use parâmetros do Parameter Store no Amazon Elastic Kubernetes Service.

Para mostrar parâmetros do Parameter Store, uma ferramenta do AWS Systems Manager, como arquivos montados em pods do Amazon EKS, você pode usar o AWS Secrets and Configuration Provider para o driver CSI do Secrets Store do Kubernetes. O ASCP funciona com o Amazon Elastic Kubernetes Service 1.17+ executando um grupo de nós do Amazon EC2. Grupos de nós do AWS Fargate não são compatíveis.

Com o ASCP, você pode armazenar e gerenciar seu parâmetro no Parameter Store e depois recuperá-lo por meio das workloads executadas no Amazon EKS. Se o parâmetro contiver vários pares de chave/valor no formato JSON, você poderá escolher quais montará no Amazon EKS. O ASCP usa a sintaxe JMESPath para consultar os pares de chave/valor no seu segredo. O ASCP também funciona com os segredos do AWS Secrets Manager.

O ASCP oferece dois métodos de autenticação com o Amazon EKS. A primeira abordagem usa perfis do IAM para contas de serviço (IRSA). A segunda abordagem usa a Identidade de Pods. Cada abordagem tem seus benefícios e casos de uso.

ASCP com perfis do IAM para contas de serviço (IRSA)

O ASCP com perfis do IAM para contas de serviço (IRSA) permite que você monte parâmetros do Parameter Store como arquivos nos pods do Amazon EKS. Essa abordagem é adequada quando:

- Você precisa montar parâmetros como arquivos nos pods.
- Você está usando a versão 1.17 ou posterior do Amazon EKS com grupos de nós do Amazon EC2.
- Você deseja recuperar pares de chave/valor específicos de parâmetros formatados em JSON.

Para obter mais informações, consulte the section called “Integrar o ASCP com IRSA para o Amazon EKS”.

ASCP com Identidade de Pods

ASCP com Identidade de Pods do EKS

O método ASCP com Identidade de Pods aumenta a segurança e simplifica a configuração para acessar os parâmetros no Parameter Store. Essa abordagem é benéfica quando:

- Você precisa de um gerenciamento de permissões mais granular no nível do pod.
- Você está usando a versão 1.24 ou posterior do Amazon EKS.
- Você quer performance e escalabilidade aprimorados.

Para obter mais informações, consulte the section called “Integrar o ASCP com a Identidade de Pods para Amazon EKS”.

Como escolher a abordagem correta

Considere os seguintes fatores ao decidir entre o ASCP com IRSA e o ASCP com Identidade de Pods:

- Versão do Amazon EKS: a Identidade de Pods requer o Amazon EKS 1.24+, enquanto o driver CSI funciona com o Amazon EKS 1.17+.
- Requisitos de segurança: a Identidade de Pods oferece um controle mais granular no nível do pod.
- Performance: a Identidade de Pods geralmente tem melhor performance em ambientes de alta escala.

- Complexidade: a Identidade de Pods simplifica a configuração ao eliminar a necessidade de contas de serviço separadas.

Escolha o método que melhor se alinhe aos seus requisitos específicos e ao ambiente do Amazon EKS.

Instalar o ASCP para Amazon EKS

Esta seção explica como instalar o AWS Secrets and Configuration Provider para Amazon EKS. Com o ASCP, você pode montar parâmetros do Parameter Store e segredos do AWS Secrets Manager como arquivos nos pods do Amazon EKS.

Pré-requisitos

- Um cluster do Amazon EKS
  - Versão 1.24 ou posterior da Identidade de Pods
  - Versão 1.17 ou posterior do IRSA
- A AWS CLI está instalada e configurada
- kubectl instalado e configurado para o cluster do Amazon EKS
- Helm (versão 3.0 ou superior)

Instalar e configurar o ASCP

O ASCP está disponível no GitHub no repositório secrets-store-csi-driver-provider-aws. O repositório também contém arquivos YAML de exemplo para criar e montar um segredo alterando o valor de `objectType` de `secretsmanager` para `ssmparameter`.

Durante a instalação, é possível configurar o ASCP para usar um endpoint do FIPS. Para obter uma lista de endpoints do Systems Manager, consulte os endpoints do serviço do Systems Manager no Referência geral da Amazon Web Services.

Para instalar o ASCP usando o Helm

1. Para garantir que o repositório esteja apontando para os gráficos mais recentes, use `helm repo update.`
2. Adicione o gráfico de driver CSI do armazenamento de segredos.

```
helm repo add secrets-store-csi-driver https://kubernetes-sigs.github.io/secrets-store-csi-driver/charts
```

3. Instale o chart. Para configurar o controle de utilização, adicione o sinalizador a seguir: `--set-json 'k8sThrottlingParams={"qps": "number of queries per second", "burst": "number of queries per second"}'`

```
helm install -n kube-system csi-secrets-store secrets-store-csi-driver/secrets-store-csi-driver
```

4. Adicione o gráfico ASCP.

```
helm repo add aws-secrets-manager https://aws.github.io/secrets-store-csi-driver-provider-aws
```

5. Instale o chart. Para usar um endpoint do FIPS, adicione o sinalizador a seguir: `--set useFipsEndpoint=true`

```
helm install -n kube-system secrets-provider-aws aws-secrets-manager/secrets-store-csi-driver-provider-aws
```

Para instalar usando o YAML no repositório

- Use os seguintes comandos.

```
helm repo add secrets-store-csi-driver https://kubernetes-sigs.github.io/secrets-store-csi-driver/charts
helm install -n kube-system csi-secrets-store secrets-store-csi-driver/secrets-store-csi-driver
kubectl apply -f https://raw.githubusercontent.com/aws/secrets-store-csi-driver-provider-aws/main/deployment/aws-provider-installer.yaml
```

Verificar as instalações

Para verificar as instalações do cluster do EKS, do driver CSI do Secrets Store e do plug-in do ASCP, siga estas etapas:

1. Verifique o cluster do EKS:

```
eksctl get cluster --name clusterName
```

Este comando deve retornar informações sobre o cluster.

2. Verifique a instalação do driver CSI do Secrets Store:

```
kubectl get pods -n kube-system -l app=secrets-store-csi-driver
```

Você deve ver pods em execução com nomes como `csi-secrets-store-secrets-store-csi-driver-xxx`.

3. Verifique a instalação do plug-in do ASCP:

YAML installation

```
$ kubectl get pods -n kube-system -l app=csi-secrets-store-provider-aws
```

Resultado do exemplo:

```
NAME                                     READY   STATUS    RESTARTS   AGE
csi-secrets-store-provider-aws-12345      1/1     Running   0          2m
```

Helm installation

```
$  kubectl get pods -n kube-system -l app=secrets-store-csi-driver-provider-aws
```

Resultado do exemplo:

```
NAME                                                READY   STATUS    RESTARTS
 AGE
secrets-provider-aws-secrets-store-csi-driver-provider-67890       1/1
 Running   0          2m
```

Você deve ver pods no estado `Running`.

Depois de executar esses comandos, se tudo estiver configurado corretamente, você deverá ver todos os componentes funcionando sem erros. Se você encontrar algum problema, talvez seja necessário solucionar verificando os logs dos pods específicos que estão com problemas.

Solução de problemas

1. Para verificar os logs do provedor do ASCP, execute:

```
kubectl logs -n kube-system -l app=csi-secrets-store-provider-aws
```

2. Verifique o status de todos os pods no namespace `kube-system`.

   Substitua o *texto do espaço reservado padrão* pelo seu próprio ID do pod:

```
kubectl -n kube-system get pods
```

```
kubectl -n kube-system logs pod/pod-id
```

   Todos os pods relacionados ao driver CSI e ao ASCP devem estar no estado “Running”.

3. Verifique a versão do driver CSI:

```
kubectl get csidriver secrets-store.csi.k8s.io -o yaml
```

   Este comando deve retornar informações sobre o driver CSI instalado.

Recursos adicionais

Para obter mais informações sobre o uso do ASCP com o Amazon EKS, consulte os seguintes recursos:

- Uso da Identidade de Pods com o Amazon EKS
- Driver CSI do AWS Secrets Store no GitHub

Usar o CSI do AWS Secrets and Configuration Provider com a Identidade de Pods para Amazon EKS

A integração do AWS Secrets and Configuration Provider com o agente da Identidade de Pods para Amazon Elastic Kubernetes Service fornece segurança aprimorada, configuração simplificada

e performance aprimorada para aplicações executadas no Amazon EKS. A Identidade de Pods simplifica a autenticação do AWS Identity and Access Management (IAM) para Amazon EKS ao recuperar parâmetros do Parameter Store do AWS Systems Manager ou segredos do Secrets Manager.

A Identidade de Pods do Amazon EKS simplifica o processo de configuração de permissões do IAM para aplicações do Kubernetes, possibilitando que as permissões sejam configuradas diretamente por meio das interfaces do Amazon EKS, reduzindo o número de etapas e eliminando a necessidade de alternar entre os serviços do Amazon EKS e do IAM. A Identidade de Pods permite o uso de um único perfil do IAM em vários clusters sem atualizar as políticas de confiança, e oferece suporte a tags de sessão de perfis para um controle de acesso mais granular. Essa abordagem não apenas simplifica o gerenciamento de políticas, permitindo a reutilização de políticas de permissão em todos os perfis, mas também aumenta a segurança ao permitir o acesso a recursos da AWS com base nas tags correspondentes.

Como funciona

1. A Identidade de Pods atribui um perfil do IAM ao pod.
2. O ASCP usa esse perfil para se autenticar com os Serviços da AWS.
3. Se autorizado, o ASCP recupera os parâmetros solicitados e os disponibiliza para o pod.

Para obter mais informações, consulte Como a Identidade de Pods do Amazon EKS funciona no Guia do usuário do Amazon EKS.

Pré-requisitos

> **Important**
>
> A Identidade de Pods é compatível somente com o Amazon EKS na nuvem. Ela não é compatível com o Amazon EKS Anywhere, o Serviço Red Hat OpenShift na AWS ou com clusters autogerenciados do Kubernetes em instâncias do Amazon EC2.

- Cluster do Amazon EKS (versão 1.24 ou posterior)
- Acesso à AWS CLI e ao cluster do Amazon EKS por meio do `kubectl`
- (Opcional) Acesso a duas Contas da AWS para acesso entre contas

Instalar o agente da Identidade de Pods do Amazon EKS

Para usar a Identidade de Pods com o cluster, você deve instalar o complemento do agente da Identidade de Pods do Amazon EKS.

Para instalar o complemento agente da Identidade de Pods

- Instale o complemento do agente da Identidade de Pods no cluster.

  Substitua o *texto do espaço reservado padrão* pelos seus próprios valores:

```
eksctl create addon \
  --name eks-pod-identity-agent \
  --cluster clusterName \
  --region region
```

Configurar o ASCP com a Identidade de Pods

1. Crie uma política de permissões que conceda as permissões `ssm:GetParameters` e `ssm:DescribeParameters` aos parâmetros que o pod precisa acessar.
2. Crie um perfil do IAM que possa ser assumido pela entidade principal do serviço do Amazon EKS para a Identidade de Pods:

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Service": "pods.eks.amazonaws.com"
        },
        "Action": [
          "sts:AssumeRole",
          "sts:TagSession"
        ],
        "Resource": "arn:aws:iam::123456789012:role/EKSPodIdentityRole"
      }
    ]
```

```
}
```

Anexe a política do IAM à função.

Substitua o *texto do espaço reservado padrão* pelos seus próprios valores:

```
aws iam attach-role-policy \
  --role-name MY_ROLE \
  --policy-arn POLICY_ARN
```

3. Crie uma associação de Identidade de Pods. Para obter um exemplo, consulte Criar uma associação de Identidade de Pods no Guia do usuário do Amazon EKS
4. Crie o `SecretProviderClass` que especifica quais parâmetros ou segredos devem ser montados no pod:

```
kubectl apply -f kubectl apply -f https://raw.githubusercontent.com/aws/
secrets-store-csi-driver-provider-aws/main/examples/ExampleSecretProviderClass-
PodIdentity.yaml
```

A principal diferença no `SecretProviderClass` entre o IRSA e a Identidade de Pods é o parâmetro opcional `usePodIdentity`. É um campo opcional que determina a abordagem de autenticação. Quando não especificado, o padrão é usar os perfis do IAM para contas de serviço (IRSA).

- Para usar a Identidade de Pods do EKS, use qualquer um destes valores: `"true", "True", "TRUE", "t", "T"`.
- Para usar explicitamente o IRSA, defina para qualquer um destes valores: `"false", "False", "FALSE", "f", or "F"`.

5. Implante o pod que monta os parâmetros ou segredos em `/mnt/secrets-store`:

```
kubectl apply -f kubectl apply -f https://raw.githubusercontent.com/aws/secrets-
store-csi-driver-provider-aws/main/examples/ExampleDeployment-PodIdentity.yaml
```

6. Se você usar um cluster privado do Amazon EKS, certifique-se de que a VPC na qual o cluster se encontra tenha um endpoint do AWS STS. Para obter informações sobre a criação de um endpoint, consulte Endpoints da VPC da interface no Guia do usuário do AWS Identity and Access Management.

Verificar a montagem do segredo

Para verificar se o parâmetro ou o segredo foi montado corretamente, execute o comando a seguir.

Substitua o *texto do espaço reservado padrão* pelos seus próprios valores:

```
kubectl exec -it $(kubectl get pods | awk '/pod-identity-deployment/{print $1}' | head
 -1) -- cat /mnt/secrets-store/MyParameter
```

Para configurar a Identidade de Pods do Amazon EKS para acessar os parâmetros no Parameter Store

1. Crie uma política de permissões que conceda as permissões `ssm:GetParameters` e `ssm:DescribeParameters` aos parâmetros que o pod precisa acessar.
2. Crie um parâmetro no Parameter Store, caso ainda não tenha um. Para mais informações, consulte Criar parâmetros do Parameter Store no Systems Manager.

Solução de problemas

É possível visualizar a maioria dos erros ao descrever a implantação do pod.

Para ver mensagens de erro para o contêiner

1. Obtenha uma lista de nomes de pods com o comando a seguir. Se você não estiver usando o namespace padrão, use `-n` *namespace*.

   ```
   kubectl get pods
   ```

2. Para descrever o pod, no comando a seguir, em *pod-id*, use o ID dos pods encontrados na etapa anterior. Se você não estiver usando o namespace padrão, use `-n` *NAMESPACE*.

   ```
   kubectl describe pod/pod-id
   ```

Para ver erros para o ASCP

- Para encontrar mais informações nos logs do provedor, no comando a seguir, em *PODID*, use a ID do pod csi-secrets-store-provedor-aws.

  ```
  kubectl -n kube-system get pods
  ```

```
kubectl -n kube-system logs pod/pod-id
```

Usar o CSI do AWS Secrets and Configuration Provider com perfis do IAM para contas de serviço (IRSA)

Tópicos

- Pré-requisitos
- Configurar o controle de acesso
- Identificar quais parâmetros montar
- Solução de problemas

Pré-requisitos

- Cluster do Amazon EKS (versão 1.17 ou posterior)
- Acesso à AWS CLI e ao cluster do Amazon EKS por meio do `kubectl`

Configurar o controle de acesso

O ASCP recupera a Identidade de Pods do Amazon EKS e a substitui por um perfil do IAM. Você define permissões em uma política do IAM para esse perfil do IAM. Quando o ASCP assume o perfil do IAM, ele obtém acesso aos parâmetros que você autorizou. Outros contêineres não podem acessar os parâmetros, a menos que você também os associe à função do IAM.

Para conceder ao seu pod do Amazon EKS acesso aos parâmetros no Parameter Store

1. Crie uma política de permissões que conceda as permissões `ssm:GetParameters` e `ssm:DescribeParameters` aos parâmetros que o pod precisa acessar.
2. Crie um provedor IAM OIDC Connect (OIDC) para o cluster, se você ainda não tiver um. Para obter mais informações, consulte Criação de um provedor IAM OIDC para o seu cluster no Guia do Usuário do Amazon EKS.
3. Crie um perfil do IAM para conta de serviço e anexe a política a ele. Para obter mais informações, consulte Criação de um perfil do IAM para uma conta de serviço no Guia do usuário do Amazon EKS.
4. Se você usar um cluster privado do Amazon EKS, certifique-se de que a VPC na qual o cluster se encontra tenha um endpoint do AWS STS. Para obter informações sobre a criação de um

endpoint, consulte Endpoints da VPC da interface no Guia do usuário do AWS Identity and Access Management.

Identificar quais parâmetros montar

Para determinar quais parâmetros o ASCP monta no Amazon EKS como arquivos no sistema de arquivos, você cria um arquivo YAML the section called “SecretProviderClass”. O `SecretProviderClass` lista os parâmetros a serem montados e o nome do arquivo no qual montá-los. O`SecretProviderClass` deve estar no mesmo namespace que o pod do Amazon EKS ao qual ele faz referência.

Montar os parâmetros como arquivos

As seguintes instruções mostram como montar parâmetros como arquivos usando exemplos de arquivos YAML: ExampleSecretProviderClass.yaml e ExampleDeployment.yaml.

Para montar parâmetros no Amazon EKS

1. Aplique o `SecretProviderClass` ao pod:

```
kubectl apply -f ExampleSecretProviderClass.yaml
```

2. Implante o pod:

```
kubectl apply -f ExampleDeployment.yaml
```

3. O ASCP monta os arquivos.

Solução de problemas

É possível visualizar a maioria dos erros ao descrever a implantação do pod.

Para ver mensagens de erro para o contêiner

1. Obtenha uma lista de nomes de pods com o comando a seguir. Se você não estiver usando o namespace padrão, use `-n` *`name-space`*.

```
kubectl get pods
```

2. Para descrever o pod, no comando a seguir, em *pod-id*, use o ID dos pods encontrados na etapa anterior. Se você não estiver usando o namespace padrão, use `-n` *nameSpace*.

```
kubectl describe pod/pod-id
```

Para ver erros para o ASCP

- Para encontrar mais informações nos logs do provedor, no comando a seguir, em *pod-id*, use o ID do pod csi-secrets-store-provider-aws.

```
kubectl -n kube-system get pods
kubectl -n kube-system logs Pod/pod-id
```

- Verifique se a CRD do **`SecretProviderClass`** está instalada:

```
kubectl get crd secretproviderclasses.secrets-store.csi.x-k8s.io
```

Este comando deve retornar informações sobre a definição do recurso personalizado `SecretProviderClass`.

- Verifique se o objeto do SecretProviderClass foi criado.

```
kubectl get secretproviderclass SecretProviderClassName -o yaml
```

Exemplos de código do provedor de configuração e do AWS Secrets

Exemplos de autenticação do ASCP e controle de acesso

Exemplo: política do IAM que permite que o serviço da Identidade de Pods do Amazon EKS (pods.eks.amazonaws.com) assuma o perfil e marque a sessão:

JSON

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

```
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "pods.eks.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:TagSession"
      ],
      "Resource": "arn:aws:iam::111122223333:role/EKSPodIdentityRole"
    }
  ]
}
```

SecretProviderClass

Você usa o YAML para descrever quais parâmetros devem ser montados no Amazon EKS usando o ASCP. Para obter exemplos, consulte the section called “Uso do SecretProviderClass”.

Estrutura SecretProviderClass do YAML

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
   name: name
spec:
  provider: aws
  parameters:
    region:
    failoverRegion:
    pathTranslation:
    usePodIdentity:
    preferredAddressType:
    objects:
```

O campo de parâmetros contém os detalhes da solicitação de montagem:

região

(Opcional) A Região da AWS do parâmetro. Se você não usar esse campo, o ASCP procurará a Região a partir da anotação no nó. Essa pesquisa adiciona sobrecarga às solicitações de

montagem, portanto, recomendamos que você forneça a região para clusters que usam um grande número de pods.

Se você também especificar `failoverRegion`, o ASCP tentará recuperar o parâmetro de ambas as regiões. Se qualquer uma das regiões retornar um erro `4xx`, por exemplo, para um problema de autenticação, o ASCP não montará nenhum parâmetro. Se o parâmetro for recuperado com êxito de `region`, o ASCP montará o valor desse parâmetro. Se o parâmetro não for recuperado com êxito de `region`, mas for recuperado com êxito de `failoverRegion`, o ASCP montará o valor desse parâmetro.

`failoverRegion`

(Opcional) Se você incluir este campo, o ASCP tentará recuperar o parâmetro das regiões definidas em `region` e neste campo. Se qualquer uma das regiões retornar um erro `4xx`, por exemplo, para um problema de autenticação, o ASCP não montará nenhum parâmetro. Se o parâmetro for recuperado com êxito de `region`, o ASCP montará o valor desse parâmetro. Se o parâmetro não for recuperado com êxito de `region`, mas for recuperado com êxito de `failoverRegion`, o ASCP montará o valor desse parâmetro. Para obter um exemplo de como usar esse campo, consulte Failover de parâmetros multirregionais.

pathTranslation

(Opcional) Um só caractere de substituição para usar se o nome de arquivo no Amazon EKS vá conter o caractere separador de caminho, como barra (/) no Linux. O ASCP não poderá criar um arquivo montado que contenha um caractere separador de caminho. Em vez disso, o ASCP substituirá o caractere separador de caminho por outro caractere. Se você não usar esse campo, o caractere substituto será um sublinhado (_), portanto, por exemplo,`My/Path/Parameter` será montado como `My_Path_Parameter`.

Para impedir a substituição de caracteres, digite a string`False`.

`usePodIdentity`

(Opcional) Determina a abordagem de autenticação. Quando não especificado, o padrão é Perfis do IAM para contas de serviço (IRSA).

- Para usar a Identidade de Pods do EKS, use qualquer um destes valores: `"true"`, `"True"`, `"TRUE"`, `"t"` ou `"T"`.
- Para usar explicitamente o IRSA, defina qualquer um destes valores: `"false"`, `"False"`, `"FALSE"`, `"f"` ou `"F""=`.

preferredAddressType

(Opcional) Especifica o tipo de endereço IP preferencial para comunicação de endpoints do agente da Identidade de Pods. O campo só será aplicável ao usar o recurso de Identidade de Pods do EKS, e será ignorado ao usar perfis do IAM para contas de serviço. Os valores não diferenciam maiúsculas de minúsculas. Os valores válidos são:

- `"ipv4"`, `"IPv4"` ou `"IPV4"`: force o uso do endpoint IPv4 do agente da Identidade de Pods
- `"ipv6"`, `"IPv6"` ou `"IPV6"`: force o uso do endpoint IPv6 do agente da Identidade de Pods
- não especificado: use a seleção automática de endpoints, testando primeiro o endpoint IPv4 e voltando para o endpoint IPv6, se o IPv4 falhar

objects

Uma string contendo uma declaração do YAML dos segredos a serem montados. Recomendamos o uso de uma string com várias linhas ou um caractere pipe (|) no YAML.

objectName

Obrigatório. Especifica o nome do parâmetro ou segredo a ser buscado. Para Parameter Store, este é o `Name` do parâmetro e pode ser o nome ou o ARN completo do parâmetro. Para o Secrets Manager, este é o parâmetro `SecretId` e pode ser o nome amigável ou o ARN completo do segredo.

objectType

Necessário se você não usar um ARN do Secrets Manager para`objectName`. Para Parameter Store, use `ssmparameter`. Para o Secrets Manager, use `secretsmanager`.

objectAlias

(Opcional) O nome do arquivo do segredo no pod do Amazon EKS. Se você não especificar esse campo, o `objectName` aparece como o nome do arquivo.

objectVersion

(Opcional) O ID da versão do parâmetro. Não recomendado porque você deverá atualizar o ID da versão sempre que atualizar o parâmetro. Por padrão, a versão mais recente é usada. Se você incluir um `failoverRegion`, esse campo representará o `objectVersion` primário.

objectVersionLabel

(Opcional) O alias da versão. O padrão é a versão mais recente `AWSCURRENT`. Se você incluir um `failoverRegion`, esse campo representará o `objectVersionLabel` primário.

jmesPath

(Opcional) Um mapa das chaves no parâmetro para os arquivos a serem montados no Amazon EKS. Para usar esse campo, o valor do parâmetro deve estar no formato JSON.

O exemplo a seguir mostra a aparência de um parâmetro codificado em JSON.

```
{
    "username" : "myusername",
    "password" : "mypassword"
}
```

As chaves sãousernameepassword. O valor associado ausernameémyusername, e o valor associado apasswordémypassword.

Se você usar esse campo, deverá incluir os subcampos `path` e `objectAlias`.

caminho

Uma chave de um par de chave/valor no JSON do valor do parâmetro. Se o campo contiver um hífen, use aspas simples para delimitá-lo, por exemplo: `path: '"hyphenated-path"'`

objectAlias

O nome do arquivo a ser montado no pod Amazon EKS. Se o campo contiver um hífen, use aspas simples para delimitá-lo, por exemplo: `objectAlias: '"hyphenated-alias"'`

failoverObject

(Opcional) Se você especificar este campo, o ASCP tentará recuperar o parâmetro especificado no `objectName` primário e o parâmetro especificado no subcampo `objectName` de `failoverObject`. Se qualquer um retornar um erro `4xx`, por exemplo, para um problema de autenticação, o ASCP não montará nenhum parâmetro. Se o parâmetro for recuperado com êxito do `objectName` primário, o ASCP montará o valor desse parâmetro. Se o parâmetro não for recuperado com êxito do `objectName` primário, mas for recuperado com êxito do `objectName` do failover, o ASCP montará o valor desse parâmetro. Se incluir esse campo, você deverá incluir o campo `objectAlias`. Para obter um exemplo de como usar esse campo, consulte Failover para um parâmetro diferente.

Normalmente, você usa esse campo quando o parâmetro de failover não é uma réplica. Para obter um exemplo de como especificar uma réplica, consulte Failover de parâmetros multirregionais.

objectName

O nome ou o ARN completo do parâmetro de failover. Se você usar um ARN, a região no ARN deverá corresponder ao campo `failoverRegion`.

objectVersion

(Opcional) O ID da versão do parâmetro. Deve corresponder ao `objectVersion` primário. Não recomendado porque você deverá atualizar o ID da versão sempre que atualizar o parâmetro. Por padrão, a versão mais recente é usada.

objectVersionLabel

(Opcional) O alias da versão. O padrão é a versão mais recente `AWSCURRENT`.

Crie uma configuração básica de SecretProviderClass para montar parâmetros nos pods do Amazon EKS.

Pod Identity

SecretProviderClass para usar um parâmetro no mesmo cluster do Amazon EKS:

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: aws-parameter-store
spec:
  provider: aws
  parameters:
    objects: |
      - objectName: "MyParameter"
        objectType: "ssmparameter"
    usePodIdentity: "true"
```

IRSA

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
```

```
metadata:
  name: deployment-aws-parameter
spec:
  provider: aws
  parameters:
    objects: |
        - objectName: "MyParameter"
          objectType: "ssmparameter"
```

Uso do SecretProviderClass

Use estes exemplos para criar configurações de `SecretProviderClass` para diferentes cenários.

Exemplo: montar parâmetros por nome ou ARN

Este exemplo mostra como montar três tipos diferentes de parâmetros:

- Um parâmetro especificado por ARN completo
- Um parâmetro especificado por nome
- Uma versão de parâmetro de um segredo

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: aws-parameters
spec:
  provider: aws
  parameters:
    objects: |
      - objectName: "arn:aws:ssm:us-east-2:777788889999:parameter:MyParameter2-d4e5f6"
      - objectName: "MyParameter3"
        objectType: "ssmparameter"
      - objectName: "MyParameter4"
        objectType: "ssmparameter"
        objectVersionLabel: "AWSCURRENT"
```

Exemplo: montar pares de chave/valor de um parâmetro

Este exemplo mostra como montar pares de chave-valor específicos de um parâmetro formatado em JSON:

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: aws-parameters
spec:
  provider: aws
  parameters:
    objects: |
      - objectName: "arn:aws:ssm:us-east-2:777788889999:parameter:MyParameter-a1b2c3"
        jmesPath:
            - path: username
              objectAlias: dbusername
            - path: password
              objectAlias: dbpassword
```

Exemplo: exemplos de configuração de failover

Estes exemplos mostram como configurar o failover para parâmetros.

Failover de parâmetros multirregionais

Este exemplo mostra como configurar o failover automático para um parâmetro replicado em várias regiões:

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: aws-parameters
spec:
  provider: aws
  parameters:
    region: us-east-1
    failoverRegion: us-east-2
    objects: |
      - objectName: "MyParameter"
```

Failover para um parâmetro diferente

Este exemplo mostra como configurar o failover para um parâmetro diferente (não uma réplica):

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
```

```
  name: aws-parameters
spec:
  provider: aws
  parameters:
    region: us-east-1
    failoverRegion: us-east-2
    objects: |
      - objectName: "arn:aws:ssm:us-east-1:777788889999:parameter:MyParameter-a1b2c3"
        objectAlias: "MyMountedParameter"
        failoverObject:
          - objectName: "arn:aws:ssm:us-
east-2:777788889999:parameter:MyFailoverParameter-d4e5f6"
```

Recursos adicionais

Para obter mais informações sobre o uso do ASCP com o Amazon EKS, consulte os seguintes recursos:

- Uso da Identidade de Pods com o Amazon EKS
- Driver CSI do AWS Secrets Store no GitHub

Usar parâmetros do Parameter Store em funções do AWS Lambda

O Parameter Store, uma ferramenta do AWS Systems Manager, oferece armazenamento hierárquico seguro para gerenciamento de dados de configuração e gerenciamento de segredos. Você pode armazenar dados, como senhas, strings de banco de dados, IDs de Amazon Machine Image (AMI) e códigos de licença como valores de parâmetro.

Para usar parâmetros do Parameter Store em funções do AWS Lambda sem usar um SDK, você pode usar a extensão do Lambda para parâmetros e segredos da AWS. Essa extensão recupera os valores dos parâmetros e os armazena em cache para uso futuro. O uso da extensão do Lambda pode reduzir custos diminuindo o número de chamadas de API para o Parameter Store. O uso da extensão também pode melhorar a latência, pois recuperar um parâmetro do cache é mais rápido do que recuperá-lo do Parameter Store.

Uma extensão do Lambda corresponde a um processo complementar que aumenta os recursos de uma função do Lambda. Uma extensão é como um cliente que é executado em paralelo a uma invocação do Lambda. Esse cliente paralelo pode interagir com a função a qualquer momento durante o seu ciclo de vida. Para obter mais informações sobre extensões do Lambda, consulte API de extensões do Lambda no Guia do desenvolvedor do AWS Lambda.

A extensão do Lambda para parâmetros e segredos da AWS funciona tanto para o Parameter Store quanto para o AWS Secrets Manager. Para saber como usar a extensão do Lambda para segredos do Secrets Manager, consulte Usar segredos do AWS Secrets Manager em funções do AWS Lambda no Guia do usuário do AWS Secrets Manager.

Informações relacionadas

Usando a extensão do Lambda Parameter and Secrets da AWS para armazenar em cache parâmetros e segredos (Blog de computação da AWS)

Como a extensão funciona

Para usar parâmetros em uma função do Lambda sem a extensão do Lambda, você deve configurar a função do Lambda para receber atualizações de configuração por meio da integração com a ação da API `GetParameter` para o Parameter Store.

Quando você usa a extensão do Lambda para parâmetros e segredos da AWS, a extensão recupera o valor do parâmetro do Parameter Store e o armazena no cache local. Em seguida, o valor em cache é usado para outras invocações até expirar. Os valores em cache expiram quando seu tempo de vida (TTL) termina. Você pode configurar o valor de TTL usando a variável de ambiente `SSM_PARAMETER_STORE_TTL`, conforme explicado mais adiante neste tópico.

Se o TTL do cache configurado não tiver expirado, o valor do parâmetro em cache será usado. Se o tempo tiver expirado, o valor em cache será invalidado e o valor do parâmetro será recuperado do Parameter Store.

Além disso, o sistema detecta os valores de parâmetros que são usados com frequência e os mantém no cache, e limpa os que expiraram ou que não foram usados.

> **Important**
>
> A extensão pode ser invocada somente na fase `INVOKE` da operação do Lambda, e não durante a fase `INIT`.

Detalhes da implantação

Use os detalhes a seguir para ajudar você a configurar a extensão do Lambda para parâmetros e segredos da AWS.

Autenticação

Para autorizar e autenticar solicitações do Parameter Store, a extensão usa as mesmas credenciais usadas para executar a própria função do Lambda. Portanto, o perfil do AWS Identity and Access Management (IAM) usada para executar a função deve ter as seguintes permissões para interagir com o Parameter Store:

- `ssm:GetParameter`: necessário para recuperar parâmetros do Parameter Store
- `kms:Decrypt`: necessário se você estiver recuperando parâmetros `SecureString` do Parameter Store

Para obter mais informações, consulte perfil do IAM para execução do AWS Lambda no Guia do desenvolvedor do AWS Lambda.

Instanciação

O Lambda cria instâncias separadas correspondentes ao nível de simultaneidade requerido por sua função. Cada instância é isolada e mantém o próprio cache local dos dados de configuração. Para obter mais informações sobre instâncias do Lambda e simultaneidade, consulte Configurar a simultaneidade reservada no Guia do desenvolvedor do AWS Lambda.

Nenhuma dependência de SDK

A extensão do Lambda para parâmetros e segredos da AWS funciona independentemente de qualquer biblioteca de linguagem do AWS SDK. Não é necessário um AWS SDK para fazer solicitações GET ao Parameter Store.

Porta do Localhost

Use o `localhost` nas solicitações GET. A extensão faz solicitações para a porta 2773 do localhost. Não é necessário especificar um endpoint externo ou interno para usar a extensão. É possível configurar a porta ao definir a variável de ambiente `PARAMETERS_SECRETS_EXTENSION_HTTP_PORT`.

Por exemplo, em Python, GET URL pode ser semelhante ao exemplo a seguir.

```
parameter_url = ('http://localhost:' + port + '/systemsmanager/parameters/get/?
name=' + ssm_parameter_path)
```

Alterações no valor de um parâmetro antes da expiração do TTL

A extensão não detecta alterações no valor do parâmetro e não realiza uma atualização automática antes que o TTL expire. Se você alterar o valor de um parâmetro, as operações que

usam o valor do parâmetro em cache poderão falhar até que o cache seja atualizado da próxima vez. Se você espera alterações frequentes no valor de um parâmetro, recomendamos que defina um valor de TTL mais curto.

Exigência de cabeçalho

Para recuperar parâmetros do cache da extensão, o cabeçalho da solicitação GET deve incluir uma referência a `X-Aws-Parameters-Secrets-Token`. Defina o token como `AWS_SESSION_TOKEN`, fornecido pelo Lambda para todas as funções em execução. O uso desse cabeçalho indica que o chamador está no ambiente do Lambda.

Exemplo

O exemplo a seguir em Python demonstra uma solicitação básica para recuperar o valor de um parâmetro em cache.

```
import urllib.request
import os
import json

aws_session_token = os.environ.get('AWS_SESSION_TOKEN')

def lambda_handler(event, context):
    # Retrieve /my/parameter from Parameter Store using extension cache
    req = urllib.request.Request('http://localhost:2773/systemsmanager/parameters/
get?name=%2Fmy%2Fparameter')
    req.add_header('X-Aws-Parameters-Secrets-Token', aws_session_token)
    config = urllib.request.urlopen(req).read()

    return json.loads(config)
```

Suporte ao ARM

A extensão oferece suporte à arquitetura ARM na maioria das Regiões da AWS em que há suporte às arquiteturas x86_64 e x86. Se estiver usando a arquitetura ARM, sugerimos verificar se sua arquitetura é compatível. Para obter listas completas de ARNs de extensão, consulte ARNs da extensão do Lambda para parâmetros e segredos da AWS.

Registro em log

O Lambda registra informações de execução sobre a extensão em conjunto com a função usando o Amazon CloudWatch Logs. Por padrão, a extensão registra uma quantidade mínima de

informações no CloudWatch. Para registrar mais detalhes em log, defina a variável de ambiente `PARAMETERS_SECRETS_EXTENSION_LOG_LEVEL` para `DEBUG`.

Adicionar a extensão a uma função do Lambda

Para usar a extensão do Lambda para parâmetros e segredos da AWS, adicione a extensão à função Lambda como uma camada.

Use um dos métodos a seguir para adicionar a extensão à função.

AWS Management Console (Opção de adicionar camada)

1. Abra o console AWS Lambda em https://console.aws.amazon.com/lambda/.
2. Escolha a função. Na área Layers (Camadas), escolha Add a layer. (Adicionar uma camada).
3. Na área Escolher uma camada, escolha a opção Camadas da AWS.
4. Em Camadas da AWS, escolha AWS-Parameters-and-Secrets-Lambda-Extension, escolha uma versão e depois escolha Adicionar.

AWS Management Console (Opção de especificar um ARN)

1. Abra o console AWS Lambda em https://console.aws.amazon.com/lambda/.
2. Escolha a função. Na área Layers (Camadas), escolha Add a layer. (Adicionar uma camada).
3. Na área Choose a layer (Escolher uma camada), escolha a opção Specify an ARN (Especificar um ARN).
4. Em Specify an ARN (Especificar um ARN), insira o ARN da extensão para a Região da AWS e a arquitetura e depois escolha Add (Adicionar).

AWS Command Line Interface

Execute o comando a seguir na AWS CLI. Substitua cada *espaço reservado para recurso de exemplo* por suas próprias informações.

```
aws lambda update-function-configuration \
    --function-name function-name \
    --layers layer-ARN
```

Informações relacionadas

Usar camadas com a função do Lambda

Configurar extensões (arquivamento de arquivo.zip)

Variáveis de ambiente da extensão do Lambda para parâmetros e segredos da AWS

Você pode configurar a extensão alterando as variáveis de ambiente a seguir. Para ver as configurações atuais, defina `PARAMETERS_SECRETS_EXTENSION_LOG_LEVEL` como `DEBUG`. Para obter mais informações, consulte Usar de variáveis de ambiente do AWS Lambda no Guia do desenvolvedor do AWS Lambda.

> **Note**
>
> O AWS Lambda registra os detalhes operacionais da extensão do Lambda e a função do Lambda no Amazon CloudWatch Logs.

Variável de ambiente	Detalhes	Obrigatório	Valores válidos	Valor padrão
`SSM_PARAMETER_STORE_TIMEOUT_MILLIS`	Tempo limite, em milissegundos, para solicitações ao Parameter Store. Um valor 0 (zero) indica que não há tempo limite.	Não	Todos os números inteiros	0 (zero)
`SECRETS_MANAGER_TIMEOUT_MILLIS`	Tempo limite, em milissegundos, para solicitações ao Secrets Manager. Um valor 0 (zero) indica que	Não	Todos os números inteiros	0 (zero)

Variável de ambiente	Detalhes	Obrigatório	Valores válidos	Valor padrão
	não há tempo limite.			
SSM_PARAMETER_STORE_TTL	Vida útil máxima válida, em segundos, de um parâmetro no cache antes de ser invalidado. Um valor de 0 (zero) indica que o cache deve ser ignorado. Essa variável será ignorada se o valor de PARAMETERS_SECRETS_EXTENSION_CACHE_SIZE for 0 (zero).	Não	0 (zero) a 300 s (5 minutos)	300 s (5 minutos)

Variável de ambiente	Detalhes	Obrigatório	Valores válidos	Valor padrão
SECRETS_MANAGER_TTL	Vida útil máxima válida, em segundos, de um segredo no cache antes de ser invalidado. Um valor de 0 (zero) indica que o cache é ignorado. Essa variável será ignorada se o valor de PARAMETERS_SECRETS_EXTENSION_CACHE_SIZE for 0 (zero).	Não	0 (zero) a 300 s (5 minutos)	300 s (5 minutos)
PARAMETERS_SECRETS_EXTENSION_CACHE_ENABLED	Determina se o cache para a extensão está habilitado. Valor: TRUE | FALSE	Não	TRUE | FALSE	VERDADEIRO

Variável de ambiente	Detalhes	Obrigatório	Valores válidos	Valor padrão
PARAMETERS_SECRETS_EXTENSION_CACHE_SIZE	O tamanho máximo do cache em termos de número de itens. Um valor de 0 (zero) indica que o cache é ignorado. Essa variável será ignorada se os dois valores de TTL do cache forem 0 (zero).	Não	0 (zero) a 1000	1000
PARAMETERS_SECRETS_EXTENSION_HTTP_PORT	A porta para o servidor HTTP local.	Não	1 - 65535	2773

Variável de ambiente	Detalhes	Obrigatório	Valores válidos	Valor padrão
`PARAMETERS_SECRETS_EXTENSION_MAX_CONNECTIONS`	Número máximo de conexões para os clientes HTTP que a extensão usa para fazer solicitações ao Parameter Store ou ao Secrets Manager. Essa é uma configuração por cliente do número de conexões que o cliente do Secrets Manager e o cliente do Parameter Store fazem aos serviços de backend.	Não	Mínimo de 1, sem limite máximo.	3

Variável de ambiente	Detalhes	Obrigatório	Valores válidos	Valor padrão
`PARAMETERS_SECRETS_EXTENSION_LOG_LEVEL`	O nível de detalhes relatado nos logs da extensão. Recomendamos usar `DEBUG` para obter o máximo de detalhes sobre a configuração de cache ao configurar e testar a extensão. Os logs das operações do Lambda são automaticamente enviados para um grupo de logs associado do CloudWatch Logs.	Não	`DEBUG | WARN | ERROR | NONE | INFO`	`INFO`

Exemplos de comandos para usar o Parameter Store do AWS Systems Manager e a extensão do AWS Secrets Manager

Os exemplos nesta seção demonstram ações de API para uso com o Parameter Store do AWS Systems Manager e a extensão do AWS Secrets Manager.

Exemplos de comandos para Parameter Store

A extensão do Lambda usa acesso somente leitura à ação da API GetParameter.

Para chamar essa ação, faça uma chamada HTTP GET semelhante à que se segue. Esse formato de comando fornece acesso aos parâmetros na camada de parâmetros padrão.

```
GET http://localhost:port/systemsmanager/parameters/get?name=parameter-
name&version=version&label=label&withDecryption={true|false}
```

Neste exemplo, *parameter-name* representa o nome completo do parâmetro, como `MyParameter`, para um parâmetro que não está em uma hierarquia, ou `%2FDev%2FProduction%2FEast %2FProject-ABC%2FMyParameter` para um parâmetro chamado `/Dev/Production/East/ Project-ABC/MyParameter` que faz parte de uma hierarquia.

> **Note**
>
> Ao usar chamadas GET, os valores dos parâmetros devem ser codificados em HTTP para preservar os caracteres especiais. Por exemplo, em vez de formatar um caminho hierárquico com `/a/b/c`, codifique caracteres que possam ser interpretados como parte da URL, como`%2Fa%2Fb%2Fc`.

version e *label* são os seletores disponíveis para uso com a ação `GetParameter`.

```
GET http://localhost:port/systemsmanager/parameters/get/?name=MyParameter&version=5
```

Para chamar um parâmetro em uma hierarquia, faça uma chamada HTTP GET semelhante à que se segue.

```
GET http://localhost:port/systemsmanager/parameters/get?name=%2Fa%2Fb%2F&label=release
```

Para chamar um parâmetro público (global), faça uma chamada HTTP GET semelhante à que se segue.

```
GET http://localhost:port/systemsmanager/parameters/get/?name=%2Faws%2Fservice%20list
%2F…
```

Para fazer uma chamada HTTP GET para um segredo do Secrets Manager usando referências ao Parameter Store, faça uma chamada HTTP GET semelhante à que se segue.

```
GET http://localhost:port/systemsmanager/parameters/get?name=%2Faws%2Freference
%2Fsecretsmanager%2F…
```

Para fazer uma chamada usando o nome do recurso da Amazon (ARN) como parâmetro, faça uma chamada HTTP GET semelhante à que se segue.

```
GET http://localhost:port/systemsmanager/parameters/get?name=arn:aws:ssm:us-
east-1:123456789012:parameter/MyParameter
```

Para fazer uma chamada que acesse um parâmetro `SecureString` com descriptografia, faça uma chamada HTTP GET semelhante à que se segue.

```
GET http://localhost:port/systemsmanager/parameters/get?
name=MyParameter&withDecryption=true
```

Você pode especificar que os parâmetros não sejam descriptografados omitindo `withDecryption` ou definindo-o explicitamente como `false`. Você também pode especificar uma versão ou um rótulo, mas não ambos. Se o fizer, somente o primeiro deles que for colocado após o ponto de interrogação (?) na URL será usado.

ARNs da extensão do Lambda para parâmetros e segredos da AWS

As tabelas a seguir fornecem ARNs de extensão para as arquiteturas e regiões compatíveis.

Tópicos

- ARNs de extensão para as arquiteturas x86_64 e x86
- ARNs de extensão para arquiteturas ARM64 e Mac with Apple silicon

ARNs de extensão para as arquiteturas x86_64 e x86

Última atualização: 26 de junho de 2025

Região	ARN
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:590474943231:layer:AWS-Parame`

Região	ARN
	ters-and-Secrets-Lambda-Ext ension:22
Leste dos EUA (Norte da Virgínia)	arn:aws:lambda:us-east-1:17 7933569100:layer:AWS-Parame ters-and-Secrets-Lambda-Ext ension:18
Oeste dos EUA (Norte da Califórnia)	arn:aws:lambda:us-west-1:99 7803712105:layer:AWS-Parame ters-and-Secrets-Lambda-Ext ension:18
Oeste dos EUA (Oregon)	arn:aws:lambda:us-west-2:34 5057560386:layer:AWS-Parame ters-and-Secrets-Lambda-Ext ension:18
África (Cidade do Cabo)	arn:aws:lambda:af-south-1:3 17013901791:layer:AWS-Param eters-and-Secrets-Lambda-Ex tension:18
Ásia Pacífico (Hong Kong)	arn:aws:lambda:ap-east-1:76 8336418462:layer:AWS-Parame ters-and-Secrets-Lambda-Ext ension:20
Ásia-Pacífico (Taipei)	arn:aws:lambda:ap-east-2:89 0742577149:layer:AWS-Parame ters-and-Secrets-Lambda-Ext ension:6

Região	ARN
Ásia-Pacífico (Haiderabade)	arn:aws:lambda:ap-south-2:070087711984:layer:AWS-Parameters-and-Secrets-Lambda-Extension:15
Ásia-Pacífico (Jacarta)	arn:aws:lambda:ap-southeast-3:490737872127:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18
Ásia-Pacífico (Melbourne)	arn:aws:lambda:ap-southeast-4:090732460067:layer:AWS-Parameters-and-Secrets-Lambda-Extension:8
Ásia-Pacífico (Malásia)	arn:aws:lambda:ap-southeast-5:381492012281:layer:AWS-Parameters-and-Secrets-Lambda-Extension:7
Ásia-Pacífico (Mumbai)	arn:aws:lambda:ap-south-1:176022468876:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18
Ásia-Pacífico (Osaka)	arn:aws:lambda:ap-northeast-3:576959938190:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18
Ásia-Pacífico (Seul)	arn:aws:lambda:ap-northeast-2:738900069198:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18

Região	ARN
Ásia-Pacífico (Singapura)	`arn:aws:lambda:ap-southeast-1:044395824272:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18`
Ásia-Pacífico (Sydney)	`arn:aws:lambda:ap-southeast-2:665172237481:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18`
Ásia-Pacífico (Tailândia)	`arn:aws:lambda:ap-southeast-7:941377119484:layer:AWS-Parameters-and-Secrets-Lambda-Extension:6`
Ásia-Pacífico (Tóquio)	`arn:aws:lambda:ap-northeast-1:133490724326:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18`
Canadá (Central)	`arn:aws:lambda:ca-central-1:200266452380:layer:AWS-Parameters-and-Secrets-Lambda-Extension:20`
Oeste do Canadá (Calgary)	`arn:aws:lambda:ca-west-1:243964427225:layer:AWS-Parameters-and-Secrets-Lambda-Extension:8`

Região	ARN
China (Pequim)	`arn:aws-cn:lambda:cn-north-1:287114880934:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18arn:aws-cn:lambda:cn-north-1:287114880934:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18`
China (Ningxia)	`arn:aws-cn:lambda:cn-northwest-1:287310001119:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18arn:aws-cn:lambda:cn-northwest-1:287310001119:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18`
Europa (Frankfurt)	`arn:aws:lambda:eu-central-1:187925254637:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:015030872274:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18`
Europa (Londres)	`arn:aws:lambda:eu-west-2:133256977650:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18`
Europa (Milão)	`arn:aws:lambda:eu-south-1:325218067255:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18`

Região	ARN
Europe (Paris)	`arn:aws:lambda:eu-west-3:780235371811:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18`
Região Europa (Espanha)	`arn:aws:lambda:eu-south-2:524103009944:layer:AWS-Parameters-and-Secrets-Lambda-Extension:17`
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:427196147048:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18`
Israel (Tel Aviv)	`arn:aws:lambda:il-central-1:148806536434:layer:AWS-Parameters-and-Secrets-Lambda-Extension:8`
Região Europa (Zurique)	`arn:aws:lambda:eu-central-2:772501565639:layer:AWS-Parameters-and-Secrets-Lambda-Extension:15`
Região México (Centro)	`arn:aws:lambda:mx-central-1:241533131596:layer:AWS-Parameters-and-Secrets-Lambda-Extension:5`
Oriente Médio (Bahrein)	`arn:aws:lambda:me-south-1:832021897121:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18`

Região	ARN
Oriente Médio (Emirados Árabes Unidos)	`arn:aws:lambda:me-central-1:858974508948:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18`
South America (São Paulo)	`arn:aws:lambda:sa-east-1:933737806257:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18`
AWS GovCloud (Leste dos EUA)	`arn:aws-us-gov:lambda:us-gov-east-1:129776340158:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18arn:aws-us-gov:lambda:us-gov-east-1:129776340158:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18`
AWS GovCloud (Oeste dos EUA)	`arn:aws-us-gov:lambda:us-gov-west-1:127562683043:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18arn:aws-us-gov:lambda:us-gov-west-1:127562683043:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18`

ARNs de extensão para arquiteturas ARM64 e Mac with Apple silicon

Última atualização: 26 de junho de 2025

Região	ARN
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:590474943231:layer:AWS-Parame`

Região	ARN
	ters-and-Secrets-Lambda-Ext ension-Arm64:22
Leste dos EUA (Norte da Virgínia)	arn:aws:lambda:us-east-1:17 7933569100:layer:AWS-Parame ters-and-Secrets-Lambda-Ext ension-Arm64:18
Região Oeste dos EUA (Norte da Califórnia)	arn:aws:lambda:us-west-1:99 7803712105:layer:AWS-Parame ters-and-Secrets-Lambda-Ext ension-Arm64:15
Oeste dos EUA (Oregon)	arn:aws:lambda:us-west-2:34 5057560386:layer:AWS-Parame ters-and-Secrets-Lambda-Ext ension-Arm64:18
Região África (Cidade do Cabo)	arn:aws:lambda:af-south-1:3 17013901791:layer:AWS-Param eters-and-Secrets-Lambda-Ex tension-Arm64:15
Região Ásia-Pacífico (Hong Kong)	arn:aws:lambda:ap-east-1:76 8336418462:layer:AWS-Parame ters-and-Secrets-Lambda-Ext ension-Arm64:17
Ásia-Pacífico (Taipei)	arn:aws:lambda:ap-east-2:89 0742577149:layer:AWS-Parame ters-and-Secrets-Lambda-Ext ension-Arm64:2

Região	ARN
Ásia-Pacífico (Haiderabade)	arn:aws:lambda:ap-south-2:070087711984:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:7
Região Ásia-Pacífico (Jacarta)	arn:aws:lambda:ap-southeast-3:490737872127:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:15
Ásia-Pacífico (Melbourne)	arn:aws:lambda:ap-southeast-4:090732460067:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:7
Ásia-Pacífico (Malásia)	arn:aws:lambda:ap-southeast-5:381492012281:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:7
Ásia-Pacífico (Mumbai)	arn:aws:lambda:ap-south-1:176022468876:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:18
Ásia-Pacífico (Osaka)	arn:aws:lambda:ap-northeast-3:576959938190:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:15
Região Ásia-Pacífico (Seul)	arn:aws:lambda:ap-northeast-2:738900069198:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:15

Região	ARN
Ásia-Pacífico (Singapura)	`arn:aws:lambda:ap-southeast-1:044395824272:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:18`
Ásia-Pacífico (Sydney)	`arn:aws:lambda:ap-southeast-2:665172237481:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:18`
Ásia-Pacífico (Tailândia)	`arn:aws:lambda:ap-southeast-7:941377119484:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:5`
Ásia-Pacífico (Tóquio)	`arn:aws:lambda:ap-northeast-1:133490724326:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:18`
Região Canadá (Central)	`arn:aws:lambda:ca-central-1:200266452380:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:17`
Oeste do Canadá (Calgary)	`arn:aws:lambda:ca-west-1:243964427225:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:7`

Região	ARN
China (Pequim)	arn:aws-cn:lambda:cn-north-1:287114880934:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:7arn:aws-cn:lambda:cn-north-1:287114880934:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:7
China (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:287310001119:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:7arn:aws-cn:lambda:cn-northwest-1:287310001119:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:7
Europa (Frankfurt)	arn:aws:lambda:eu-central-1:187925254637:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:18
Europa (Irlanda)	arn:aws:lambda:eu-west-1:015030872274:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:18
Europa (Londres)	arn:aws:lambda:eu-west-2:133256977650:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:18

Região	ARN
Região Europa (Milão)	arn:aws:lambda:eu-south-1:325218067255:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:15
Região Europa (Paris)	arn:aws:lambda:eu-west-3:780235371811:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:15
Região Europa (Espanha)	arn:aws:lambda:eu-south-2:524103009944:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:9
Região Europa (Estocolmo)	arn:aws:lambda:eu-north-1:427196147048:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:15
Israel (Tel Aviv)	arn:aws:lambda:il-central-1:148806536434:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:7
Região Europa (Zurique)	arn:aws:lambda:eu-central-2:772501565639:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:7
Região México (Centro)	arn:aws:lambda:mx-central-1:241533131596:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:4

Região	ARN
Região Oriente Médio (Bahrein)	`arn:aws:lambda:me-south-1:832021897121:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:15`
Oriente Médio (Emirados Árabes Unidos)	`arn:aws:lambda:me-central-1:858974508948:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:7`
Região América do Sul (São Paulo)	`arn:aws:lambda:sa-east-1:933737806257:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:15`
AWS GovCloud (Leste dos EUA)	`arn:aws-us-gov:lambda:us-gov-east-1:129776340158:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:7arn:aws-us-gov:lambda:us-gov-east-1:129776340158:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:7`
AWS GovCloud (Oeste dos EUA)	`arn:aws-us-gov:lambda:us-gov-west-1:127562683043:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:7arn:aws-us-gov:lambda:us-gov-west-1:127562683043:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:7`

Integração com outros produtos e serviços

O AWS Systems Manager tem integração interna para os produtos e serviços mostrados na tabela abaixo.

Ansible	O Ansible é uma plataforma de automação de TI que torna aplicações e sistemas mais fáceis de implantar. O Systems Manager fornece o documento do Systems Manager (documento SSM) `AWS-ApplyAnsiblePlaybooks` que permite a você criar associações do State Manager que executam manuais do Ansible. Saiba mais Criar associações que executam manuais do Ansible
Chef	O Chef é uma ferramenta de automação de TI que facilita a implantação de seus sistemas e aplicações. O Systems Manager fornece o documento do SSM `AWS-ApplyChefRecipes` que permite criar associações no State Manager, uma ferramenta do AWS Systems Manager que executa receitas do Chef. Saiba mais Criar associações que executam receitas do Chef O Systems Manager também se integra aos perfis do Chef InSpec, permitindo que você execute verificações de conformidade e visualize nós compatíveis e não compatíveis.

	Saiba mais Usar os perfis do Chef InSpec com o Systems Manager Compliance
GitHub	O GitHub fornece hospedagem para o controle de versão de desenvolvimento de software e colaboração. O Systems Manager fornece o documento do SSM `AWS-RunDocument` , que permite a você executar outros documentos do SSM armazenados no GitHub, e o documento do SSM `AWS-RunRemoteScript` , que permite executar scripts armazenados no GitHub. Saiba mais • Executar documentos do em locais remotos • Executar scripts do GitHub
Jenkins	O Jenkins é um servidor de automação de código aberto que permite que os desenvolv edores criem, testem e implantem software de forma confiável. O Automation, uma ferramenta do Systems Manager, pode ser usado como uma etapa pós-compilação para pré-instalar as versões da aplicação nas Amazon Machine Images (AMIs). Saiba mais Atualizar AMIs usando o Automation e Jenkins

ServiceNow	O ServiceNow é um sistema de gerenciamento de serviços corporativos que permite gerenciar seus serviços e operações de TI. Automation, Change Manager, Incident Manager e OpsCenter, todos ferrament as do Systems Manager, são integrados ao ServiceNow por meio do AWS Service Management Connector. Com essa integração, você pode visualizar, criar, atualizar, adicionar correspondência e resolver casos do AWS Support diretamente no ServiceNow. Saiba mais Integração com o ServiceNow

Terraform	O HashiCorp Terraform é uma ferramenta de software de infraestrutura como código (IaC) de código aberto que fornece um fluxo de trabalho de interface de linha de comandos (CLI) para gerenciar vários serviços em nuvem. No Systems Manager, você pode usar o Terraform para gerenciar ou provisionar o seguinte: Recursos • aws_ssm_activation • aws_ssm_association • aws_ssm_default_patch_baseline • aws_ssm_document • aws_ssm_maintenance_window • aws_ssm_maintenance_window_target • aws_ssm_maintenance_window_task • aws_ssm_parameter • aws_ssm_patch_baseline • aws_ssm_patch_group • aws_ssm_resource_data_sync • aws_ssm_service_setting Fontes de dados • aws_ssm_document • aws_ssm_instances • ssm_maintenance_windows • aws_ssm_parameter • aws_ssm_parameters_by_path • aws_ssm_patch_baseline

Tópicos

- Executar scripts do GitHub
- Usar os perfis do Chef InSpec com o Systems Manager Compliance
- Integração com o ServiceNow

Executar scripts do GitHub

Esta seção descreve como usar o documento do Systems Manager (documento SSM) predefinido `AWS-RunRemoteScript` para baixar scripts do GitHub, inclusive manuais do Ansible e scripts Python, Ruby e PowerShell. Usando esse documento do SSM, você não precisa mais modificar scripts manualmente no Amazon Elastic Compute Cloud (Amazon EC2) ou encapsulá-los em documentos do SSM. A integração do AWS Systems Manager com o GitHub promove a infraestrutura como código, o que reduz o tempo necessário para gerenciar nós ao padronizar configurações em toda a frota.

Você pode também criar documentos SSM personalizados que permitem baixar e executar scripts ou outros documentos SSM de locais remotos. Para ter mais informações, consulte Criar documentos compostos.

Você pode também fazer download de um diretório que inclua vários scripts. Ao executar o script principal no diretório, o Systems Manager executa também qualquer script referenciado que estiver incluído no diretório.

Observe os detalhes essenciais a seguir sobre a execução de scripts do GitHub.

- O Systems Manager não verifica se o script pode ser executado em um nó. Antes de baixar e executar o script, verifique se o software necessário está instalado em seu nó. Ou você pode criar um documento composto que instala o software usando o Run Command ou o State Manager, ambos ferramentas do AWS Systems Manager, e, em seguida, baixa e executa o script.
- Você é responsável por garantir que todos os requisitos do GitHub sejam atendidos. Isso inclui a atualização de seu token de acesso, conforme necessário. Tome cuidado para não ultrapassar o número de solicitações autenticadas ou não autenticadas. Para obter mais informações, consulte a documentação do GitHub.
- Não há suporte a repositórios do GitHub Enterprise.

Tópicos

- Executar manuais do Ansible via GitHub

- Executar scripts Python no GitHub

Executar manuais do Ansible via GitHub

Esta seção inclui procedimentos para ajudar você a executar manuais do Ansible via GitHub usando o console ou a AWS Command Line Interface (AWS CLI).

Antes de começar

Se você planeja executar um script armazenado em um repositório privado do GitHub, crie um parâmetro `SecureString` do AWS Systems Manager para o token de acesso de segurança do GitHub. Você não pode acessar um script em um repositório privado do GitHub transmitindo manualmente o token via SSH. O token de acesso deve ser passado como um parâmetro `SecureString` do Systems Manager. Para obter mais informações sobre como criar um parâmetro `SecureString`, consulte Criar parâmetros do Parameter Store no Systems Manager.

Executar um manual do Ansible via GitHub (console)

Executar um manual do Ansible via GitHub

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Run Command.
3. Selecione Run command.
4. Na lista Command document (Documento do comando), escolha **`AWS-RunRemoteScript`**.
5. Em Command parameters, faça o seguinte:

   - Em Tipo de origem, selecione GitHub.
   - Na caixa de texto Source Info (Informações da origem) insira as informações necessárias para acessar a origem no seguinte formato:

     ```
     {
       "owner": "owner_name",
       "repository": "repository_name",
       "getOptions": "branch:branch_name",
       "path": "path_to_scripts_or_directory",
       "tokenInfo": "{{ssm-secure:SecureString_parameter_name}}"
     }
     ```

     Este exemplo faz download de um arquivo chamado `webserver.yml`.

```
{
    "owner": "TestUser1",
    "repository": "GitHubPrivateTest",
    "getOptions": "branch:myBranch",
    "path": "scripts/webserver.yml",
    "tokenInfo": "{{ssm-secure:mySecureStringParameter}}"
}
```

> **Note**
>
> `"branch"` é necessário somente se o documento do SSM estiver armazenado em uma ramificação diferente de `master`.
> Para usar a versão de seus scripts que estão em uma confirmação específica no repositório, use `commitID` com `getOptions` em vez de `branch`. Por exemplo: `"getOptions": "commitID:bbc1ddb94...b76d3bEXAMPLE",`

- No campo Command Line (Linha de comando), digite os parâmetros para a execução do script. Aqui está um exemplo.

  `ansible-playbook -i “localhost,” --check -c local webserver.yml`

- (Opcional) No campo Working Directory (Diretório de trabalho), insira o nome de um diretório do nó em que você deseja baixar e executar o script.
- (Opcional) Em Execution Timeout, especifique o número de segundos para o sistema aguardar antes de a execução do comando de script falhar.

6. Na seção Targets (Destinos), escolha os nós gerenciados nos quais você quer executar essa operação, especificando as etiquetas, selecionando as instâncias ou dispositivos de borda manualmente ou especificando um grupo de recursos.

> **Tip**
>
> Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

7. Para Other parameters (Outros parâmetros):

   - Em Comment (Comentário), digite as informações sobre esse comando.

- Em Timeout (seconds) (Tempo limite [segundos]), especifique o número de segundos para o sistema aguardar até a falha de execução do comando total.

8. Para Rate control (Controle de taxa):

   - Em Concurrency (Concorrência), especifique um número ou uma porcentagem de nós gerenciados nos quais executar o comando ao mesmo tempo.

     > **Note**
     >
     > Se você selecionou destinos especificando tags aplicadas a instâncias a nós gerenciados ou especificando grupos de recursos da AWS, e não tiver certeza de quantas instâncias são direcionadas, restrinja o número de instâncias que poderão executar o documento ao mesmo tempo, especificando uma porcentagem.

   - Em Error threshold (Limite de erro), especifique quando parar de executar o comando em outros nós depois de falhar em alguns ou em uma porcentagem de nós. Por exemplo, se você especificar três erros, o Systems Manager deixará de enviar o comando quando o 4° erro for recebido. Os nós gerenciados que continuam processando o comando também podem enviar erros.

9. (Opcional) Em Output options (Opções de saída), para salvar a saída do comando em um arquivo, selecione a caixa Write command output to an S3 bucket (Gravar saída do comando em um bucket do S3). Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

   > **Note**
   >
   > As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil de instância (para instâncias do EC2) ou perfil de serviço do IAM (máquinas ativadas para ambientes híbridos) atribuído à instância, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço do IAM associado ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

10. Na seção SNS notifications (Notificações do SNS), se quiser enviar notificações sobre o status da execução do comando, marque a caixa de seleção Enable SNS notifications (Habilitar notificações do SNS).

    Para obter mais informações sobre a configuração de notificações do Amazon SNS para o Run Command, consulte Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS.
11. Escolha Executar.

Executar um manual do Ansible no GitHub usando a AWS CLI

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.
2. Execute o comando a seguir para baixar e executar um script do GitHub.

```
aws ssm send-command \
    --document-name "AWS-RunRemoteScript" \
    --instance-ids "instance-IDs"\
    --parameters '{"sourceType":["GitHub"],"sourceInfo":["{\"owner
\":\"owner_name\", \"repository\": \"repository_name\", \"path
\": \"path_to_file_or_directory\", \"tokenInfo\":\"{{ssm-
secure:name_of_your_SecureString_parameter}}\" }"],"commandLine":
["commands_to_run"]}'
```

   Veja a seguir um comando de exemplo a ser executado em uma máquina Linux local.

```
aws ssm send-command \
    --document-name "AWS-RunRemoteScript" \
    --instance-ids "i-02573cafcfEXAMPLE" \
    --parameters '{"sourceType":["GitHub"],"sourceInfo":["{\"owner\":\"TestUser1\",
 \"repository\": \"GitHubPrivateTest\", \"path\": \"scripts/webserver.yml\",
 \"tokenInfo\":\"{{ssm-secure:mySecureStringParameter}}\" }"],"commandLine":
["ansible-playbook -i “localhost,” --check -c local webserver.yml"]}'
```

Executar scripts Python no GitHub

Esta seção inclui procedimentos para ajudar você a executar scripts do Python no GitHub usando o console do AWS Systems Manager ou a AWS Command Line Interface (AWS CLI).

Executar um script do Python no GitHub (console)

Executar um script do Python no GitHub

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Run Command.
3. Selecione Run command.
4. Na lista Command document (Documento do comando), escolha **`AWS-RunRemoteScript`**.
5. Em Command parameters (Parâmetros do comando), faça o seguinte:
   - Em Tipo de origem, selecione GitHub.
   - Na caixa de texto Source Info (Informações da origem) insira as informações necessárias para acessar a origem no seguinte formato:

```
{
    "owner": "owner_name",
    "repository": "repository_name",
    "getOptions": "branch:branch_name",
    "path": "path_to_document",
    "tokenInfo": "{{ssm-secure:SecureString_parameter_name}}"
}
```

O exemplo a seguir baixa um diretório de scripts chamado complex-script.

```
{
    "owner": "TestUser1",
    "repository": "SSMTestDocsRepo",
    "getOptions": "branch:myBranch",
    "path": "scripts/python/complex-script",
    "tokenInfo": "{{ssm-secure:myAccessTokenParam}}"
}
```

> **Note**
>
> `"branch"` é necessário somente se seus scripts estiverem armazenados em uma ramificação diferente de `master`.
> Para usar a versão de seus scripts que estão em uma confirmação específica no repositório, use `commitID` com `getOptions` em vez de `branch`. Por exemplo:

```
    "getOptions": "commitID:bbc1ddb94...b76d3bEXAMPLE",
```

- No campo Command Line (Linha de comando), digite os parâmetros para a execução do script. Aqui está um exemplo.

```
mainFile.py argument-1 argument-2
```

  Este exemplo executa o `mainFile.py` que, por sua vez, pode executar outros scripts no diretório `complex-script`.

- (Opcional) Em Working Directory (Diretório de trabalho), digite o nome de um diretório do nó em que deseja baixar e executar o script.
- (Opcional) Em Execution Timeout (Tempo limite de execução), especifique o número de segundos para o sistema aguardar antes de a execução do comando de script falhar.

6. Na seção Targets (Destinos), escolha os nós gerenciados nos quais você quer executar essa operação, especificando as etiquetas, selecionando as instâncias ou dispositivos de borda manualmente ou especificando um grupo de recursos.

> **Tip**
>
> Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

7. Para Other parameters (Outros parâmetros):

- Em Comment (Comentário), digite as informações sobre esse comando.
- Em Timeout (seconds) (Tempo limite [segundos]), especifique o número de segundos para o sistema aguardar até a falha de execução do comando total.

8. Para Rate control (Controle de taxa):

- Em Concurrency (Concorrência), especifique um número ou uma porcentagem de nós gerenciados nos quais executar o comando ao mesmo tempo.

> **Note**
>
> Se você selecionou destinos especificando tags aplicadas a instâncias a nós gerenciados ou especificando grupos de recursos da AWS, e não tiver certeza de

quantas instâncias são direcionadas, restrinja o número de instâncias que poderão executar o documento ao mesmo tempo, especificando uma porcentagem.

- Em Error threshold (Limite de erro), especifique quando parar de executar o comando em outros nós depois de falhar em alguns ou em uma porcentagem de nós. Por exemplo, se você especificar três erros, o Systems Manager deixará de enviar o comando quando o 4° erro for recebido. Os nós gerenciados que continuam processando o comando também podem enviar erros.

9. (Opcional) Em Output options (Opções de saída), para salvar a saída do comando em um arquivo, selecione a caixa Write command output to an S3 bucket (Gravar saída do comando em um bucket do S3). Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

> **Note**
>
> As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil de instância (para instâncias do EC2) ou perfil de serviço do IAM (máquinas ativadas para ambientes híbridos) atribuído à instância, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço do IAM associado ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

10. Na seção SNS notifications (Notificações do SNS), se quiser enviar notificações sobre o status da execução do comando, marque a caixa de seleção Enable SNS notifications (Habilitar notificações do SNS).

    Para obter mais informações sobre a configuração de notificações do Amazon SNS para o Run Command, consulte Monitorar alterações de status do Systems Manager usando as notificações do Amazon SNS.

11. Escolha Executar.

Executar um script do Python no GitHub usando a AWS CLI

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Execute o comando a seguir para baixar e executar um script do GitHub.

```
aws ssm send-command --document-name "AWS-RunRemoteScript" --instance-
ids "instance-IDs" --parameters '{"sourceType":["GitHub"],"sourceInfo":
["{\"owner\":\"owner_name\", \"repository\":\"repository_name\", \"path\":
 \"path_to_script_or_directory\"}"],"commandLine":["commands_to_run"]}'
```

Aqui está um exemplo.

```
aws ssm send-command --document-name "AWS-RunRemoteScript" --instance-ids
 "i-02573cafcfEXAMPLE" --parameters '{"sourceType":["GitHub"],"sourceInfo":
["{\"owner\":\"TestUser1\", \"repository\":\"GitHubTestPublic\", \"path\":
 \"scripts/python/complex-script\"}"],"commandLine":["mainFile.py argument-1
 argument-2 "]}'
```

Este exemplo baixa um diretório de scripts denominado `complex-script`. A entrada `commandLine` executa o `mainFile.py` que, por sua vez, pode executar outros scripts no diretório `complex-script`.

Usar os perfis do Chef InSpec com o Systems Manager Compliance

O AWS Systems Manager é integrado ao Chef InSpec. O Chef InSpec é uma estrutura de teste de código aberto que permite criar perfis legíveis para armazenamento no GitHub ou no Amazon Simple Storage Service (Amazon S3). Em seguida, você pode usar o Systems Manager para executar verificações de compatibilidade e visualizar nós compatíveis e não compatíveis. Um perfil é um requisito de segurança, compatibilidade ou política de seu ambiente de computação. Por exemplo, é possível criar perfis que executam as seguintes verificações ao verificar os nós com o Compliance, uma ferramenta do AWS Systems Manager:

- Verificar se portas específicas estão abertas ou fechadas.
- Verificar se aplicativos específicos estão em execução.
- Verificar se determinados pacotes estão instalados.
- Verificar propriedades específicas em chaves do Registro do Windows.

Você pode criar perfis do InSpec apenas para instâncias do Amazon Elastic Compute Cloud (Amazon EC2) que você gerencia com o Systems Manager. Os servidores on-premises ou as máquinas virtuais (VMs) não são compatíveis. O seguinte exemplo do perfil do Chef InSpec verifica se a porta 22 está aberta.

```
control 'Scan Port' do
impact 10.0
title 'Server: Configure the service port'
desc 'Always specify which port the SSH server should listen to.
Prevent unexpected settings.'
describe sshd_config do
its('Port') { should eq('22') }
end
end
```

O InSpec inclui um conjunto de recursos que ajudam você a escrever rapidamente verificações e controles de auditoria. O InSpec usa a Domain-specific Language (DSL) do InSpec para escrever esses controles no Ruby. Você também pode usar perfis criados por uma grande comunidade de usuários do InSpec. Por exemplo, o projeto DevSec chef-os-hardening no GitHub inclui dezenas de perfis para ajudar você a proteger seus nós. Você pode criar e armazenar perfis no GitHub ou no Amazon S3.

Como funciona

Veja a seguir a forma como o processo de usar perfis do InSpec com o Compliance funciona:

1. Identifique perfis do InSpec predefinidos que você deseja usar ou crie seus próprios. Você pode usar perfis predefinidos no GitHub para começar. Para obter informações sobre como criar seus próprios perfis do InSpec, consulte Perfis do Chef InSpec
2. Armazene perfis em um repositório GitHub público ou privado ou em um bucket do S3.
3. Execute o Compliance com seus perfis do InSpec usando o documento do Systems Manager (documento do SSM) `AWS-RunInspecChecks`. Você pode iniciar uma verificação do Compliance usando o Run Command, uma ferramenta do AWS Systems Manager, para verificações sob demanda ou programar verificações regulares do Compliance usando o State Manager, uma ferramenta do AWS Systems Manager.
4. Identifique os nós incompatíveis usando a API Compliance ou o console do Compliance.

> **Note**
>
> Observe as seguintes informações:
>
> - O Chef usa um cliente em seus nós gerenciados para processar o perfil. Você não precisa instalar o cliente. Quando o Systems Manager executa o documento `AWS-RunInspecChecks`, o sistema verifica se o cliente está instalado. Caso contrário, o Systems Manager instala o cliente do Chef durante a verificação e, em seguida, desinstala o cliente após a verificação ser concluída.
> - A execução do documento do SSM `AWS-RunInspecChecks`, conforme descrito neste tópico, atribui uma entrada de conformidade do tipo `Custom:Inspec` para cada nó de destino. Para atribuir esse tipo de conformidade, o documento chama a operação de API PutComplianceItems.

Executar uma verificação de conformidade no InSpec

Esta seção inclui informações sobre como executar uma verificação de conformidade do InSpec usando o console do Systems Manager e a AWS Command Line Interface (AWS CLI). O procedimento do console mostra como configurar o State Manager para executar a verificação. O procedimento da AWS CLI mostra como configurar o Run Command para executar a verificação.

Executar uma verificação de conformidade no InSpec com State Manager (console)

Como executar uma verificação de conformidade no InSpec com o State Manager usando o console do AWS Systems Manager

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha State Manager.
3. Selecione Criar associação.
4. Na seção Provide association details (Fornecer detalhes da associação), digite um nome.
5. Na lista Document (Documento), escolha **`AWS-RunInspecChecks`**.
6. Na lista Document version (Versão do documento), escolha Latest at runtime (Mais recente em runtime).
7. Na seção Parâmetros, na lista Tipo de origem, escolha GitHub ou S3.

Se você escolher GitHub, digite o caminho para um perfil do InSpec em um repositório do GitHub público ou privado no campo Informações da origem. Este é um caminho de exemplo para um perfil público fornecido pela equipe do Systems Manager no seguinte local: https://github.com/awslabs/amazon-ssm/tree/master/Compliance/InSpec/PortCheck.

```
{"owner":"awslabs","repository":"amazon-ssm","path":"Compliance/InSpec/
PortCheck","getOptions":"branch:master"}
```

Se você escolher S3, insira um URL válido para um perfil do InSpec em um bucket do S3, no campo Source Info (Informações de origem).

Para obter mais informações sobre como o Systems Manager se integra com o GitHub e com o Amazon S3, consulte Executar scripts do GitHub.

8. Na seção Targets (Destinos), escolha os nós gerenciados nos quais você quer executar essa operação, especificando as tags, selecionando as instâncias ou dispositivos de borda manualmente ou especificando um grupo de recursos.

> **Tip**
>
> Se um nó gerenciado que você espera ver não estiver listado, consulte Solução de problemas de disponibilidade do nó gerenciado para obter dicas de solução de problemas.

9. Na seção Specify schedule (Especificar programação), use as opções do construtor de programação para criar uma programação que especifica quando você deseja que a verificação do Compliance seja executada.
10. Para Rate control (Controle de taxa):

    - Em Concurrency (Concorrência), especifique um número ou uma porcentagem de nós gerenciados nos quais executar o comando ao mesmo tempo.

> **Note**
>
> Se você selecionou destinos especificando tags aplicadas a instâncias a nós gerenciados ou especificando grupos de recursos da AWS, e não tiver certeza de

quantas instâncias são direcionadas, restrinja o número de instâncias que poderão executar o documento ao mesmo tempo, especificando uma porcentagem.

- Em Error threshold (Limite de erro), especifique quando parar de executar o comando em outros nós depois de falhar em alguns ou em uma porcentagem de nós. Por exemplo, se você especificar três erros, o Systems Manager deixará de enviar o comando quando o 4° erro for recebido. Os nós gerenciados que continuam processando o comando também podem enviar erros.

11. (Opcional) Em Output options (Opções de saída), para salvar a saída do comando em um arquivo, selecione a caixa Write command output to an S3 bucket (Gravar saída do comando em um bucket do S3). Digite os nomes de bucket e prefixo (pastas) nas caixas de texto.

> **Note**
>
> As permissões do S3 que concedem a possibilidade de gravar os dados em um bucket do S3 são as do perfil de instância (para instâncias do EC2) ou perfil de serviço do IAM (máquinas ativadas para ambientes híbridos) atribuído à instância, e não as do usuário do IAM que realiza essa tarefa. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager ou Criar um perfil de serviço do IAM para um ambiente híbrido. Além disso, se o bucket do S3 especificado estiver em uma conta da Conta da AWS diferente, verifique se o perfil da instância ou a função de serviço do IAM associado ao nó gerenciado tenha as permissões necessárias para gravar nesse bucket.

12. Escolha Create Association (Criar associação). O sistema cria a associação e executa automaticamente a verificação do Compliance.
13. Aguarde vários minutos até que a verificação seja concluída e, em seguida, escolha Compliance no painel de navegação.
14. Em Corresponding managed instances (Instâncias gerenciadas correspondentes), localize os nós em que a coluna Compliance Type (Tipo de compatibilidade) seja Custom:Inspec.
15. Escolha o ID de um nó para visualizar os detalhes dos status de incompatibilidade.

Executar uma verificação de conformidade no InSpec com Run Command (AWS CLI)

1. Instale e configure a AWS Command Line Interface (AWS CLI), caso ainda não o tenha feito.

   Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

2. Execute um dos seguintes comandos para executar um perfil do InSpec no GitHub ou no Amazon S3.

   O comando usa os seguintes parâmetros:

   - sourceType: GitHub ou Amazon S3
   - sourceInfo: URL para a pasta de perfil do InSpec no GitHub ou em um bucket do S3. A pasta base deve conter o arquivo base do InSpec (*.yml) e todos os controles relacionados (*.rb).

   GitHub

   ```
   aws ssm send-command --document-name "AWS-RunInspecChecks" --targets
    '[{"Key":"tag:tag_name","Values":["tag_value"]}]' --parameters '{"sourceType":
   ["GitHub"],"sourceInfo":["{\"owner\":\"owner_name\", \"repository\":
   \"repository_name\", \"path\": \"Inspec.yml_file"}"]}'
   ```

   Aqui está um exemplo.

   ```
   aws ssm send-command --document-name "AWS-RunInspecChecks" --targets
    '[{"Key":"tag:testEnvironment","Values":["webServers"]}]' --parameters
    '{"sourceType":["GitHub"],"getOptions":"branch:master","sourceInfo":["{\"owner\":
   \"awslabs\", \"repository\":\"amazon-ssm\", \"path\": \"Compliance/InSpec/PortCheck
   \"}"]}'
   ```

   Amazon S3

   ```
   aws ssm send-command --document-name "AWS-RunInspecChecks" --targets
    '[{"Key":"tag:tag_name","Values":["tag_value"]}]' --parameters'{"sourceType":
   ["S3"],"sourceInfo":["{\"path\":\"https://s3.aws-api-domain/amzn-s3-demo-
   bucket/Inspec.yml_file\"}"]}'
   ```

   Aqui está um exemplo.

   ```
   aws ssm send-command --document-name "AWS-RunInspecChecks" --targets
    '[{"Key":"tag:testEnvironment","Values":["webServers"]}]' --
   parameters'{"sourceType":["S3"],"sourceInfo":["{\"path\":\"https://s3.aws-api-
   domain/amzn-s3-demo-bucket/InSpec/PortCheck.yml\"}"]}'
   ```

3. Execute o comando a seguir para visualizar um resumo da verificação do Compliance.

```
aws ssm list-resource-compliance-summaries --filters
 Key=ComplianceType,Values=Custom:Inspec
```

4. Execute o comando a seguir para ver detalhes de um nó que não é compatível.

```
aws ssm list-compliance-items --resource-ids node_ID --resource-type
 ManagedInstance --filters Key=DocumentName,Values=AWS-RunInspecChecks
```

Integração com o ServiceNow

O ServiceNow fornece um sistema de gerenciamento de serviços baseado na nuvem para criar e gerenciar fluxos de trabalho em nível organizacional, como serviços de TI, sistemas de chamados e suporte. O AWS Service Management Connector integra o ServiceNow ao Systems Manager para provisionar, gerenciar e operar recursos da AWS diretamente no ServiceNow. É possível usar o AWS Service Management Connector para integrar o ServiceNow ao Automation, Change Manager, Incident Manager e OpsCenter, todos eles ferramentas do AWS Systems Manager.

Ao usar o ServiceNow, é possível executar as seguintes tarefas:

- Executar playbooks de automação do Systems Manager.
- Visualizar, atualizar e resolver incidentes de OpsItems do Systems Manager.
- Visualizar e gerenciar itens operacionais, como incidentes, por meio do OpsCenter do Systems Manager.
- Visualizar e executar solicitações de alteração do Systems Manager com base em uma lista selecionada de modelos pré-aprovados de alteração.
- Gerenciar e resolver incidentes envolvendo aplicativos hospedados na AWS por meio de integração com o Incident Manager.

> **Note**
>
> Para obter informações sobre como fazer a integração com o ServiceNow, consulte Configurar integrações ao serviço da AWS no Guia do administrador do AWS Service Manager Connector.

Referência do AWS Systems Manager

As informações e os tópicos a seguir podem ajudá-lo a melhorar a implementação de soluções do AWS Systems Manager.

Principal

No AWS Identity and Access Management (IAM), você pode conceder ou negar acesso de um serviço a recursos que usam o elemento de Política principal. O valor do elemento de Política principal para o Systems Manager é `ssm.amazonaws.com`.

Regiões da AWS e endpoints com suporte

Consulte Systems Manager service endpoints no Referência geral da Amazon Web Services.

Service Quotas

Consulte as cotas de serviço do Systems Manager no Referência geral da Amazon Web Services.

API Reference

Consulte Referência da API do AWS Systems Manager.

Referência de comandos da AWS CLI

Consulte a seção AWS Systems Manager da Referência de comandos da AWS CLI.

Referência do cmdlet do Ferramentas da AWS para PowerShell

Consulte a seção AWS Systems Manager da Referência de cmdlets do Ferramentas da AWS para PowerShell.

Repositório do SSM Agent no GitHub

Consulte aws/amazon-ssm-agent.

Faca uma pergunta

Problemas do Systems Manager em AWSre:Post

Notícias do blog da AWS

Ferramentas de gerenciamento

Mais tópicos de referência

- Usar este serviço com um AWS SDK
- Referência: buckets do Amazon S3 para operações de aplicação de patches
- Referência: padrões e tipos de eventos do Amazon EventBridge para o Systems Manager
- Referência: Expressões cron e rate para o Systems Manager
- Referência: ec2messages, ssmmessages e outras operações da API
- Referência: Formatos de strings de data e hora para o Systems Manager

Usar este serviço com um AWS SDK

Os kits de desenvolvimento de software (SDKs) da AWS estão disponíveis para muitas linguagens de programação populares. Cada SDK fornece uma API, exemplos de código e documentação que permitem que os desenvolvedores criem facilmente aplicações em seu idioma de preferência.

Documentação do SDK	Exemplos de código
AWS SDK para C++	Exemplos de código do AWS SDK para C++
AWS CLI	Exemplos de código do AWS CLI
AWS SDK para Go	Exemplos de código do AWS SDK para Go
AWS SDK para Java	Exemplos de código do AWS SDK para Java
AWS SDK para JavaScript	Exemplos de código do AWS SDK para JavaScript
AWS SDK para Kotlin	Exemplos de código do AWS SDK para Kotlin
AWS SDK para .NET	Exemplos de código do AWS SDK para .NET
AWS SDK para PHP	Exemplos de código do AWS SDK para PHP
Ferramentas da AWS para PowerShell	Exemplos de código do Ferramentas da AWS para PowerShell
AWS SDK para Python (Boto3)	Exemplos de código do AWS SDK para Python (Boto3)

Documentação do SDK	Exemplos de código
AWS SDK para Ruby	Exemplos de código do AWS SDK para Ruby
AWS SDK para Rust	Exemplos de código do AWS SDK para Rust
SDK da AWS para SAP ABAP	Exemplos de código do SDK da AWS para SAP ABAP
AWS SDK for Swift	Exemplos de código do AWS SDK for Swift

> **Exemplo de disponibilidade**
>
> Não consegue encontrar o que precisa? Solicite um exemplo de código usando o link Fornecer feedback na parte inferior desta página.

Referência: buckets do Amazon S3 para operações de aplicação de patches

Durante a execução de várias operações de aplicação de patches do Patch Manager, o AWS Systems Manager Agent (SSM Agent) acessa uma série de buckets do Amazon Simple Storage Service (Amazon S3) que são de propriedade e mantidos pela Amazon Web Services (AWS). Esses buckets do S3 são acessíveis publicamente e, por padrão, SSM Agent se conecta a eles usando chamadas HTTP.

No entanto, se você estiver usando um endpoint da nuvem privada virtual (VPC) nas operações do Systems Manager, deverá fornecer permissão explícita em um perfil de instância do Amazon Elastic Compute Cloud (Amazon EC2) para o Systems Manager ou em um perfil de serviço para máquinas que não são do EC2 em um ambiente híbrido e multinuvem. Caso contrário, seus recursos não poderão acessar esses buckets públicos.

Este tópico de referência lista os buckets relacionados a patches para cada Região da AWS compatível.

Para mais informações sobre como especificar esses buckets do S3 em perfis de instância do EC2, consulte Comunicações do SSM Agent com os buckets do S3 gerenciados pela AWS.

Para obter mais informações sobre como usar endpoints da VPC com o Systems Manager, consulte Melhorar a segurança das instâncias do EC2 usando endpoints da VPC para o Systems Manager.

Tópicos

- Buckets contendo documentos do SSM Command para operações de aplicação de patches (Linux e Windows Server)
- Buckets contendo documentos do SSM Command para operações de aplicação de patches (macOS)
- Buckets contendo snapshots da lista de referência de patches gerenciadas pela AWS

Buckets contendo documentos do SSM Command para operações de aplicação de patches (Linux e Windows Server)

Os buckets com o formato `aws-patch-manager-`*`region`*`-`*`unique-suffix` contêm os seguintes documentos usados pelas operações de aplicação de patch do Patch Manager em sistemas operacionais Linux e Windows Server:

- `AWS-RunPatchBaseline`
- `AWS-RunPatchBaselineAssociation`
- `AWS-RunPatchBaselineWithHooks`
- `AWS-InstanceRebootWithHooks`
- `AWS-PatchAsgInstance`
- `AWS-PatchInstanceWithRollback`

Nome da região	Código da região	Bucket do **`aws-patch-manager-`** ***`region`*****`-`*****`suffix`**
Leste dos EUA (Ohio)	us-east-2	aws-patch-manager-us-east-2-552881074
Leste dos EUA (Norte da Virgínia)	us-east-1	aws-patch-manager-us-east-1-1970c647d
Oeste dos EUA (N. da Califórnia)	us-west-1	aws-patch-manager-us-west-1-8badb4304

Nome da região	Código da região	Bucket do **aws-patch-manager-** ***region-suffix***
Oeste dos EUA (Oregon)	us-west-2	aws-patch-manager-us-west-2-34d7f99f8
África (Cidade do Cabo)	af-south-1	aws-patch-manager-af-south-1-bdd5f65a9
Ásia-Pacífico (Hong Kong)	ap-east-1	aws-patch-manager-ap-east-1-632356271
Ásia-Pacífico (Hyderabad)	ap-south-2	aws-patch-manager-ap-south-2-32f4b4128
Ásia-Pacífico (Jacarta)	ap-southeast-3	aws-patch-manager-ap-southeast-3-aa48fc462
Ásia-Pacífico (Melbourne)	ap-southeast-4	aws-patch-manager-ap-southeast-4-01e2c40d3
Ásia-Pacífico (Mumbai)	ap-south-1	aws-patch-manager-ap-south-1-cb7c62ff9
Ásia-Pacífico (Osaka)	ap-northeast-3	aws-patch-manager-ap-northeast-3-67373598a
Ásia-Pacífico (Seul)	ap-northeast-2	aws-patch-manager-ap-northeast-2-10467995c
Ásia-Pacífico (Singapura)	ap-southeast-1	aws-patch-manager-ap-southeast-1-7fdfd9ef7
Ásia-Pacífico (Sydney)	ap-southeast-2	aws-patch-manager-ap-southeast-2-17283a275
Ásia-Pacífico (Tóquio)	ap-northeast-1	aws-patch-manager-ap-northeast-1-4849fa78f

Nome da região	Código da região	Bucket do **aws-patch-manager-** ***region-suffix***
Canadá (Central)	ca-central-1	aws-patch-manager-ca-central-1-3148e69e3
Oeste do Canadá (Calgary)	ca-west-1	aws-patch-manager-ca-west-1-9e3a4b2f9
Europa (Frankfurt)	eu-central-1	aws-patch-manager-eu-central-1-9163fdaaf
Europa (Irlanda)	eu-west-1	aws-patch-manager-eu-west-1-5522fb710
Europa (Londres)	eu-west-2	aws-patch-manager-eu-west-2-902a2bc74
Europa (Milão)	eu-south-1	aws-patch-manager-eu-south-1-c52f3f594
Europa (Paris)	eu-west-3	aws-patch-manager-eu-west-3-29bf85721
Europa (Espanha)	eu-south-2	aws-patch-manager-eu-south-2-a4cf248b1
Europa (Estocolmo)	eu-north-1	aws-patch-manager-eu-north-1-795879e9b
Europa (Zurique)	eu-central-2	aws-patch-manager-eu-central-2-184ce43c8
Israel (Tel Aviv)	il-central-1	aws-patch-manager-il-central-1-e221cb57b
Oriente Médio (Barém)	me-south-1	aws-patch-manager-me-south-1-a53fc9dce

Nome da região	Código da região	Bucket do **aws-patch-manager-** ***region-suffix***
Oriente Médio (Emirados Árabes Unidos)	me-central-1	aws-patch-manager-me-central-1-2932f2f80
América do Sul (São Paulo)	sa-east-1	aws-patch-manager-sa-east-1-ddf4b6a09

Buckets contendo documentos do SSM Command para operações de aplicação de patches (macOS)

Os buckets com o formato `aws-patchmanager-macos-`*`region-unique-suffix* contêm os seguintes documentos usados pelas operações de aplicação de patch do Patch Manager em sistemas operacionais macOS:

- `AWS-RunPatchBaseline`
- `AWS-RunPatchBaselineAssociation`
- `AWS-RunPatchBaselineWithHooks`
- `AWS-InstanceRebootWithHooks`
- `AWS-PatchAsgInstance`
- `AWS-PatchInstanceWithRollback`

Nome da região	Código da região	Bucket do **aws-patchmanager-macos-** ***region-suffix***
Leste dos EUA (Ohio)	us-east-2	aws-patchmanager-macos-us-east-2-552881074
Leste dos EUA (Norte da Virgínia)	us-east-1	aws-patchmanager-macos-us-east-1-1970c647d
Oeste dos EUA (N. da Califórnia)	us-west-1	aws-patchmanager-macos-us-west-1-8badb4304

Nome da região	Código da região	Bucket do **aws-patchmanager-macos-*region-suffix***
Oeste dos EUA (Oregon)	us-west-2	aws-patchmanager-macos-us-west-2-34d7f99f8
África (Cidade do Cabo)	af-south-1	aws-patchmanager-macos-af-south-1-bdd5f65a9
Ásia-Pacífico (Hong Kong)	ap-east-1	aws-patchmanager-macos-ap-east-1-632356271
Ásia-Pacífico (Hyderabad)	ap-south-2	aws-patchmanager-macos-ap-south-2-32f4b4128
Ásia-Pacífico (Jacarta)	ap-southeast-3	aws-patchmanager-macos-ap-southeast-3-aa48fc462
Ásia-Pacífico (Melbourne)	ap-southeast-4	aws-patchmanager-macos-ap-southeast-4-01e2c40d3
Ásia-Pacífico (Mumbai)	ap-south-1	aws-patchmanager-macos-ap-south-1-cb7c62ff9
Ásia-Pacífico (Osaka)	ap-northeast-3	aws-patchmanager-macos-ap-northeast-3-67373598a
Ásia-Pacífico (Seul)	ap-northeast-2	aws-patchmanager-macos-ap-northeast-2-10467995c
Ásia-Pacífico (Singapura)	ap-southeast-1	aws-patchmanager-macos-ap-southeast-1-7fdfd9ef7
Ásia-Pacífico (Sydney)	ap-southeast-2	aws-patchmanager-macos-ap-southeast-2-17283a275
Ásia-Pacífico (Tóquio)	ap-northeast-1	aws-patchmanager-macos-ap-northeast-1-4849fa78f

Nome da região	Código da região	Bucket do **aws-patchmanager-macos-*region-suffix***
Canadá (Central)	ca-central-1	aws-patchmanager-macos-ca-central-1-3148e69e3
Oeste do Canadá (Calgary)	ca-west-1	aws-patchmanager-macos-ca-west-1-9e3a4b2f9
Europa (Frankfurt)	eu-central-1	aws-patchmanager-macos-eu-central-1-9163fdaaf
Europa (Irlanda)	eu-west-1	aws-patchmanager-macos-eu-west-1-5522fb710
Europa (Londres)	eu-west-2	aws-patchmanager-macos-eu-west-2-902a2bc74
Europa (Milão)	eu-south-1	aws-patchmanager-macos-eu-south-1-c52f3f594
Europa (Paris)	eu-west-3	aws-patchmanager-macos-eu-west-3-29bf85721
Europa (Espanha)	eu-south-2	aws-patchmanager-macos-eu-south-2-a4cf248b1
Europa (Estocolmo)	eu-north-1	aws-patchmanager-macos-eu-north-1-795879e9b
Europa (Zurique)	eu-central-2	aws-patchmanager-macos-eu-central-2-184ce43c8
Israel (Tel Aviv)	il-central-1	aws-patchmanager-macos-il-central-1-e221cb57b
Oriente Médio (Barém)	me-south-1	aws-patchmanager-macos-me-south-1-a53fc9dce

Nome da região	Código da região	Bucket do **aws-patchmanager-macos-*region-suffix***
Oriente Médio (Emirados Árabes Unidos)	me-central-1	aws-patchmanager-macos-me-central-1-2932f2f80
América do Sul (São Paulo)	sa-east-1	aws-patchmanager-macos-sa-east-1-ddf4b6a09

Buckets contendo snapshots da lista de referência de patches gerenciadas pela AWS

Os buckets com o formato `patch-baseline-snapshot-`*`region`* ou `patch-baseline-snapshot-`*`region`*`-`*`unique-suffix`* contêm snapshots da lista de referência de patches gerenciadas pela AWS. O acesso a este bucket do S3 será necessário se você usar qualquer um dos seguintes documentos do SSM:

- `AWS-RunPatchBaseline`
- `AWS-RunPatchBaselineAssociation`
- `AWS-RunPatchBaselineWithHooks`
- `AWS-ApplyPatchBaseline` (um documentos do SSM legado)

Nome da região	Código da região	Bucket do **patch-baseline-snapshot-***
Leste dos EUA (Ohio)	us-east-2	patch-baseline-snapshot-us-east-2
Leste dos EUA (Norte da Virgínia)	us-east-1	patch-baseline-snapshot-us-east-1
Oeste dos EUA (N. da Califórnia)	us-west-1	patch-baseline-snapshot-us-west-1

Nome da região	Código da região	Bucket do **patch-baseline-snapshot-***
Oeste dos EUA (Oregon)	us-west-2	patch-baseline-snapshot-us-west-2
África (Cidade do Cabo)	af-south-1	patch-baseline-snapshot-af-south-1-tbxdb5b9
Ásia-Pacífico (Hong Kong)	ap-east-1	patch-baseline-snapshot-ap-east-1
Ásia-Pacífico (Hyderabad)	ap-south-2	patch-baseline-snapshot-ap-south-2-50209442
Ásia-Pacífico (Jacarta)	ap-southeast-3	patch-baseline-snapshot-ap-southeast-3-be0a3174
Ásia-Pacífico (Melbourne)	ap-southeast-4	patch-baseline-snapshot-ap-southeast-4-dc6f76ce
Ásia-Pacífico (Mumbai)	ap-south-1	patch-baseline-snapshot-ap-south-1
Ásia-Pacífico (Osaka)	ap-northeast-3	patch-baseline-snapshot-ap-northeast-3
Ásia-Pacífico (Seul)	ap-northeast-2	patch-baseline-snapshot-ap-northeast-2
Ásia-Pacífico (Singapura)	ap-southeast-1	patch-baseline-snapshot-ap-southeast-1
Ásia-Pacífico (Sydney)	ap-southeast-2	patch-baseline-snapshot-ap-southeast-2
Ásia-Pacífico (Tóquio)	ap-northeast-1	patch-baseline-snapshot-ap-northeast-1

Nome da região	Código da região	Bucket do **patch-baseline-snapshot-***
Canadá (Central)	ca-central-1	patch-baseline-snapshot-ca-central-1
Oeste do Canadá (Calgary)	ca-west-1	patch-baseline-snapshot-ca-west-1
Europa (Frankfurt)	eu-central-1	patch-baseline-snapshot-eu-central-1
Europa (Irlanda)	eu-west-1	patch-baseline-snapshot-eu-west-1
Europa (Londres)	eu-west-2	patch-baseline-snapshot-eu-west-2
Europa (Milão)	eu-south-1	patch-baseline-snapshot-eu-south-1
Europa (Paris)	eu-west-3	patch-baseline-snapshot-eu-west-3
Europa (Espanha)	eu-south-2	patch-baseline-snapshot-eu-south-2-df2c9d70
Europa (Estocolmo)	eu-north-1	patch-baseline-snapshot-eu-north-1
Europa (Zurique)	eu-central-2	patch-baseline-snapshot-eu-central-2
Israel (Tel Aviv)	il-central-1	patch-baseline-snapshot-il-central-1
Oriente Médio (Barém)	me-south-1	patch-baseline-snapshot-me-south-1-uduvl7q8

Nome da região	Código da região	Bucket do **patch-baseline-snapshot-***
Oriente Médio (Emirados Árabes Unidos)	me-central-1	patch-baseline-snapshot-me-central-1
América do Sul (São Paulo)	sa-east-1	patch-baseline-snapshot-sa-east-1

Referência: padrões e tipos de eventos do Amazon EventBridge para o Systems Manager

> **Note**
>
> O Amazon EventBridge é a maneira preferencial de gerenciar seus eventos. O CloudWatch Events e o EventBridge são o mesmo serviço subjacente e API, mas o EventBridge oferece mais recursos. Alterações feitas no CloudWatch ou no EventBridge são refletidas em cada console. Para obter mais informações, consulte o Guia do Usuário do Amazon EventBridge.

Usando o Amazon EventBridge, você pode criar regras que correspondam à entrada de eventos e encaminhá-las aos destinos para processamento.

Um evento indica uma alteração em um ambiente em suas próprias aplicações, em aplicações de software como serviço (SaaS) ou em um AWS service (Serviço da AWS). Eventos são emitidos com base no melhor esforço. Depois que um tipo de evento especificado em uma regra for detectado, o EventBridge o encaminha para um destino especificado para o processamento. Os destinos podem incluir instâncias do Amazon Elastic Compute Cloud (Amazon EC2), funções do AWS Lambda, fluxos do Amazon Kinesis, tarefas do Amazon Elastic Container Service (Amazon ECS), máquinas de estado do AWS Step Functions, tópicos do Amazon Simple Notification Service (Amazon SNS), filas do Amazon Simple Queue Service (Amazon SQS), metas internas e muito mais.

Para obter informações sobre a criação de regras do EventBridge, consulte os seguintes tópicos:

- Monitorar eventos do Systems Manager com o Amazon EventBridge

- Exemplos de eventos do Amazon EventBridge para Systems Manager
- Conceitos básicos do Amazon EventBridge no Guia do usuário do Amazon EventBridge

O restante deste tópico descreve os tipos de eventos do Systems Manager que podem ser incluídos nas regras do EventBridge.

Tipo de evento: automação

Nome do tipo de evento	Descrição dos eventos que você pode adicionar a uma regra
Notificação de alteração de status de execução de automação do EC2	O status geral de um fluxo de trabalho de automação é alterado. É possível adicionar uma ou mais das alterações de status a seguir a uma regra de eventos: • Aprovado • Cancelado • Falha • PendingApproval • PendingChangeCalendarOverride • Rejeitado • Programado • Bem-sucedida • TimedOut
Notificação de alteração do status da etapa de automação do EC2	O status de uma etapa específica em um fluxo de trabalho de automação é alterado. É possível adicionar uma ou mais das alterações de status a seguir a uma regra de eventos: • Cancelado • Falha • Bem-sucedida

Nome do tipo de evento	Descrição dos eventos que você pode adicionar a uma regra
	• TimedOut

Tipo de evento: Change Calendar

Nome do tipo de evento	Descrição dos eventos que você pode adicionar a uma regra
Alteração do estado do calendário	Altera o estado de um alarme do Change Calendar. É possível adicionar uma ou ambas as alterações de estado a seguir a uma regra de eventos: • OPEN • FECHADO Alterações de estado para calendários compartilhados de outras contas da Contas da AWS não são permitidas.

Tipo de evento: Change Manager

Nome do tipo de evento	Descrição dos eventos que você pode adicionar a uma regra
Atualização do status da solicitação de alteração	O estado de uma solicitação de alteração do Change Manager. É possível usar os seguintes estados em uma regra de evento: • Aprovado • Rejeitado • InProgress

Tipo de evento: conformidade com a configuração

Nome do tipo de evento	Descrição dos eventos que você pode adicionar a uma regra
Alteração do estado de compatibilidade da configuração	O estado de um nó gerenciado muda, para manter a conformidade da associação ou a conformidade do patch. É possível adicionar uma ou mais das alterações de estado a seguir a uma regra de eventos: • compatível • non_compliant

Tipo de evento: inventário

Nome do tipo de evento	Descrição dos eventos que você pode adicionar a uma regra
Alteração do estado dos recursos do inventário	A exclusão de inventário personalizado e uma chamada PutInventory que usa uma versão de esquema antiga. É possível adicionar uma ou mais das alterações de estado a seguir a uma regra de eventos: • Evento de tipo de inventário excluído em um nó específico. O EventBridge envia um evento por nó para cada InventoryType personalizado. • Evento de tipo de inventário excluído em todos os nós. • Chamada PutInventory com evento de versão do esquema antigo. O EventBrid ge envia esse evento quando a versão do esquema for menor do que o esquema atual.

Nome do tipo de evento	Descrição dos eventos que você pode adicionar a uma regra
	Este evento se aplica a todos os tipos de inventário. Para obter mais informações, consulte Sobre o monitoramento de eventos do Inventory do EventBridge.

Tipo de evento: janela de manutenção

Nome do tipo de evento	Descrição dos eventos que você pode adicionar a uma regra
Notificação de alteração do status da janela de manutenção	O status geral de uma ou mais janelas de manutenção é alterado. É possível adicionar uma ou mais das alterações de estado a seguir a uma regra de eventos: • DESATIVADA • ENABLED
Notificação de registro de destino da janela de manutenção	O estado de uma ou mais metas da janela de manutenção é alterado. É possível adicionar uma ou mais das alterações de estado a seguir a uma regra de eventos: • CANCELAR REGISTRO • REGISTRADO • UPDATED
Notificação de alteração do estado de execução da janela de manutenção	O status geral de uma janela de manutenção muda enquanto ela estiver sendo executada. É possível adicionar uma ou mais das alterações de estado a seguir a uma regra de eventos:

Nome do tipo de evento	Descrição dos eventos que você pode adicionar a uma regra
	• CANCELADO • CANCELAMENTO • COM FALHA • IN_PROGRESS • PENDING • SKIPPED_OVERLAPPING • SUCCESS • TIMED_OUT
Notificação de alteração do estado de execução de tarefas da janela de manutenção	O status de uma tarefa em uma janela de manutenção é alterado enquanto ela estiver sendo executada. É possível adicionar uma ou mais das alterações de estado a seguir a uma regra de eventos: • CANCELADO • CANCELAMENTO • COM FALHA • IN_PROGRESS • SUCCESS • TIMED_OUT

Nome do tipo de evento	Descrição dos eventos que você pode adicionar a uma regra
Notificação de alteração do estado de invocação de tarefas da janela de manutenção	O estado de uma tarefa de janela de manutenção em um destino específico é alterado. Esta notificação é totalmente suportada apenas para tarefas do Run Command. Para este tipo de tarefa, é possível adicionar uma ou mais das alterações de estado a seguir a uma regra de eventos: • CANCELADO • CANCELAMENTO • COM FALHA • IN_PROGRESS • SUCCESS • TIMED_OUT Para ó Automation, AWS Lambda e tarefas do AWS Step Functions, o EventBridge relata somente os estados `IN_PROGRESS` e `COMPLETE`. O estado `COMPLETE` é relatado se a tarefa foi ou não bem-sucedida.
Notificação de registro de tarefas da janela de manutenção	O estado de uma ou mais tarefas da janela de manutenção é alterado. É possível adicionar uma ou mais das alterações de estado a seguir a uma regra de eventos: • CANCELAR REGISTRO • REGISTRADO • UPDATED

Tipo de evento: OpsCenter

Nome do tipo de evento	Descrição dos eventos que você pode adicionar a uma regra
Criação de OpsItem	Ocorre quando um OpsItem é criado. É possível adicionar regras para um dos seguintes tipos OpsItem: • /aws/issue • /aws/task • /aws/insight • /aws/actionitem
Atualização de OpsItem	Ocorre quando um OpsItem é atualizado. É possível adicionar regras para um dos seguintes tipos OpsItem: • /aws/issue • /aws/task • /aws/insight • /aws/actionitem

Tipo de evento: Parameter Store

Nome do tipo de evento	Descrição dos eventos que você pode adicionar a uma regra
Alteração do repositório de parâmetros	O estado de um parâmetro muda. É possível adicionar uma ou mais das alterações de estado a seguir a uma regra de eventos: • Criar • Atualizar • Excluir

Nome do tipo de evento	Descrição dos eventos que você pode adicionar a uma regra
	• LabelParameterVersion Para obter mais informações, consulte Configurar regras do EventBridge para parâmetros e políticas de parâmetros.
Ação da política do repositório de parâmetros	Uma condição de uma alteração de política de parâmetro avançada é atendida. É possível adicionar uma ou mais das alterações de status a seguir a uma regra de eventos: • Expiração • ExpirationNotification • NoChangeNotification Para obter mais informações, consulte Configurar regras do EventBridge para parâmetros e políticas de parâmetros.

Tipo de evento: Run Command

Nome do tipo de evento	Descrição dos eventos que você pode adicionar a uma regra
Notificação de alteração de status de invocação do comando do EC2	O status de um comando enviado a uma instância gerenciada individual é alterado. É possível adicionar uma ou mais das alterações de status a seguir a uma regra de eventos: • Bem-sucedida • InProgress • TimedOut

Nome do tipo de evento	Descrição dos eventos que você pode adicionar a uma regra
	• Cancelado • Falha
Notificação de alteração de status de comando do EC2	O status geral de um comando é alterado. É possível adicionar uma ou mais das alterações de status a seguir a uma regra de eventos: • Bem-sucedida • InProgress • TimedOut • Cancelado • Falha

Tipo de evento: State Manager

Nome do tipo de evento	Descrição dos eventos que você pode adicionar a uma regra
Alteração do estado de associação do EC2 State Manager	O estado geral de uma Associação muda à medida que ela estiver sendo aplicada. É possível adicionar uma ou mais das alterações de estado a seguir a uma regra de eventos: • Falha • Pendente • Bem-sucedida
Alteração do estado da associação da instância do State Manager no EC2	O estado de uma instância única gerenciada que é direcionada por alterações de associaçã o. É possível adicionar uma ou mais das alterações de estado a seguir a uma regra de eventos:

Nome do tipo de evento	Descrição dos eventos que você pode adicionar a uma regra
	• Falha • Pendente • Bem-sucedida

Referência: Expressões cron e rate para o Systems Manager

Ao criar uma associação do State Manager ou uma janela de manutenção no AWS Systems Manager, você especifica uma programação para quando a janela ou a associação deve ser executada. É possível especificar um agendamento como uma entrada baseada em horário, chamada de expressão cron, ou como uma entrada baseada em frequência, chamada de expressão rate.

Informações gerais sobre as expressões cron e rate

As informações a seguir são aplicáveis a expressões cron e rate para janelas de manutenção e associações.

Programações com execução única

Ao criar uma associação ou uma janela de manutenção, é possível especificar um carimbo de data/hora no formato Tempo Universal Coordenado (UTC) para que ela seja executada uma vez no horário especificado. Por exemplo: `"at(2020-07-07T15:55:00)"`

Deslocamentos de programação

Associações e janelas de manutenção oferecem suporte a deslocamentos de programação apenas para expressões cron. Um deslocamento de programação é o número de dias de espera após a data e a hora especificadas por uma expressão do cron antes de executar a associação ou janela de manutenção.

Maintenance window example

No exemplo acima, a expressão CRON agenda a execução de uma janela de manutenção na terceira terça-feira de cada mês às 23h30. No entanto, como o deslocamento de programação é 2, a janela de manutenção só será executada dois dias depois às 23h30.

```
aws ssm create-maintenance-window \
```

```
    --name "My-Cron-Offset-Maintenance-Window" \
    --allow-unassociated-targets \
    --schedule "cron(30 23 ? * TUE#3 *)" \
    --duration 4 \
    --cutoff 1 \
    --schedule-offset 2
```

Association example

No comando a seguir, a expressão cron agenda uma associação para ser executada na segunda quinta-feira de cada mês. No entanto, como a diferença de cronograma é 3, a associação não funcionará até o próximo domingo, três dias depois.

```
aws ssm create-association \
    --name "AWS-UpdateSSMAgent" \
    --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f" \
    --schedule-expression "cron(0 0 ? * THU#2 *)"  \
    --schedule-offset 3
    --apply-only-at-cron-interval
```

> **Note**
>
> Para usar um deslocamento com uma associação, você deve especificar a opção `--apply-only-at-cron-interval`. Essa opção diz ao sistema para não executar uma associação imediatamente após sua criação.

Se você criar uma associação ou janela de manutenção com uma expressão do cron que tenha como destino um dia que já passou em relação ao período atual, mas adicionar uma data futura de deslocamento de programação, a associação ou janela de manutenção não será executada no período. Ela entrará em vigor no período seguinte. Por exemplo, se você especificar uma expressão cron que teria executado uma janela de manutenção ontem e adicionar um deslocamento de programação de dois dias, a janela de manutenção não será executada amanhã.

Campos obrigatórios

Expressões cron para janelas de manutenção têm seis campos obrigatórios. Expressões cron para associações têm cinco. (no momento, o State Manager não oferece suporte à especificação de meses em expressões cron para associações.) Um campo adicional, o campo `Seconds` (o primeiro em uma expressão cron), é opcional. Os campos são separados por um espaço.

Exemplos de expressão cron

Minutos	Horas	Dia do mês	Mês	Dia da semana	Ano	Significado
0	10	*	*	?	*	Executada às 10h (UTC) todos os dias
15	12	*	*	?	*	Executada às 12h15 (UTC) todos os dias
0	18	?	*	SEG-SEX	*	Executada às 18h (UTC) de segunda a sexta
0	8	1	*	?	*	Executada às 8h (UTC) todo o primeiro dia do mês

Valores com suporte

A tabela a seguir mostra os valores compatíveis com as entradas cron necessárias.

Valores compatíveis com expressões cron

Campo	Valores	Curingas
Minutos	`0-59`	`, - * /`

Campo	Valores	Curingas
Horas	0-23	, - * /
Dia do mês	1-31	, - * ? / L W
Mês (somente para janelas de manutenção)	1-12 ou JAN-DEC	, - * /
Dia da semana	1-7 ou SUN-SAT	, - * ? / L #
Ano	1970-2199	, - * /

> **Note**
>
> Não é possível especificar os campos de dia do mês e dia da semana na mesma expressão cron. Se você especificar um valor em um dos campos, use um ? (ponto de interrogação) no outro.

Curingas para expressões cron

A tabela a seguir mostra os valores curinga compatíveis com expressões cron.

> **Note**
>
> As expressões Cron que levam a taxas mais rápidas do que 5 (cinco) minutos não têm suporte. O suporte para especificar um valor de dia da semana e dia do mês ao mesmo tempo não está completo. Use o ponto de interrogação (?) em um desses campos.

Curingas compatíveis com expressões cron

Curinga	Descrição
,	O curinga , (vírgula) inclui valores adicionais. No campo Mês, JAN, FEV, MAR incluiria janeiro, fevereiro e março.

Curinga	Descrição
-	O curinga - (traço) especifica faixas. No campo Dia, 1-15 incluiria dias 1 a 15 do mês especificado.
*	O curinga * (asterisco) inclui todos os valores no campo. No campo Horas, * incluiria cada hora.
/	O curinga / (barra) especifica incrementos. No campo Minutos, você pode inserir 1/10 para especificar cada décimo minuto a partir do primeiro minuto da hora. Portanto, 1/10 especifica o primeiro, 11º, 21º e 31º minuto, e assim por diante.
?	O curinga ? (interrogação) especifica um ou outro. No campo Dia do mês, você pode inserir 7 e se não se importa com qual dia da semana era o 7º, pode inserir ? no campo Dia da semana.
L	O curinga L nos campos Dia do mês ou Dia da semana especifica o último dia do mês ou da semana.
W	O curinga `W` no campo Dia do mês especific a um dia da semana. No campo Dia do mês, 3W especifica o dia mais próximo do terceiro dia da semana do mês.
#	O caractere curinga # no campo day-of-week (dia-da-semana) seguido de um número entre um e cinco especifica um determinado dia do mês. 5#3 especifica a terceira quinta-feira do mês.

Expressões rate

Expressões rate têm os seguintes dois campos obrigatórios. Os campos são separados por espaços.

Campos obrigatórios para expressões rate

Campo	Valores
Valor	número positivo, como 1 ou 15

Unidade	`minute`
	`minutes`
	`hour`
	`hours`
	`day`
	`days`

Se o valor for igual a 1, a unidade deverá ser singular. Da mesma forma, para valores maiores do que 1, a unidade deve ser plural. Por exemplo, `rate(1 hours)` e `rate(5 hour)` não são válidas, mas `rate(1 hour)` e `rate(5 hours)` são.

Tópicos

- Expressões cron e rate para associações
- Expressões cron e rate para janelas de manutenção

Expressões cron e rate para associações

Esta seção inclui exemplos de associações do State Manager com expressões cron e rate. Antes de criar uma dessas expressões, fique atento às informações a seguir:

- As associações comportam as seguintes expressões cron: a cada 1/2, 1, 2, 4, 8 ou 12 horas; todo dia, toda semana ou cada dia e hora especificados da semana; um dia específico em uma semana específica do mês ou o último x dia do mês na hora específica.

- As associações comportam apenas as seguintes expressões de taxas: intervalos de 30 minutos ou mais e menos de 31 dias.
- Se você especificar o campo opcional `Seconds`, o valor poderá ser 0 (zero). Por exemplo: `cron(0 */30 * * * ? *)`
- Para uma associação que coleta metadados para o Inventory, uma ferramenta do AWS Systems Manager, recomendamos o uso de uma expressão rate.
- Atualmente, o State Manager não oferece suporte à especificação de meses em expressões cron para associações.

Associações comportam expressões do cron que incluem um dia da semana e o sinal numérico (#) para designar o nº dia de um mês para executar uma associação. Aqui está um exemplo que executa uma programação do cron na terceira terça-feira de cada mês às 23h30 UTC:

```
cron(30 23 ? * TUE#3 *)
```

Aqui está um exemplo que acontece na segunda quinta-feira de cada mês à meia-noite UTC:

```
cron(0 0 ? * THU#2 *)
```

Associações também aceitam o sinal (L) para indicar o último dia X do mês. Aqui está um exemplo que executa uma programação do cron na última terça-feira de cada mês à meia-noite UTC:

```
cron(0 0 ? * 3L *)
```

Para controlar ainda mais quando uma associação é executada, por exemplo, se você quiser executar uma associação dois dias após o patch de terça-feira, você pode especificar um deslocamento. Um deslocamento define quantos dias esperar após o dia programado para executar uma associação. Por exemplo, se você especificou uma programação do cron de `cron(0 0 ? * THU#2 *)`, você pode especificar o número 3 no campo Schedule offset (Deslocamento da programação) para executar a associação todos os domingos após a segunda quinta-feira do mês.

Para usar deslocamentos, você deve escolher a opção Apply association only at the next specified Cron interval (Aplicar associação somente no próximo intervalo de Cron especificado) no console ou você deve especificar o parâmetro de uso `--apply-only-at-cron-interval` a partir da linha de comando. Essa opção diz ao State Manager para não executar uma associação imediatamente após sua criação.

A tabela a seguir apresenta exemplos cron para associações.

Exemplos de cron para associações

Exemplo	Detalhes
cron(0/30 * * * ? *)	A cada 30 minutos
cron(0 0/1 * * ? *)	A cada hora
cron(0 0/2 * * ? *)	A cada 2 horas
cron(0 0/4 * * ? *)	A cada 4 horas
cron(0 0/8 * * ? *)	A cada 8 horas
cron(0 0/12 * * ? *)	A cada 12 horas
cron(15 13 ? * * *)	Todos os dia às 13h15
cron(15 13 ? * MON *)	Todas as segundas às 13h15
cron(30 23 ? * TUE#3 *)	A terceira terça-feira de cada mês às 23h30

Veja alguns exemplos de associações com rate.

Exemplos de rate para associações

Exemplo	Detalhes
rate(30 minutes)	A cada 30 minutos
rate(1 hour)	A cada hora
rate(5 hours)	A cada 5 horas
rate(15 days)	A cada 15 dias

Exemplos da AWS CLI para associações

Para criar associações State Manager usando a AWS CLI, inclua o parâmetro `--schedule-expression` com uma expressão cron ou rate. Os exemplos a seguir usam a AWS CLI em uma máquina Linux local.

> **Note**
>
> Por padrão, quando você cria uma nova associação, o sistema a executa imediatamente após sua criação e de acordo com a programação especificada. Especifique `--apply-only-at-cron-interval` para que a associação não seja executada imediatamente após a criação. Esse parâmetro não é compatível com expressões rate.

```
aws ssm create-association \
    --association-name "My-Cron-Association" \
    --schedule-expression "cron(0 2 ? * SUN *)" \
    --targets Key=tag:ServerRole,Values=WebServer \
    --name AWS-UpdateSSMAgent
```

```
aws ssm create-association \
    --association-name "My-Rate-Association" \
    --schedule-expression "rate(7 days)" \
    --targets Key=tag:ServerRole,Values=WebServer \
    --name AWS-UpdateSSMAgent
```

```
aws ssm create-association \
    --association-name "My-Rate-Association" \
    --schedule-expression "at(2020-07-07T15:55:00)" \
    --targets Key=tag:ServerRole,Values=WebServer \
    --name AWS-UpdateSSMAgent \
    --apply-only-at-cron-interval
```

Expressões cron e rate para janelas de manutenção

Esta seção inclui exemplos de expressões cron e rate para janelas de manutenção.

Ao contrário das associações do State Manager, as janelas de manutenção são compatíveis com todas as expressões cron e de taxa. Isso inclui suporte para valores no campo de segundos.

Por exemplo, a expressão cron de 6 campos a seguir executa uma janela de manutenção às 9:30 AM todos os dias.

```
cron(30 09 ? * * *)
```

Ao adicionar um valor ao campo `Seconds`, a expressão cron de 7 campos a seguir executa uma janela de manutenção às 9:30:24 hs todos os dias.

```
cron(24 30 09 ? * * *)
```

A tabela a seguir fornece exemplos adicionais de cron de 6 campos para janelas de manutenção.

Exemplos de cron para janelas de manutenção

Exemplo	Detalhes
cron(0 2 ? * THU#3 *)	02:00, na terceira quinta-feira de cada mês
cron(15 10 ? * * *)	10:15, todos os dias
cron(15 10 ? * MON-FRI *)	10:15 todas as segundas, terças, quartas, quintas e sexta-feiras
cron(0 2 L * ? *)	02:00, no último dia de cada mês
cron(15 10 ? * 6L *)	10:15, na última sexta-feira de cada mês

A tabela a seguir fornece exemplos de rate para janelas de manutenção.

Exemplos de rate para janelas de manutenção

Exemplo	Detalhes
rate(30 minutes)	A cada 30 minutos
rate(1 hour)	A cada hora
rate(5 hours)	A cada 5 horas
rate(25 days)	A cada 25 dias

Exemplos de AWS CLI para janelas de manutenção

Para criar janelas de manutenção usando a AWS CLI, inclua o parâmetro `--schedule` com uma expressão cron ou rate ou um carimbo de data/hora. Os exemplos a seguir usam a AWS CLI em uma máquina Linux local.

```
aws ssm create-maintenance-window \
    --name "My-Cron-Maintenance-Window" \
    --allow-unassociated-targets \
    --schedule "cron(0 16 ? * TUE *)" \
    --schedule-timezone "America/Los_Angeles" \
    --start-date 2021-01-01T00:00:00-08:00 \
    --end-date 2021-06-30T00:00:00-08:00 \
    --duration 4 \
    --cutoff 1
```

```
aws ssm create-maintenance-window \
    --name "My-Rate-Maintenance-Window" \
    --allow-unassociated-targets \
    --schedule "rate(7 days)" \
    --duration 4 \
    --schedule-timezone "America/Los_Angeles" \
    --cutoff 1
```

```
aws ssm create-maintenance-window \
    --name "My-TimeStamp-Maintenance-Window" \
    --allow-unassociated-targets \
    --schedule "at(2021-07-07T13:15:30)" \
    --duration 4 \
    --schedule-timezone "America/Los_Angeles" \
    --cutoff 1
```

Mais informações

- Expressão CRON, no Site da Wikipedia

Referência: ec2messages, ssmmessages e outras operações da API

Se você monitorar operações de API, poderá ver chamadas para as seguintes operações:

- ec2messages:AcknowledgeMessage
- ec2messages:DeleteMessage
- ec2messages:FailMessage
- ec2messages:GetEndpoint
- ec2messages:GetMessages
- ec2messages:SendReply
- ssmmessages:CreateControlChannel
- ssmmessages:CreateDataChannel
- ssmmessages:OpenControlChannel
- ssmmessages:OpenDataChannel
- ssm:DescribeDocumentParameters
- ssm:DescribeInstanceProperties
- ssm:GetCalendar
- ssm:GetManifest
- ssm:ListInstanceAssociations
- ssm:PutCalendar
- ssm:PutConfigurePackageResult
- ssm:RegisterManagedInstance
- ssm:RequestManagedInstanceRoleToken
- ssm:UpdateInstanceAssociationStatus
- ssm:UpdateInstanceInformation
- ssm:UpdateManagedInstancePublicKey

Essas são operações especiais usadas pelo AWS Systems Manager, conforme descrito no restante deste tópico.

Operações de API relacionadas a agentes (endpoints **ssmmessages** e **ec2messages**)

Operações de API ssmmessages

O Systems Manager usa o endpoint `ssmmessages` para estes tipos de operações de API:

- Operações do Systems Manager Agent (SSM Agent) para o serviço do Systems Manager na nuvem.
- Operações do SSM Agent para o Session Manager, uma ferramenta do AWS Systems Manager, na nuvem. Esse endpoint é necessário para criar e excluir canais de sessão com o serviço Session Manager na nuvem. Além disso, se a conectividade for permitida, o SSM Agent receberá documentos do `Command` por meio deste Amazon Message Gateway Service. Se a conectividade não for permitida, o SSM Agent receberá documentos do `Command` via Amazon Message Delivery Service. Para obter mais informações, consulte Ações, recursos e chaves de condição do Amazon Message Gateway Service.
- Operações de Run Command.

Operações da API ec2messages

As operações de API `ec2messages:*` são feitas para o endpoint do Amazon Message Delivery Service. O Systems Manager usa esse endpoint para fazer operações de API do Systems Manager Agent (SSM Agent) para o serviço Systems Manager na nuvem.

> **Important**
>
> As operações da API do `ec2messages:*` são aceitas somente nas Regiões da AWS lançadas antes de 2024. Nas regiões lançadas em 2024 e posteriormente, somente as operações da API do `ssmmessages:*` são aceitas.

Precedência de conexão do endpoint

A partir da versão 3.3.40.0 do SSM Agent, o Systems Manager começou a usar o endpoint `ssmmessages:*` (Amazon Message Gateway Service) sempre que disponível, em vez do endpoint `ec2messages:*` (Amazon Message Delivery Service).

Se você fornecer acesso a `ssmmessages:*` em suas políticas de permissão do AWS Identity and Access Management (IAM), o SSM Agent se conectará ao endpoint `ssmmessages:*`, mesmo que seu perfil de instância do IAM esteja configurado para permitir os dois endpoints. Isso inclui políticas para perfis de instância do IAM e perfis de serviço do IAM que você mesmo criou e para perfis de instância do IAM criados pela Configuração de gerenciamento de hosts de Quick Setup e pela configuração padrão de gerenciamento de hosts.

Se você tiver fornecido permissões para ambos os endpoints e monitorado as operações de API usando, por exemplo, o CloudWatch Metrics, você não verá nenhuma chamada para. `ec2messages:*`

Para Regiões da AWS lançadas antes de 2024: é possível remover as permissões de `ec2messages:*` com segurança das suas políticas.

Failover de conexão do endpoint

Somente para Regiões da AWS lançadas antes de 2024: se seu perfil de instância do IAM não fornecer permissões para `ssmmessages:*` no momento que o agente for iniciado, mas apenas `ec2messages:*`, o SSM Agent se conectará ao endpoint `ec2messages:*`. Se você tiver `ssmmessages:*` e `ec2messages:*` ao mesmo tempo no no momento que SSM Agent iniciar, mas remover `ssmmessages:*` após a inicialização do agente, o SSM Agent logo transferirá a conexão para o endpoint `ec2messages:*`. Para regiões lançadas em 2024 e posteriormente, somente o endpoint `ssmmessages:*` é aceito.

Para obter mais informações sobre os endpoints `ssmmessages` e `ec2messages:*`, consulte os seguintes tópicos na Referência de autorização de serviço da AWS.

- Ações, recursos e chaves de condição para o Amazon Message Gateway Service (`ssmmessages`).
- Ações, recursos e chaves de condição para o Amazon Message Delivery Service (`ec2messages:*`)

Operações de API relacionadas à instância do namespace **`ssm:*`**

`DescribeDocumentParameters`

O Systems Manager executa essa operação da API para renderizar nós específicos no console do Amazon EC2. Os resultados da operação `DescribeDocumentParameters` são exibidos em seu nó Documentos.

`DescribeInstanceProperties`

O Systems Manager executa essa operação da API para renderizar nós específicos no console do Amazon EC2. Os resultados da operação `DescribeInstanceProperties` são exibidos em seu nó Fleet Manager.

`GetCalendar`

O Systems Manager executa essa operação da API para renderizar documentos do tipo Change Calendar no console do Change Calendar.

`GetManifest`

O SSM Agent executa essa operação da API para determinar os requisitos do sistema para instalar ou atualizar uma versão específica de um pacote AWS Systems Manager Distributor. Essa é uma operação de API herdada e não está disponível nas Regiões da AWS iniciadas após 2017.

`ListInstanceAssociations`

O SSM Agent executa essa operação da API para ver se uma nova associação do State Manager está disponível. Essa operação de API é essencial para o State Manager funcionar.

`PutCalendar`

O Systems Manager executa essa operação da API para atualizar documentos do tipo Change Calendar no console do Change Calendar.

`PutConfigurePackageResult`

O SSM Agent executa essa operação da API para publicar métricas de erro de instalação e latência de pacotes públicos do Distributor na conta do proprietário do pacote.

`RegisterManagedInstance`

O SSM Agent executa essa operação de API para os seguintes cenários:

- Para registrar um servidor on-premises ou máquina virtual (VM) com o Systems Manager como uma instância gerenciada usando um código de ativação e um ID.
- Para registrar as credenciais do AWS IoT Greengrass Version 2.

Essa operação também é chamada por instâncias do Amazon EC2 que executam a versão 3.1.x ou posterior do SSM Agent.

`RequestManagedInstanceRoleToken`

O SSM Agent executa essa operação da API para recuperar credenciais temporárias para acessar o nó gerenciado.

`UpdateInstanceAssociationStatus`

SSM Agent: o agente executa essa operação de API para atualizar uma associação. Essa operação de API é essencial para o State Manager, uma ferramenta do AWS Systems Manager, funcionar.

`UpdateInstanceInformation`

O SSM Agent chama o serviço Systems Manager na nuvem a cada cinco minutos para fornecer informações sobre pulsação. Essa chamada é necessária para manter uma pulsação com o agente, para que o serviço saiba que o agente está funcionando conforme esperado.

`UpdateManagedInstancePublicKey`

O SSM Agent executa essa operação da API para fornecer a chave pública depois de alternar o par de chaves no nó gerenciado. A chave pública é usada para autenticar as solicitações, assinadas com a chave privada, para obter credenciais temporárias do Systems Manager.

ssm:* namespace outras operações de API

`ExecuteApi`

Os administradores delegados do Systems Manager que gerenciam OpsItems no OpsCenter exigem acesso a essa ação de API para que possam visualizar detalhes de recursos relacionados a OpsItems em várias Contas da AWS. Especificamente, essa API dá a um administrador delegado permissão para visualizar os seguintes detalhes de OpsItem no AWS Management Console: a descrição de OpsItem, tags, modelo do AWS CloudFormation, alterações no AWS Config, alarmes do CloudWatch Logs e eventos do AWS CloudTrail. Para obter mais informações sobre como trabalhar com OpsItems entre contas diferentes, consulte (Opcional) Configuração do OpsCenter para gerenciar OpsItems de forma centralizada entre contas. Para obter mais informações sobre detalhes de recursos relacionados para OpsItems, consulte Adição de recursos relacionados para um OpsItem.

Referência: Formatos de strings de data e hora para o Systems Manager

As operações da API do AWS Systems Manager aceitam filtros para limitar o número de resultados retornados por uma solicitação. Algumas dessas operações da API aceitam filtros que exigem uma string formatada para representar uma data e uma hora específicas.

Por exemplo, a operação da API `DescribeSessions` aceita as chaves `InvokedBefore` e `InvokedAfter` como alguns valores válidos para um objeto `SessionFilter`. Outro exemplo é a ação da API `DescribeAutomationExecutions`, que aceita as chaves `StartTimeAfter` e `StartTimeBefore` como alguns dos valores válidos para um objeto `AutomationExecutionFilter`. Os valores fornecidos para essas chaves ao filtrar as solicitações devem corresponder ao padrão ISO 8601. Para obter informações sobre o ISO 8601, consulte ISO 8601.

Essas strings de data e hora formatadas não estão limitadas a filtros. Há também operações de API que exigem uma string no formato ISO 8601 para representar uma data e uma hora específicas ao fornecer um valor para um parâmetro de solicitação. Por exemplo, o parâmetro de solicitação `AtTime` da operação `GetCalendarState`. Essas strings são difíceis de criar. Use os exemplos neste tópico para criar strings de data e hora formatadas para usar com operações da API do Systems Manager.

Formatar strings de data e hora para o Systems Manager

Veja a seguir um exemplo de uma string de data e hora no formato ISO 8601.

```
2024-05-08T15:16:43Z
```

Ela representa: 8 de maio de 2024, às 15h16, padrão UTC (Tempo Universal Coordenado). A parte da data do calendário da string é representada por um ano de quatro dígitos, um mês de dois dígitos e um dia de dois dígitos separados por hifens. Isso pode ser representado no formato a seguir.

```
YYYY-MM-DD
```

A parte de hora da string começa com a letra "T" como um delimitador e, depois, é representada por uma hora de dois dígitos, um minuto de dois dígitos e um segundo de dois dígitos, separados por dois pontos. Isso pode ser representado no formato a seguir.

```
hh:mm:ss
```

A parte de hora da string termina com a letra "Z", indicando o padrão UTC.

Criar strings personalizados de data e hora para o Systems Manager

É possível criar strings personalizadas de data e hora na máquina local usando sua ferramenta de linha de comando preferencial. A sintaxe usada para criar uma string de data e hora no formato ISO

8601 difere dependendo do sistema operacional da máquina local. Veja a seguir exemplos de como você pode usar `date` no coreutils do GNU no Linux ou no PowerShell do Windows para criar uma string de data e hora no formato ISO 8601.

coreutils

```
date '+%Y-%m-%dT%H:%M:%SZ'
```

PowerShell

```
(Get-Date).ToString("yyyy-MM-ddTH:mm:ssZ")
```

Ao trabalhar com operações da API do Systems Manager, talvez seja necessário criar strings históricas de data e hora para fins de geração de relatórios ou solução de problemas. Veja a seguir exemplos de como você pode criar e usar strings históricas e personalizadas de data e hora no formato ISO 8601 para o Ferramentas da AWS para PowerShell e a AWS Command Line Interface (AWS CLI).

AWS CLI

- Recupere a última semana do histórico de comandos para um documento do SSM.

```
lastWeekStamp=$(date '+%Y-%m-%dT%H:%M:%SZ' -d '7 days ago')

docFilter='{"key":"DocumentName","value":"AWS-RunPatchBaseline"}'
timeFilter='{"key":"InvokedAfter","value":'\"$lastWeekStamp\"'}'

commandFilters=[$docFilter,$timeFilter]

aws ssm list-commands \
    --filters $commandFilters
```

- Recupere a última semana do histórico de execução de automação.

```
lastWeekStamp=$(date '+%Y-%m-%dT%H:%M:%SZ' -d '7 days ago')

aws ssm describe-automation-executions \
    --filters Key=StartTimeAfter,Values=$lastWeekStamp
```

- Recupere o último mês do histórico da sessão.

```
lastWeekStamp=$(date '+%Y-%m-%dT%H:%M:%SZ' -d '30 days ago')

aws ssm describe-sessions \
    --state History \
    --filters key=InvokedAfter,value=$lastWeekStamp
```

Ferramentas da AWS para PowerShell

- Recupere a última semana do histórico de comandos para um documento do SSM.

```
$lastWeekStamp = (Get-Date).AddDays(-7).ToString("yyyy-MM-ddTH:mm:ssZ")

$docFilter = @{
    Key="DocumentName"
    Value="AWS-InstallWindowsUpdates"
}
$timeFilter = @{
    Key="InvokedAfter"
    Value=$lastWeekStamp
}

$commandFilters = $docFilter,$timeFilter

Get-SSMCommand `
    -Filters $commandFilters
```

- Recupere a última semana do histórico de execução de automação.

```
$lastWeekStamp = (Get-Date).AddDays(-7).ToString("yyyy-MM-ddTH:mm:ssZ")

Get-SSMAutomationExecutionList `
    -Filters @{Key="StartTimeAfter";Values=$lastWeekStamp}
```

- Recupere o último mês do histórico da sessão.

```
$lastWeekStamp = (Get-Date).AddDays(-30).ToString("yyyy-MM-ddTH:mm:ssZ")

Get-SSMSession `
    -State History `
    -Filters @{Key="InvokedAfter";Value=$lastWeekStamp}
```

Casos de uso e melhores práticas

Esse tópico lista os casos de uso comuns e as melhores práticas para ferramentas do AWS Systems Manager. Se disponível, esse tópico também inclui links para postagens de blog relevantes e a documentação técnica.

> **Note**
>
> O título de cada seção aqui é um link ativo na seção correspondente na documentação técnica.

Automação

- Crie runbooks do Automation do autoatendimento para infraestrutura.
- Use o Automation, uma ferramenta do AWS Systems Manager, para simplificar a criação de Amazon Machine Images (AMIs) do AWS Marketplace ou de uma AMIs personalizada, usando documentos públicos do Systems Manager (documentos SSM) ou criando seus próprios fluxos de trabalho.
- Crie e mantenha as AMIs usando o `AWS-UpdateLinuxAmi` e runbooks do Automation do `AWS-UpdateWindowsAmi` ou runbooks do Automation personalizados criados por você.

Conformidade

- Como prática recomendada de segurança, recomendamos que você atualize o perfil (IAM) AWS Identity and Access Management usado pelos nós gerenciados para restringir a capacidade do nó de usar a ação da API PutComplianceItems. Essa ação da API registra um tipo de conformidade e outros detalhes de conformidade em um recurso designado, como uma instância do Amazon EC2 ou um nó gerenciado. Para obter mais informações, consulte Configuração de permissões para Compliance.

Inventário

- Use o Inventory, uma ferramenta do AWS Systems Manager, com o AWS Config para auditar suas configurações de aplicações ao longo do tempo.

Maintenance Windows

- Defina um agendamento para realizar ações potencialmente disruptivas em seus nós, como patches de sistema operacional (SO), atualizações de drivers ou instalações de software.
- Para obter informações sobre as diferenças entre State Manager e Maintenance Windows, uma ferramenta do AWS Systems Manager, consulte Selecionar entre State Manager e Maintenance Windows.

Parameter Store

- Use o Parameter Store, uma ferramenta do AWS Systems Manager, para gerenciar de forma centralizada as opções de configuração globais.
- Como o AWS Systems ManagerParameter Store usa o AWS KMS
- Fazer referência a segredos do AWS Secrets Manager de parâmetros do Parameter Store.

Patch Manager

- Use o Patch Manager, uma ferramenta do AWS Systems Manager, para distribuir patches em grande escala e aumentar a visibilidade da conformidade da frota em seus nós.
- Integre o Patch Manager ao AWS Security Hub para receber alertas quando os nós da sua frota saírem de conformidade e para monitorar o status da aplicação de patches das suas frotas de um ponto de vista de segurança. Será cobrada uma taxa para o uso do Security Hub. Para obter mais informações, consulte Preço.
- Use somente um método por vez para verificar a conformidade com os patches nos nós gerenciados, a fim de evitar sobrescrever acidentalmente dados de conformidade.

Run Command

- Gerenciar instâncias em grande escala sem acesso ao SSH usando o recurso Run Command do EC2.
- Audite todas as chamadas de API feitas por ou em nome do Run Command, uma ferramenta do AWS Systems Manager, usando o AWS CloudTrail.
- Ao enviar um comando usando o Run Command, não inclua informações confidenciais formatadas como texto sem formatação, como senhas, dados de configuração ou outros segredos. Todas as atividades de API do Systems Manager em sua conta são registradas em um bucket do S3 para

logs do AWS CloudTrail. Isso significa que qualquer usuário com acesso ao bucket do S3 poderá visualizar os valores de texto simples desses segredos. Por esse motivo, recomendamos a criação e o uso de parâmetros `SecureString` para criptografar os dados sigilosos que você usa nas operações do Systems Manager.

Para obter mais informações, consulte Restringir o acesso a parâmetros do Parameter Store usando políticas do IAM.

> **Note**
>
> Por padrão, os arquivos de log entregues pelo CloudTrail ao seu bucket são criptografados pela criptografia do servidor da Amazon com chaves de criptografia gerenciadas pelo Amazon S3 (SSE-S3). Para oferecer uma camada de segurança diretamente gerenciável, é possível usar a criptografia do lado do servidor com chaves gerenciadas por AWS KMS(SSE-KMS) para os arquivos de log do CloudTrail.
> Para obter mais informações, consulte Criptografar arquivos de log do CloudTrail com chaves gerenciadas pelo AWS KMS (SSE-KMS), no Guia do usuário do AWS CloudTrail.

- Use os destinos e recursos de controle de taxa em Run Command para realizar uma operação de comando em etapas.
- Use permissões de acesso detalhadas para o Run Command (e todas as ferramentas do Systems Manager), usando as políticas do AWS Identity and Access Management (IAM).

Session Manager

- Registrar as atividades da sessão em log em sua Conta da AWS usando o AWS CloudTrail.
- Registre os dados da sessão na sua Conta da AWS usando o Amazon CloudWatch Logs ou Amazon S3.
- Controle o acesso da sessão do usuário às instâncias.
- Restringir acesso a comandos em uma sessão.
- Desativar ou ativar permissões administrativas da conta ssm-user.

State Manager

- Atualize o SSM Agent pelo menos uma vez por mês usando o documento `AWS-UpdateSSMAgent` pré-configurado.

- (Windows) Carregue o módulo PowerShell ou DSC no Amazon Simple Storage Service (Amazon S3) e use o `AWS-InstallPowerShellModule`.
- Use etiquetas para criar grupos de aplicações para seus nós. Em seguida, defina destinos para os nós gerenciados usando o parâmetro `Targets` em vez de especificar IDs de instância individuais.
- Corrija automaticamente as constatações geradas pelo Amazon Inspector usando o Systems Manager.
- Use um repositório de configuração centralizado para todos os documentos SSM e compartilhe documentos em toda a sua organização.
- Para obter informações sobre as diferenças entre State Manager e Maintenance Windows, consulte Selecionar entre State Manager e Maintenance Windows.

Nós gerenciados

- O Systems Manager requer referências de tempo precisas para realizar suas operações. Se a data e a hora do nó não estiverem definidas corretamente, talvez elas não correspondam à data de assinatura das solicitações da API. Isso pode causar erros ou funcionalidades incompletas. Por exemplo, as instâncias com configurações de tempo incorretas não serão incluídas em suas listas de nós gerenciados.

  Para obter mais informações sobre como definir o horário de seus nós, consulte, Definir o horário da sua instância do Amazon EC2.
- Em nós gerenciados pelo Linux, verifique a assinatura do SSM Agent.

Mais informações

- Melhores práticas de segurança do Systems Manager

Excluir recursos e artefatos do Systems Manager

Como prática recomendada, recomendamos que você exclua recursos e artefatos do Systems Manager se não precisar mais exibir dados sobre esses recursos ou usar os artefatos de alguma maneira. A tabela a seguir lista cada ferramenta ou artefato do Systems Manager e um link para obter mais informações sobre como excluir os recursos ou artefatos criados pelo Systems Manager.

Recurso ou artefato	Detalhes
Application Manager	Você não pode excluir uma aplicação no Application Manager, mas pode remover uma aplicação do serviço excluindo as tags subjacentes, os grupos de recursos, ou as pilhas do AWS CloudFormation
Automação	Se você criar recursos do AWS usando o Systems Manager Automation, você deverá excluir manualmente esses recursos usando o AWS Management Console correspondente. Se você criou um runbook personalizado, poderá excluir o documento do SSM subjacent e. Para ter mais informações, consulte Excluir documentos do SSM personalizados.
Change Calendar	Você pode excluir um calendário de alteraçõe s e um evento do calendário de alterações. Para ter mais informações, consulte Excluir um calendário de alterações e Excluir um evento do Change Calendar.
Change Manager	Você pode excluir um modelo de alteração. Para ter mais informações, consulte Excluir modelos de alteração.
Compliance	O Systems Manager Compliance exibe automaticamente os dados de conformid ade sobre a aplicação de patches do Patch Manager e as associações do State Manager. Você não pode excluir esses dados. Caso tenha configurado uma sincronização de dados de recursos para centralizar os dados de conformidade em um bucket do S3, você poderá excluir a sincronização. Para ter mais

Recurso ou artefato	Detalhes
	informações, consulte Excluir uma sincronização de dados de recursos para o Compliance.
Distributor	Você pode excluir pacotes no Distributor. Para ter mais informações, consulte Excluir um pacote do Distributor.
Explorer	Você pode desconectar as fontes das quais o Explorer coleta dados operacionais. Para ter mais informações, consulte Edite as fontes de dados do Systems Manager Explorer. Você também pode excluir uma sincronização de dados do recurso usada pelo Explorer para agregar OpsData e OpsItems em várias contas e Regiões da AWS em um único bucket do Amazon Simple Storage Service (Amazon S3). Para ter mais informações, consulte Excluir a sincronização de dados de recursos do Systems Manager Explorer. Para obter informações sobre como excluir um bucket do S3, consulte Excluir um bucket no Guia do desenvolvedor do Amazon Simple Email Service.
Fleet Manager	Você não pode excluir um nó gerenciado usando o Fleet Manager. Use o Amazon Elastic Compute Cloud (Amazon EC2) Para obter mais informações, consulte Encerrar a instância (Linux) e Encerrar a instância (Windows).

Recurso ou artefato	Detalhes
Inventory	Você pode interromper a coleta de dados do Inventory, excluindo as associações do State Manager que definem a programação e os recursos a partir dos quais coletar metadados. Para ter mais informações, consulte interromper a coleta de dados e excluir os dados do inventário. Se você não quiser mais usar o AWS Systems Manager Inventory para exibir metadados sobre a AWS, também recomendamos excluir sincronizações de dados de recursos usadas para a coleta de dados do inventário. Para ter mais informações, consulte Excluir uma sincronização de dados de recursos do Inventory.
Maintenance Windows	Você pode excluir uma janela de manutenção, um destino da janela de manutenção e uma tarefa da janela de manutenção. Para ter mais informações, consulte Atualize ou exclua recursos da janela de manutenção usando o console.
OpsCenter	Você pode excluir uma pessoa OpsItem chamando a operação da OpsItem API Delete usando AWS Command Line Interface ou AWS SDK. Você não pode excluir um OpsItem no AWS Management Console. Para ter mais informações, consulte Exclua OpsItems.
Parameter Store	Você pode excluir um parâmetro que criou. Para ter mais informações, consulte Excluir parâmetros do Parameter Store.

Recurso ou artefato	Detalhes
Patch Manager	Você pode excluir uma lista de referência de patches personalizada. Para ter mais informaçõ es, consulte Atualizar ou excluir uma linha de base de patches personalizada.
Instalação rápida	Você pode excluir associações criadas pelo Quick Setup. As associações são armazenad as e processadas pelo State Manager. Para ter mais informações, consulte Excluir associaçõ es.
Run Command	Depois que um comando terminar o processam ento, as informações sobre ele serão armazenadas na guia Command history (Histórico de comandos). Você não pode excluir informações da guia Command history (Histórico de comandos).
Automação	Depois que uma automação terminar o processamento, as informações sobre ela serão armazenadas na guia Execuções. Você não pode excluir informações da guia Execuções.
Perfil vinculado a serviço	O Systems Manager cria automaticamente perfis vinculadas ao serviço para algumas ferramentas. Você pode excluir essas funções. Para ter mais informações, consulte Excluir uma função vinculada ao serviço `AWSServiceRoleForAmazonSSM` para o Systems Manager.
Session Manager	O Session Manager não retém dados sobre os recursos depois que você encerra uma sessão. Para encerrar uma sessão, consulte Encerrar uma sessão.

Recurso ou artefato	Detalhes
SSM Agent	Você pode desinstalar manualmente o SSM Agent de seus nós. Para obter mais informaçõ es, consulte os tópicos a seguir. • Linux: Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para Linux • macOS: Instalar e desinstalar o SSM Agent manualmente em instâncias do EC2 para macOS • Windows Server: Abrir Painel de controle e, depois, escolha Adicionar/remover programas.
State Manager	Exclua uma associação. Para ter mais informações, consulte Excluir associações.
Documentos do serviço do Systems Manager.	Você não pode excluir runbooks fornecidos pela AWS ou pelo AWS Support, mas pode excluir runbooks personalizados. Para ter mais informações, consulte Excluir documentos do SSM personalizados.

Selecionar entre State Manager e Maintenance Windows

O State Manager e o Maintenance Windows, ambos ferramentas do AWS Systems Manager, podem executar alguns tipos semelhantes de atualizações nos seus nós gerenciados. Qual deles escolher depende se você precisa automatizar a conformidade do sistema ou executar tarefas de alta prioridade e sensíveis ao tempo durante os períodos especificados.

State Manager e Maintenance Windows: Casos de uso principais

O State Manager, uma ferramenta do AWS Systems Manager, define e mantém a configuração de estado de destino para nós gerenciados e recursos da AWS em sua Conta da AWS. Você pode definir combinações de configurações e destinos como objetos de associação. O State Manager

é a ferramenta recomendada se você deseja manter todos os nós gerenciados em sua conta em um estado consistente, usar o Amazon EC2 Auto Scaling para gerar novos nós ou ter requisitos rigorosos de relatórios de conformidade para os nós gerenciados em sua conta.

Os principais casos de uso para o State Manager são os seguintes:

- Cenários do Auto Scaling: O State Manager pode monitorar todas os novos nós gerenciados iniciados em uma conta manualmente ou por meio de grupos do Auto Scaling. Se qualquer associação na conta estiver definindo esse novo nó como destino (por meio de etiquetas ou de todos os nós), essa associação específica será automaticamente aplicada ao novo nó.
- Relatórios de conformidade: o State Manager pode gerar relatórios de conformidade dos estados requeridos para recursos em sua conta.
- Suporte a todos nós: o State Manager pode direcionar todos os nós em uma determinada conta.

A janela de manutenção realiza uma ou mais ações nos recursos da AWS em uma determinada janela de tempo. Você pode definir uma única janela de manutenção com horários de início e término. Você pode especificar várias tarefas a serem executadas nessa janela de manutenção. Use o Maintenance Windows, uma ferramenta do AWS Systems Manager, se suas operações de alta prioridade incluírem a aplicação de patches em nós gerenciados, a execução de vários tipos de tarefas em seus nós durante um período de atualização ou controlar quando as operações de atualização podem ser executadas em seus nós.

Os principais casos de uso para o Maintenance Windows são os seguintes:

- Executar vários documentos: as janelas de manutenção podem executar várias tarefas. Cada tarefa pode usar um tipo de documento diferente. Como resultado, você pode criar fluxos de trabalho complexos usando diferentes tarefas em uma única janela de manutenção.
- Aplicação de patches: uma janela de manutenção pode fornecer suporte à aplicação de patches para todos os nós gerenciados de uma única região marcada com uma etiqueta ou grupo de recursos específico. Como a aplicação de patches geralmente envolve a eliminação de nós (por exemplo, remoção dos nós de um balanceador de carga), a aplicação de patches e a aplicação de patches após o processamento (retornar os nós de para a produção), podem ser feitas como uma série de tarefas dentro da janela de tempo determinada de um patch.

> **Note**
>
> Com uma janela de manutenção, sua operação de aplicação de patches é limitada a uma única região em uma única conta. Com uma política de patch criada na Quick Setup, uma

ferramenta do Systems Manager, é possível, em vez disso, configurar a aplicação de patches para algumas ou todas as contas e regiões de uma organização criada no AWS Organizations. Para ter mais informações, consulte Configurações de políticas de patches em Quick Setup.

- Ações da janela: as janelas de manutenção podem fazer com que um ou mais conjuntos de ações sejam iniciados dentro de uma janela de tempo específica. As janelas de manutenção não começam fora dessa janela. As ações já iniciadas continuam até serem concluídas, mesmo que terminem fora da janela de tempo.

A tabela a seguir compara os principais recursos do State Manager e Maintenance Windows.

Atributo	State Manager	Maintenance Windows
Integração do AWS CloudFormation	Os modelos AWS CloudFormation são compatíveis com as associações do State Manager.	Os modelos do AWS CloudFormation são compatíveis com as janelas de manutenção, os destinos da janela e as tarefas da janela.
Conformidade	Cada associação do State Manager relata a conformidade com relação ao estado requerido do recurso de destino. Você pode usar o painel de do Compliance para agregar e exibir a conformidade relatada.	Não aplicável.
Integração com o gerenciamento de configurações	O State Manager oferece suporte a soluções externas para o estado de destino, como a Desired State Configuration (DSC) do Microsoft PowerShell, playbooks do Ansible e receitas do Chef. Você pode	Não aplicável.

Atributo	State Manager	Maintenance Windows
	usar as associações do State Manager para testar se as soluções de gerenciamento de configurações funcionam e aplicar suas alterações de configuração aos nós quando você estiver pronto.	
Documentos	As configurações do State Manager podem ser definidas como documentos de política (para coletar informações de inventário), runbooks do Automation, para os recursos da AWS, como buckets do Amazon Simple Storage Service (Amazon S3) ou documentos do Systems Manager Command (documentos SSM) para nós gerenciados.	As configurações das Maintenance Windows podem ser definidas como documentos de automação (ações em diversas etapas com fluxos de trabalho de aprovação opcionais) ou documentos do SSM (estado requerido para nós gerenciados).
Monitoramento	O State Manager monitora alterações na configuração, associação ou no estado de um nó gerenciado (por exemplo, novos nós ficam online). Quando o State Manager detecta essas alterações, a associação determinada é reaplicada aos nós originalmente seleciona dos com essa associação.	Não aplicável.

Atributo	State Manager	Maintenance Windows
Prioridades nas tarefas	Não aplicável.	As tarefas em uma janela de manutenção podem ser consideradas uma prioridade. Todas as tarefas com a mesma prioridade são executadas em paralelo. As tarefas com prioridades mais baixas são executadas depois que as tarefas com prioridades mais altas atingem um estado final. Não há como executar tarefas condicionalmente. Depois que uma tarefa de prioridade mais alta atinge seu estado final, a próxima tarefa de prioridade é executada, independentemente do estado da tarefa anterior.

Atributo	State Manager	Maintenance Windows
Controles de segurança	O State Manager suporta dois controles de segurança ao implantar configurações em uma frota grande. Você pode usar a simultaneidade máxima para definir quantos nós ou recursos simultâne os devem ter a configuração aplicada. Você pode definir uma taxa de erro máxima que pode ser usada para pausar a associação do State Manager, se ocorrer um certo número ou porcentagem de erros em toda a frota.	As janelas de manutençã o suportam dois controles de segurança ao implantar configurações em uma grande frota. Você pode usar a simultaneidade máxima para definir quantos nós ou recursos simultâneos devem ter a configuração aplicada. Você pode definir uma taxa de erro máxima que pode ser usada para pausar as ações em uma janela de manutençã o, se ocorrer um certo número ou porcentagem de erros em toda a frota.

Atributo	State Manager	Maintenance Windows
Programação	Você pode executar associações do State Manager sob demanda, em um intervalo cron específico, em uma determinada taxa ou após a criação delas. Isso é útil se você deseja manter o estado requerido para os recursos de maneira consistente e oportuna. **Important** Expressões cron para associações do State Manager não oferecem suporte ao campo de meses, como `03` ou `MAR` para o mês de março. Se você precisa de atualizações de configuração mensais ou trimestrais, uma janela de manutenção pode atender melhor às suas necessidades. Para ter mais informações, consulte Referência: Expressões cron e rate para o Systems Manager.	As janelas de manutenção suportam várias opções de agendamento, incluindo expressões `at` (por exemplo, `"at(2021-07-07T13:15:30)"`), expressões cron e de taxa, cron com deslocamentos e horários de início e fim para quando as janelas de manutenção devem ser executadas, além de horários de corte para especificar quando interromper o agendamento dentro de uma determinada janela de tempo.

Atributo	State Manager	Maintenance Windows
Definição de destino	As associações do State Manager podem direcionar um ou mais nós usando o ID do nó, etiqueta ou grupo de recursos. O State Manager pode direcionar todos os nós gerenciados em uma determinada conta.	As janelas de manutenção podem ter como destino um ou mais nós, usando IDs, etiquetas ou grupos de recursos do nó.

Atributo	State Manager	Maintenance Windows
Tarefas em janelas de manutenção	Não aplicável.	As janelas de manutençã o podem suportar uma ou mais tarefas em que cada tarefa tem como destino um runbook do Automation ou uma ação do documento de comando específica. Todas as tarefas dentro de uma janela de manutenção são executadas em paralelo, a menos que prioridades diferentes sejam definidas para tarefas diferentes. Em geral, as janelas de manutenção oferecem suporte à quatro tipos de tarefas: • Comandos do Run Command do AWS Systems Manager • Fluxos de trabalho do AWS Systems Manager Automation • Funções do AWS Lambda • Tarefas do AWS Step Functions

Informações relacionadas

Os recursos relacionados a seguir podem ajudar você à medida que trabalha com este serviço.

Preços

Algumas ferramentas do Systems Manager cobram uma taxa. Para obter mais informações, consulte Preços do AWS Systems Manager.

Biblioteca de documentação do AWS Systems Manager

Documentação do AWS Systems Manager: acesse toda a documentação do usuário do Systems Manager, incluindo AWS AppConfig, Incident Manager e AWS Systems Manager para SAP.

AWS re:Post

AWS re:Post: serviço gerenciado de perguntas e respostas da AWS que oferece respostas coletivas e revisadas por especialistas para suas perguntas técnicas.

Blog e podcast da AWS

Leia as postagens do blog sobre o Systems Manager na categoria de ferramentas de gerenciamento da AWS, e outros posts marcados com #Systems Manager.

Cotas de serviço

Confira as cotas de serviço do Systems Manager no Referência geral da Amazon Web Services. A menos que especificado de outra forma, cada cota se aplica a uma única região em uma Conta da AWS.

Referência de autorização do serviço para o Systems Manager

Na Referência de autorização do serviço da AWS, veja as informações sobre as ações, os recursos e as chaves de contexto de condição que você pode usar nas políticas do AWS Identity and Access Management (IAM) para o Systems Manager.

Acordo de Serviço do AWS Systems Manager

O Acordo de Serviço (SLA) do AWS Systems Manager é uma política que rege o uso do Systems Manager e se aplica separadamente a cada Conta da AWS, usando o Systems Manager.

Recursos gerais da AWS

Os seguintes recursos relacionados podem ajudar você ao trabalhar com o AWS.

- Aulas e workshops — Links para cursos de especialidades e baseados em perfil, bem como laboratórios autoguiados para ajudar a aperfeiçoar suas habilidades na AWS e a obter experiência prática.
- Centro dos desenvolvedores da AWS — Explore tutoriais, baixe ferramentas e informe-se sobre eventos para desenvolvedores da AWS.
- Ferramentas do desenvolvedor da AWS — Links para ferramentas de desenvolvedor, SDKs, toolkits de IDE e ferramentas da linha de comando para desenvolver e gerenciar aplicações da AWS.
- Centro de recursos de conceitos básicos — Saiba como configurar a Conta da AWS, participar da comunidade da AWS e lançar sua primeira aplicação.
- Tutoriais práticos — Siga os tutoriais passo a passo para iniciar sua primeira aplicação na AWS.
- Whitepapers da AWS — Links para uma lista abrangente de whitepapers técnicos da AWS que abrangem tópicos como arquitetura, segurança e economia, elaborados pelos arquitetos de soluções da AWS ou por outros especialistas técnicos.
- AWS Support Center – a central para criar e gerenciar seus casos do AWS Support. Também inclui links para outros recursos úteis, como fóruns, perguntas frequentes técnicas, status de integridade do serviço e AWS Trusted Advisor.
- Suporte: a página Web principal para obter informações sobre o Suporte, um canal de suporte de resposta rápida e com atendimento individual para ajudar a construir e a executar aplicações na nuvem.
- Entrar em contato: um ponto central de contato para consultas relativas a faturas da AWS, contas, eventos, uso abusivo e outros problemas.
- Termos do site da AWS – informações detalhadas sobre nossos direitos autorais e marca registrada; sua conta, licença e acesso ao site, entre outros tópicos.

Histórico do documento

A tabela a seguir descreve as alterações importantes na documentação desde a última versão do AWS Systems Manager. Para receber notificações sobre atualizações dessa documentação, inscreva-se em um feed RSS.

Alteração	Descrição	Data
Nova orientação de solução de problemas para falhas de dependência de bloqueio de pacotes zypper em nós gerenciados pelo SLES	Adicionada uma orientaçã o abrangente de solução de problemas para resolver falhas de dependência de bloqueio de pacotes zypper que ocorrem durante as operações `Install` do Patch Manager em nós gerenciados pelo SUSE Linux Enterprise Server. A nova seção Problema: falhas na dependência do bloqueio de pacotes zypper em nós gerenciados pelo SLES fornece informaçõ es detalhadas sobre como identificar, diagnosticar e resolver conflitos de bloqueio de pacotes que impedem a instalação bem-sucedida do patch.	28 de julho de 2025
Orientação de solução de problemas aprimorada para associações de inventário ignoradas	Atualizada a seção de solução de problemas Ignorada para fornecer uma orientação abrangente sobre o tratament o prioritário da associação de inventário. A documenta ção aprimorada agora inclui	24 de julho de 2025

informações detalhadas sobre associações de inventário da Quick Setup, cenários de segmentação explícita e procedimentos de resolução passo a passo para gerenciar associações de inventário conflitantes.

Os principais aprimoramentos são:

- Explicação da ordem de prioridade da associaçã o de inventário (associaç ões globais, explícitas e de Quick Setup)
- Cenários comuns com exemplos específicos e resultados esperados
- Etapas detalhadas de resolução para remover as configurações da Quick Setup ao usar associações de inventário personalizadas
- Observações importantes sobre o comportamento da coleta de dados e a precedência da associação

Atualizada a política gerenciada: AWS-SSM-RemediationAutomation-ExecutionRolePolicy	O Systems Manager atualizou a política gerenciada `AWS-SSM-RemediationAutomation-ExecutionRolePolicy` para melhorar a postura de segurança da API ssm:StartAutomationExecution exigindo permissões para os tipos de recursos de “document” e “automation-execution”. Essa atualização está sendo realizada para melhorar a postura de segurança da API ssm:StartAutomationExecution ao exigir permissões para os tipos de recursos de “document” e “automation-execution”, fornecendo permissões mais abrangentes e detalhadas para a execução da automação de correção. Para obter mais informações, consulte Atualizações do Systems Manager para políticas gerenciadas da AWS.	16 de julho de 2025

Atualizada a política gerenciada: AWS-SSM-RemediationAutomation-AdministrationRolePolicy	O Systems Manager atualizou a política gerenciada `AWS-SSM-RemediationAutomation-AdministrationRolePolicy` para fornecer suporte às melhorias na autorização da API para operações de automação de correção. A política atualizada aprimora as permissões para a execução de atividades definidas nos documentos de automação, com controles de segurança aprimorados e padrões de acesso a recursos para fluxos de trabalho de correção. Para obter mais informações, consulte Atualizações do Systems Manager para políticas gerenciadas da AWS.	16 de julho de 2025

Atualizada a política gerenciada: AWS-SSM-DiagnosisAutomation-ExecutionRolePolicy	O Systems Manager atualizou a política gerenciada `AWS-SSM-DiagnosisAutomation-ExecutionRolePolicy` para fornecer permissões mais detalhadas e precisas para a execução da automação de diagnóstico. A política atualizada inclui descrições aprimoradas para acesso aos recursos do Amazon EC2 e da Amazon VPC, permissões de automação do SSM mais específicas e descrições aprimoradas das permissões do AWS KMS e do IAM com restrições de recursos adequadas. Para obter mais informações, consulte Atualizações do Systems Manager para políticas gerenciadas da AWS.	16 de julho de 2025

Atualizada a política gerenciada: AWS-SSM-DiagnosisAutomation-AdministrationRolePolicy	O Systems Manager atualizou a política gerenciada `AWS-SSM-DiagnosisAutomation-AdministrationRolePolicy` para incluir permissões e condições de segurança mais específicas para operações de automação de diagnóstico. A política atualizada fornece controles de segurança aprimorados para uso de chaves do AWS KMS, acesso ao bucket do Amazon S3 e suposições de perfis, com condições mais rígidas baseadas em recursos e restrições em nível de conta. Para obter mais informações, consulte Atualizações do Systems Manager para políticas gerenciadas da AWS.	16 de julho de 2025
Adicionada documentação para a política gerenciada AmazonSSMAutomationRole	O Systems Manager atualizou a documentação da política gerenciada `AmazonSSMAutomationRole` existente. Para obter mais informações, consulte Atualizações do Systems Manager para políticas gerenciadas da AWS.	15 de julho de 2025

Aprimorada a segurança para documentos do SSM com interpolação de variáveis de ambiente	Os documentos de comandos do AWS Systems Manager agora são compatíveis com a interpolação de variáveis de ambiente ao processar parâmetros. Esse recurso, disponível na versão 2.2 do esquema e com a versão 3.3.2746.0 ou superior do SSM Agent, ajuda a evitar ataques de injeção de comando tratando os valores dos parâmetros como strings literais em vez de comandos potencialmente executáveis. Quando definido como `ENV_VAR`, o agente cria uma variável de ambiente no formato `SSM_parameter-name`, que contém o valor do parâmetro. Para obter mais informações, consulte os tópicos a seguir. • Componentes do documento • Exemplos de tratamento seguro de parâmetros • Elementos e parâmetros de dados • Referência de plug-ins de documentos de comando	14 de julho de 2025

Sincronização de dados de recursos do Explorer disponível em quatro Regiões da AWS de adesão	Por padrão, as sincronizações de dados de recursos do Explorer entre contas e regiões não oferecem suporte à agregação de dados em regiões de adesão. Suporte foi adicionado às seguintes regiões de adesão em 30 de junho de 2025. • Europa (Milão) • África (Cidade do Cabo) • Oriente Médio (Bahrein) • Ásia-Pacífico (Hong Kong) Observe que não é possível usar uma conta de administrador delegado para criar uma sincronização de dados de recursos em regiões de adesão. Você deve usar uma conta gerencial do AWS Organizations. Para obter mais informações, consulte Noções básicas sobre sincronizações de dados de recursos de várias contas e regiões.	30 de junho de 2025

Novas versões da extensão do Lambda para parâmetros e segredos da AWS	Novas versões da Extensão do Lambda para parâmetros e segredos da AWS já estão disponíveis para todas as arquiteturas e regiões com suporte. Além disso, foi adicionado suporte à região Ásia-Pacífico (Taipei) (ap-east-2).	26 de junho de 2025
Novo tópico: Integração do Change Calendar com o Amazon Eventbridge	O tópico Integração do Change Calendar com o Amazon EventBridge fornece informações sobre essa integração nas seguintes categorias: • Tempo e confiabilidade dos eventos • Considerações sobre modificações e tempo dos eventos • Eventos adjacentes e sobrepostos • Práticas recomendadas	25 de junho de 2025
Atualização de suporte ao runtime para a ação de automação "aws:executeScript"	O parâmetro `Runtime` na ação de automação aws:executeScript agora oferece suporte a scripts do PowerShell 7.4 (`dotnet8`).	24 de junho de 2025

O SSM Agent e o Patch Manager agora oferecem suporte ao RHEL 10 e ao Ubuntu Server 25.04	Agora é possível gerenciar computadores Red Hat Enterprise Linux 10 e Ubuntu Server 25.04 com o Systems Manager, incluindo a aplicação de patches em nós gerenciados via Patch Manager. Para obter listas completas das versões do SOs compatíveis, consulte os seguintes tópicos: • Sistemas operacionais compatíveis com o Systems Manager • Sistemas operacionais compatíveis com o Patch Manager	23 de junho de 2025

Atualizações em tópicos sobre o acesso a nós just-in-time	As seguintes atualizações foram feitas no conteúdo do capítulo Acesso a nós just-in-time usando o Systems Manager: • O suporte à autenticação foi esclarecido para acesso a nós just-in-time. Para obter mais informações, consulte Configurar o acesso just-in-time com o Systems Manager. • Esclarecimento do requisito de marcar as chaves do AWS KMS usadas para criptografia e gravação RDP no acesso a nós just-in-time no Session Manager com a chave de tag `SystemsManagerJustInTimeNodeAccessManaged` e o valor de tag `true`. Para obter informações, consulte Atualizar as preferências de sessão de acesso a nós just-in-time e Configurar a criptografia de bucket do S3 para gravações de RDP. • Atualização de Configurar notificações para solicitações de acesso just-in-time com detalhes adicionais de configuração.	17 de junho de 2025

Gravação de RDP melhorada para acesso ao nó just-in-time	O acesso ao nó just-in-time do AWS Systems Manager agora oferece informações adicionais sobre configurações de gravação de RDP. Ao usar a criptografia do AWS KMS em buckets do S3 nos quais as gravações são armazenadas, é necessário fornecer acesso à chave do KMS utilizada para criptografia do bucket para a entidade principal do serviço do `ssm-guiconnect` . Esta versão também apresenta um novo status de gravação, `ProcessingError` , indicando falhas intermediárias durante o processamento de vídeo que podem ser originadas por falhas em dependências de serviços ou pela ausência de permissões. Para mais informações, consulte Gravação de conexões RDP.	13 de junho de 2025
Informações atualizadas sobre a seleção e a instalação de patches no Patch Manager	Foram adicionadas informações para explicar como o Patch Manager trata pacotes obsoletos durante a seleção e instalação de patches. Para mais informações, consulte Como os patches de segurança são selecionados e Como os patches são instalados.	12 de junho de 2025

Atualização da política gerenciada para Quick Setup: AWSQuickSetupDeploymentRolePolicy	O Systems Manager atualizou a política gerenciada `AWSQuickSetupDeploymentRolePolicy` , de forma que o Quick Setup seja capaz de criar associações utilizando essa política gerenciada no lugar de políticas em linha. Para obter mais informações, consulte Atualizações do Systems Manager para políticas gerenciadas da AWS.	10 de junho de 2025
Apresentação expandida para o Patch Manager	Para apresentar melhor o Patch Manager a possíveis clientes e usuários, adicionamos as três seções a seguir ao capítulo Patch Manager: • Como o Patch Manager beneficia minha organização? • Quem deve usar o Patch Manager? • Quais são os principais recursos do Patch Manager?	21 de maio de 2025

Política gerenciada atualizada para Quick Setup: AWSQuickSetupSSMDeploymentRolePolicy	O Systems Manager atualizou a política gerenciada `AWSQuickSetupSSMDeploymentRolePolicy` para fornecer permissões para marcar perfis do IAM e recursos do Lambda criados para o console unificado. Para obter mais informações, consulte Atualizações do Systems Manager para políticas gerenciadas da AWS.	7 de maio de 2025
Opções de configuração de recursos para o console unificado	O Systems Manager agora fornece opções de configuração de recursos ao configurar o console unificado. Essas configurações de recursos permitem que você escolha se deseja definir a Configuração de gerenciamento de host padrão (DHMC), a coleta de metadados de inventário e as atualizações automáticas do SSM Agent. Para obter mais informações, consulte Configurar o console unificado do Systems Manager para uma organização e Configurar o console unificado do Systems Manager para uma única conta e região.	7 de maio de 2025

Política gerenciada atualizada para Quick Setup: AWSSystemsManagerJustInTimeNodeAccessRolePropagationPolicy	O Systems Manager atualizou a política gerenciada `AWSSystemsManagerJustInTimeNodeAccessRolePropagationPolicy` para fornecer permissões para marcar um recurso compartilhado pelo AWS Resource Access Manager para acesso a nós just-in-time. Para obter mais informações, consulte Atualizações do Systems Manager para políticas gerenciadas da AWS.	29 de abril de 2025
Política gerenciada atualizada para Quick Setup: AWSQuickSetupManageJITNAResourcesExecutionPolicy	O Systems Manager atualizou a política gerenciada `AWSQuickSetupManageJITNAResourcesExecutionPolicy` para fornecer permissões para marcar perfis do IAM criados para acesso a nós just-in-time. Para obter mais informações, consulte Atualizações do Systems Manager para políticas gerenciadas da AWS.	29 de abril de 2025

Systems Manager introduz o acesso a nós just-in-time	O AWS Systems Manager agora é compatível com um método de acesso just-in-time para conexão com nós gerenciados. Esse novo método de conexão com nós gerenciados ajuda a melhorar a segurança de seus nós, fornecendo acesso temporário e com limite de tempo aos nós somente quando necessário, eliminando a necessidade de permissões prolongadas. Além disso, o Systems Manager fornece gravação de sessões para sessões RDP em nós do Windows Server para ajudar você a atender aos requisitos de conformidade, realizar análises de causa raiz e muito mais. Para obter mais informações, consulte Just-in-time node access using Systems Manager.	29 de abril de 2025
Novas políticas gerenciadas para acesso a nós just-in-time	O Systems Manager lançou várias novas políticas gerenciadas para oferecer suporte ao acesso a nós just-in-time. Para obter mais informações, consulte Atualizações do Systems Manager para políticas gerenciadas da AWS.	29 de abril de 2025

Novas versões da extensão do Lambda para parâmetros e segredos da AWS	Novas versões da Extensão do Lambda para parâmetros e segredos da AWS já estão disponíveis para todas as arquiteturas e regiões com suporte.	23 de abril de 2025
Novo tópico: “O que é conformidade no Patch Manager?”	A conformidade de patches para os nós gerenciados em sua frota não é definida pela AWS ou por terceiros . O novo tópico Em que consiste a conformidade do Patch Manager? esclarece que a conformidade de patches para um nó gerenciado pelo Systems Manager significa que todos os patches foram instalados no nó que atendem aos critérios de aprovação especificados na lista de referência de patches associada. A conformidade é definida por você, nas regras que você configura em uma lista de referência de patches ou em uma lista de referência de patches predefinida fornecida pela AWS, que você para suas operações de aplicação de patches.	11 de abril de 2025

Novas versões da extensão do Lambda para parâmetros e segredos da AWS	Novas versões da Extensão do Lambda para parâmetros e segredos da AWS já estão disponíveis. Além disso, foi adicionado suporte à extensão para as seguintes regiões: • Ásia-Pacífico (Tailândia) (ap-southeast-7) • México (Centro) (mx-centr al-1)	3 de abril de 2025
Novo tópico: configuração de permissões para Compliance	Como prática recomendada de segurança, recomenda mos que você atualize o perfil (IAM) AWS Identity and Access Management usado pelos nós gerenciados para restringir a capacidade do nó de usar a ação da API PutComplianceItems. Essa ação da API registra um tipo de conformidade e outros detalhes de conformidade em um recurso designado, como uma instância do Amazon EC2 ou um nó gerenciado. Para obter mais informaçõ es, consulte Configuração de permissões para Compliance.	11 de março de 2025

Tópico atualizado: revisão de insights do nó	No console unificado do Systems Manager, a página Revisar insights do nó informa o número de instâncias do EC2 gerenciadas e não gerenciadas em sua organização ou conta. Esclarecemos que o Systems Manager inclui uma instância do EC2 na contagem "Não gerenciada" quando está em um estado `Stopped`. Para obter mais informações, consulte O que é uma instância não gerenciada?.	25 de fevereiro de 2025
Tópico atualizado: Verificar a assinatura do SSM Agent	Os pacotes do instalador Debian e RPM do agente do AWS Systems Manager (SSM Agent) para instâncias Linux são assinados criptograficamente. Use a chave pública para verificar se o arquivo de download do agente é original e não modificado. Se os arquivos sofrerem quaisquer danos ou alterações, a verificação falhará. Você pode verificar a assinatura do pacote instalador usando RPM ou GPG. O tópico inclui uma nova chave pública para SSM Agent. Para obter mais informações, consulte Verificar a assinatura do SSM Agent.	14 de fevereiro de 2025

Novo tópico: Verificar a assinatura do plug-in do Session Manager	Os pacotes do instalador RPM e Debian do plug-in do Session Manager para instâncias Linux são assinados criptograficamente. É possível usar uma chave pública para verificar se o binário e o pacote do plug-in são originais e não modificados. Se o arquivo for alterado ou danificado, a verificação falhará. A assinatura do pacote do instalador pode ser verificada com a ferramenta GNU Privacy Guard (GPG). Para obter mais informações, consulte Verificar a assinatura do plug-in do Session Manager.	13 de fevereiro de 2025
Atualização do plug-in do Session Manager	O plug-in do Session Manager foi atualizado recentemente com os seguintes aprimoramentos: Atualização da versão do Go para 1.23 no Dockerfile. Atualização da etapa de configuração da versão no arquivo README. Para obter mais informações sobre o plug-in do Session Manager, consulte Instalar o plug-in do Session Manager para a AWS CLI.	6 de fevereiro de 2025

Cotas de TPS publicadas para ações da API do Automation	Publicamos as cotas de transações por segundo (TPS), anteriormente conhecidas como limitadas , para várias ações de API relacionadas ao Systems Manager Automation. Para obter informações, consulte as cotas de serviço do Systems Manager no Referência geral da Amazon Web Services.	21 de janeiro de 2025
Mudança de terminologia: recursos do Systems Manager renomeados para "ferramentas"	Até agora, as unidades funcionais que compõem o Systems Manager eram chamadas de recursos, como o recurso Automation, o recurso Parameter Store e assim por diante. Essas unidades funcionais agora são chamadas de ferramentas, como a ferramenta Automation e a ferramenta Parameter Store. Este guia do usuário foi atualizado para refletir a alteração.	10 de janeiro de 2025

Suporte do Patch Manager a versões adicionais do macOS (AMIs do Amazon EC2)	O Patch Manager agora oferece suporte às versões 12.7, 13.6, 13.7 e 15.x do macOS. Para obter uma lista completa de sistemas operacionais e versões compatíveis com o Patch Manager, consulte Sistemas operacionais compatíveis com o Patch Manager.	8 de janeiro de 2025
Suporte do SSM Agent e do Patch Manager a versões adicionais: CentOS Stream, Ubuntu Server e Windows Server	O Patch Manager agora oferece suporte ao CentOS Stream 9, Ubuntu Server 24.10 e Windows Server 2025. O SSM Agent agora oferece suporte ao Ubuntu Server 24.10 e Windows Server 2025. (O suporte do agente ao CentOS Stream 9 já havia sido lançado.) Para obter listas completas das versões do SOs compatíveis, consulte os seguintes tópicos: • Sistemas operacionais compatíveis com o Systems Manager • Sistemas operacionais compatíveis com o Patch Manager	22 de novembro de 2024

Novo tópico: Criptografia do AWS KMS para parâmetros SecureString do Parameter Store	Saiba como o AWS Systems Manager Parameter Store usa o AWS Key Management Service para criptografar os valores dos parâmetros `SecureString` no Parameter Store no seguinte tópico: • Criptografia do AWS KMS para parâmetros SecureString no Parameter Store	22 de novembro de 2024
Políticas gerenciadas novas e atualizadas para o Systems Manager	Para oferecer suporte aos novos recursos do Systems Manager, estamos lançando várias novas políticas gerenciadas para oferecer suporte às novas configurações e operações do Systems Manager, bem como atualizando outras políticas gerenciadas. Para obter mais informações, consulte Atualizações do Systems Manager para políticas gerenciadas pela AWS.	21 de novembro de 2024

Uma experiência de gerenciamento de nós nova e simplificada para o Systems Manager	O AWS Systems Manager lançou uma nova experiênc ia de console unificada para gerenciar nós em grande escala entre contas e regiões. Agora é possível ver todos os nós gerenciados e não gerenciados em nas regiões e Contas da AWS das suas organizações em um único lugar. Também é possível identificar, diagnosticar e corrigir nós não gerenciados. O Systems Manager também está integrado ao Amazon Q Developer (Amazon Q), o que amplia sua capacidade de ver e controlar os nós de qualquer lugar no AWS Managemen t Console inserindo prompts em linguagem natural. Com essa versão, agora você também pode usar o AWS Organizations para permitir que uma conta de administr ador delegado gerencie nós em toda a organização a partir de um ponto de vista central. Para obter mais informações, consulte os tópicos a seguir. • O que é AWS Systems Manager? • Configurar o AWS Systems Manager	21 de novembro de 2024

	• Concluir tarefas de nós com o AWS Systems Manager	
Correção de bug do plug-in do Session Manager	O plug-in do Session Manager foi atualizado recentemente com a seguinte correção de bug: Reversão de alteração que adicionava credenciais às solicitações do OpenDataChannel.	20 de novembro de 2024
Aprimoramentos no plug-in do Session Manager	Esta versão de runtime foi descontinuada em 20/11/2024. O plug-in do Session Manager foi atualizado recentemente com os seguintes aprimoramentos. • Adição de credenciais às solicitações do OpenDataChannel. • Atualização dos pacotes dependentes `testify` e `objx`.	6 de novembro de 2024
Suporte a versão de sistema operacional adicional para o macOS	O Systems Manager agora oferece suporte à versão 15. x (Sequoia) do sistema operacional macOS (somente instâncias do EC2). Para obter uma lista de todos os sistemas operacionais e versões compatíveis, consulte Sistemas operacionais compatíveis com Systems Manager.	6 de novembro de 2024

Patch Manager: atualizações em formatos de nomes de pacotes com suporte para listas de patches aprovados e rejeitados	Para vários sistemas operacionais, atualizamos e expandimos as listas de formatos para nomes de pacotes que você pode especificar nas listas Patches aprovados e Patches rejeitados em suas listas de referência de patches. Para obter mais informações, consulte Formatos de nome de pacote para Amazon Linux 1, Amazon Linux 2, Amazon Linux 2022, Amazon Linux 2023, CentOS, Oracle Linux e Red Hat Enterprise Linux (RHEL).	1.º de novembro de 2024
Suporte adicional à versão operacional para Patch Manager	O Patch Manager agora oferece suporte a versões adicionais do Oracle Linux (8.10 e 9.4) e Ubuntu Server (23.10 e 24.04). Para obter listas dos sistemas operacionais e versões com suporte, consulte os seguintes tópicos: • Sistemas operacionais compatíveis com o Systems Manager • Sistemas operacionais compatíveis com o Patch Manager	1.º de novembro de 2024

Suporte do SSM Agent a versões adicionais: CentOS Stream, Oracle Linux e Ubuntu Server	O SSM Agent agora oferece suporte ao CentOS Stream 9, Oracle Linux 8.10 e 9.4 e Ubuntu Server 24.04 LTS, além das versões anteriores compatíveis. Para obter uma lista completa de sistemas operacionais e versões compatíveis para o Systems Manager, consulte Sistemas operacionais compatíveis com Systems Manager.	25 de outubro de 2024
Aprimoramento de plug-ins do Session Manager	Adição de suporte para passar a versão do plug-in com solicitações de OpenDataC hannel.	10 de outubro de 2024
Novo: visualize detalhes das conexões RDP feitas via Fleet Manager	Agora você pode visualiza r informações sobre as conexões do Remote Desktop Protocol que foram feitas pelos usuários em sua Conta da AWS. Para obter informaçõ es, consulte Visualizar informações sobre conexões atuais e concluídas.	10 de outubro de 2024

O Patch Manager agora oferece suporte ao SLES versão 15.6	O suporte de patches para SUSE Linux Enterprise Server (SLES) 15.6 foi lançado. Agora você pode aplicar patches em máquinas SLES 15.6 usando o Patch Manager. Para obter uma lista completa de sistemas operacionais e versões compatíveis com o Patch Manager, consulte Sistemas operacionais com suporte para o Patch Manager.	29 de setembro de 2024

Novas versões da extensão do Lambda para parâmetros e segredos da AWS	Novas versões da Extensão do Lambda para parâmetros e segredos da AWS já estão disponíveis. O suporte a todas as arquiteturas foi introduzido na região Ásia-Pacífico (Malásia) (ap-southeast-5). Além disso, foi adicionado suporte ao ARM64 e à extensão de arquitetura Mac with Apple silicon para as seguintes regiões: • Ásia-Pacífico (Hyderabad) (ap-south-2) • Ásia-Pacífico (Melbourne) (ap-southeast-4) • Oeste do Canadá (Calgary) ca-west-1 • Europa (Zurique) (eu-central-2) • Europa (Espanha) (eu-south-2) • Oriente Médio (EAU) (me-central-1) • China (Pequim) (cn-north-1) • China (Ningxia) (cn-northwest-1) • Israel (Tel Aviv) (il-central-1) • AWS GovCloud (Leste dos EUA) (us-gov-east-1) • AWS GovCloud (Oeste dos EUA) (us-gov-wast-1)	19 de setembro de 2024

Novo tópico: Configuração de permissões para janelas de manutenção usando a AWS CLI	O tópico Configuração de permissões para janelas de manutenção usando a AWS CLI fornece instruções para criar um perfil de serviço personalizado (e suas políticas) para executar tarefas da janela de manutenção em nome do usuário.	19 de agosto de 2024
O SSM Agent e o Patch Manager são compatíveis com versões adicionais: AlmaLinux, Oracle Linux e Rocky Linux	O SSM Agent e o Patch Manager agora oferecem suporte às versões 8.10, 9.3 e 9.4 do AlmaLinux ao Rocky Linux, e à versão 9.3 do Oracle Linux, além das versões que já tinham suporte anteriormente. Para obter listas completas das versões do SOs compatíveis, consulte os seguintes tópicos: • Sistemas operacionais compatíveis com o Systems Manager • Sistemas operacionais compatíveis com o Patch Manager	14 de agosto de 2024

Nova condição da política do IAM para o suporte de Parameter Store: ssm:Policies	Usando `ssm:Policies` uma condição recente de suporte para políticas do IAM, você pode impedir que Entidades criem ou atualizem parâmetros que incluam uma política de parâmetros. Para obter mais informações, consulte os tópicos a seguir. • ssm:Policies: impeça a criação ou a atualização de parâmetros que usam políticas de parâmetros • Chaves de condição para o AWS Systems Manager na Referência de Autorização de Serviço.	14 de agosto de 2024
Política gerenciada atualizada para Quick Setup: SSMQuickSetupRolePolicy	O Systems Manager atualizou a política gerenciada `SSMQuickSetupRolePolicy` para fornecer acesso a conjuntos de pilhas adicionais do AWS CloudFormation. Para obter mais informações, consulte Atualizações do Systems Manager para políticas gerenciadas da AWS.	13 de agosto de 2024

Suporte para provisionamento e gerenciamento de recursos do Systems Manager usando o Terraform	Adicionamos o HashiCorp Terraform à lista de integrações de terceiros compatíveis com o Systems Manager. O Terraform é uma ferramenta de software de infraestrutura como código (IaC) de código aberto que fornece um fluxo de trabalho de interface de linha de comandos (CLI) para gerenciar vários serviços em nuvem. Você pode usar o Terraform para provisionar e gerenciar vários recursos e fontes de dados comumente usados do Systems Manager. Para obter informações sobre essa e outras integrações de terceiros com o Systems Manager, consulte Integração com outros produtos e serviços.	1.º de agosto de 2024

Nova experiência de console e API da Quick Setup	A Quick Setup do Systems Manager lançou uma nova experiência de console e API. Agora você pode interagir com essa API usando o console, a AWS CLI, o AWS CloudFormation e os SDKs. Você pode aderir ao novo console usando o console de Quick Setup. Para obter mais informações sobre a integração à nova experiênciada Quick Setup, consulte Getting started with Quick Setup. Para obter mais detalhes sobre as operações de API disponíveis por meio da API da Quick Setup, consulte a Referência da API da Quick Setup.	1 de agosto de 2024

Novo tópico: opções de lista de patches rejeitados em listas de referência de patches personalizados	Para operações de aplicação de patch que usam uma lista de referência de patches personalizada no Patch Manager, esclarecemos o comportamento quando a ação `Allow as dependency` é atribuída a um patch adicionado à lista de patches rejeitados. Como o Windows Server não é compatível com o conceito de dependências de patches, os patches que ainda não estão instalados em um nó gerenciado são ignorados. Os patches que já estão instalados no nó recebem o status `INSTALLED_REJECTED`. Para obter mais informações, consulte Opções de lista de patches rejeitados em listas de referência de patches personalizados e Valores de conformidade de patches para outros sistemas operacionais.	23 de julho de 2024

Novo tópico: configurar o SSM Agent para usar com o Federal Information Processing Standard (FIPS, Padrão federal de processamento de informações)	Fornecemos instruções de configuração do SSM Agent para uso com o padrão FIPS (Federal Information Processing Standard). Para obter mais informações, consulte Configurar o SSM Agent para usar com o Federal Information Processing Standard (FIPS, Padrão federal de processamento de informações).	22 de julho de 2024
Atualização: suporte esclarecido para símbolos @ em nomes de usuário do Fleet Manager	Se um nome de usuário do IAM Identity Center contiver um ou mais símbolos @, o RDP do Fleet Manager desconsiderará o primeiro símbolo @ e todos os caracteres que o seguem, independentemente de @ introduzir ou não a parte do domínio de um endereço de e-mail. Para obter mais informações sobre caracteres compatíveis com nomes de usuário em conexões RDP do Fleet Manager, consulte Autenticar conexões da Área de Trabalho Remota.	21 de julho de 2024

Política gerenciada atualizada: AmazonSSMManagedEC2InstanceDefaultPolicy	O Systems Manager atualizou a política gerenciada pela `AmazonSSMManagedEC2InstanceDefaultPolicy` fornecendo IDs de declaração (Sids) em linha para esclarecer a finalidade de cada declaração de política. Para obter mais informações, consulte Atualizações do Systems Manager para políticas gerenciadas da AWS.	18 de julho de 2024

Alterações de nome em buckets gerenciados da AWS para operações de aplicação de patches do Patch Manager	A AWS possui e mantém vários buckets do Amazon S3 que o SSM Agent acessa durante a execução de várias operações de aplicação de patches do Patch Manager. Esses buckets do S3 são acessíveis publicamente e, por padrão, SSM Agent se conecta a eles usando chamadas HTTP. No entanto, se você estiver usando um endpoint da nuvem privada virtual (VPC) nas operações do Systems Manager, deverá fornecer permissão explícita em um perfil de instância do Amazon EC2 para o Systems Manager ou em um perfil de serviço para máquinas que não são do EC2 em um ambiente híbrido e multinuvem. Caso contrário, seus recursos não poderão acessar esses buckets públicos. Na maioria dos casos, estamos alterando os nomes desses buckets. Por exemplo, para operações de aplicação de patches, o bucket `aws-patchmanager-macos-us-east-2` é substituído por `aws-patchmanager-macos-us-east-2-552881074` , e o bucket `aws-ssm-us-east-2` é substituí	18 de julho de 2024

	do por `aws-patch-manager-us-east-2-552881074` . Para obter mais informações, consulte os tópicos a seguir. • Comunicações do SSM Agent com os buckets do S3 gerenciados pela AWS • Referência: buckets do Amazon S3 para operações de aplicação de patches	
Nova função vinculada a serviços do Quick Setup	O Systems Manager lançou uma nova função vinculada a serviços, `AWSServiceRoleForSSMQuickSetup` . O Systems Manager usa essa função para verificar a integridade da configuração dos recursos configurados usando o Quick Setup, para garantir o uso uniforme dos parâmetros e dos recursos provisionados e para corrigir os recursos quando o desvio é detectado. A política gerenciada associada a essa função é `SSMQuickSetupRolePolicy` . Para obter mais informações, consulte AWSServiceRoleForSSMQuickSetup service-linked role permissions for Systems Manager.	3 de julho de 2024

Novas políticas gerenciadas para tipos de configuração da Quick Setup	O Systems Manager lançou mais 12 novas políticas gerenciadas para oferecer suporte a vários tipos e processos de configuração da Quick Setup. Para obter mais informações, consulte Atualizações do Systems Manager para políticas gerenciadas da AWS.	3 de julho de 2024
Suporte ao RHEL 8.10 e 9.4	O Systems Manager e o Patch Manager agora oferecem suporte às versões 8.10 e 9.4 do Red Hat Enterprise Linux. Para obter mais informações, consulte Sistemas operacionais e tipos de máquinas compatíveis e Sistemas operacionais compatíveis com o Patch Manager.	26 de junho de 2024
Suporte do Patch Manager às versões 8.8 e 8.9: AlmaLinux, Oracle Linux e Rocky Linux	O Patch Manager agora oferece suporte às versões 8.8 e 8.9 do AlmaLinux, Oracle Linux e Rocky Linux, além das versões 8.x mais antigas. Para obter uma lista completa de sistemas operacionais e versões compatíveis com o Patch Manager, consulte Sistemas operacionais compatíveis com Patch Manager.	17 de junho de 2024

Novos parâmetros públicos para AMIs do Amazon EC2 no macOS	Parâmetros públicos foram lançados para dar suporte a Amazon Machine Images para instâncias macOS do Amazon Elastic Compute Cloud. Para obter mais informações, consulte os tópicos a seguir. • Localizar parâmetros públicos • Chamar parâmetros públicos da AMI para macOS • Executar uma instância Mac no Guia do usuário do Amazon EC2 • Instâncias do Mac do Amazon EC2 no Guia do usuário do Amazon EC2 • Usar parâmetros do AWS Systems Manager em vez de IDs de AMI no Guia do usuário do Amazon EC2 Auto Scaling	17 de junho de 2024

Atualização: disponibilidade regional do caminho de parâmetro `/aws/service/global-infrastructure`	Esclarecemos de quais regiões comerciais o caminho do parâmetro público `/aws/service/global-infrastructure` pode ser consultado e como executar uma consulta para o caminho se você estiver trabalhando em uma Região da AWS comercial diferente. Para obter informações, consulte Chamar parâmetros públicos para serviços, regiões, endpoints, zonas de disponibilidade, zonas locais e zonas do Wavelength da AWS.	12 de junho de 2024
Novo: capítulo de exemplos de código	Um novo capítulo, Exemplos de código para o Systems Manager usando AWS SDKs, fornece exemplos em diferentes linguagens de SDK para ensinar a trabalhar com o serviço Systems Manager.	8 de maio de 2024

Alterações no suporte a endpoints `ec2messages:*`	Para Regiões da AWS lançadas em 2024 ou posteriormente, os endpoints `ec2messages:*` não são aceitos pelo SSM Agent para enviar informações de status e execução de volta para o serviço Systems Manager. As contas nessas regiões devem usar `ssmmessages:*`. Nas regiões lançadas antes de 2024, ainda há suporte a `ssmmessages:*` e `ec2messages:*`, mas recomendamos usar somente o endpoint `ssmmessages:*` (Amazon Message Gateway Service) agora. É possível remover as permissões de `ec2messages:*` com segurança das suas políticas. Para obter mais informações, consulte Como trabalhar com o SSM Agent e Operações de API relacionadas ao agente (endpoints ssmmessages e ec2messages).	3 de maio de 2024
Runtimes adicionais disponíveis para execução de scripts em runbooks do Automation	A ação `aws:executeScript` agora oferece suporte aos runtimes Python 3.9, 3.10 e 3.11. Para obter mais informações sobre como usar essa ação, consulte `aws:executeScript`.	23 de abril de 2024

Suporte às versões 8.8 e 8.9: AlmaLinux, Oracle Linux e Rocky Linux	O Systems Manager agora oferece suporte às versões 8.8 e 8.9 do AlmaLinux, Oracle Linux e Rocky Linux, além das versões 8.x mais antigas. Para obter uma lista completa de sistemas operacionais e versões compatíveis, consulte Sistemas operacionais compatíveis com Systems Manager.	22 de abril de 2024

Patch Manager: alteração para o status de patch "INSTALLED_PENDING_REBOOT"	Anteriormente, somente os patches instalados pelo Patch Manager podiam ser marcados como `INSTALLED_PENDING_REBOOT`. Os patches instalados fora do Patch Manager nunca recebiam esse status. Agora, `INSTALLED_PENDING_REBOOT` pode ser aplicado a qualquer patch que tenha sido aplicado a um nó gerenciado desde a última reinicialização. Isso inclui patches instalados pelo Patch Manager com a opção `NoReboot` selecionada e para patches instalados fora do Patch Manager após a reinicialização mais recente do nó. Para obter descrições de todos os valores de status de patch do Patch Manager, consulte Compreender os valores do estado de conformidade do patch.	16 de abril de 2024
Suporte ao RHEL 8.9 e 9.3	O Systems Manager, incluindo o Patch Manager, agora oferece suporte ao Red Hat Enterprise Linux (RHEL) versões 8.9 e 9.3, além das versões anteriores 8.x e 9.x.	26 de março de 2024

Atualização de tópico: Políticas gerenciadas pela AWS para o AWS Systems Manager	O tópico Políticas gerenciadas pela AWS para o AWS Systems Manager fornecia informações sobre as quatro políticas gerenciadas do Systems Manager que foram introduzidas ou atualizadas desde 12 de março de 2021. Adicionamos uma seção a este tópico com informações sobre as outras 12 políticas gerenciadas para uso com o Systems Manager que foram criadas ou atualizadas pela última vez antes dessa data. Para obter detalhes, consulte Políticas gerenciadas adicionais para o Systems Manager.	1.º de março de 2024

O Parameter Store agora oferece suporte ao compartilhamento entre contas	Agora você pode compartilhar parâmetros avançados de forma segura e eficiente em suas AWS ou dentro da sua organização da Contas da AWS configurando o compartilhamento de recursos. O compartilhamento de recursos permite centralizar o gerenciamento da configuração de aplicações e reduzir a sobrecarga operacional de compartilhar os parâmetros com cada conta que você possui. Os parâmetros podem ser compartilhados entre contas usando o console do Parameter Store, o console do AWS RAM ou a AWS CLI. Para obter mais informações, consulte Trabalhar com parâmetros compartilhados.	21 de fevereiro de 2024
Aprimoramento de ações do Automation	Agora é possível usar as propriedades `onFailure` e `isCritical` com a ação `aws:approve`. Para obter mais informações sobre a ação `aws:approve`, consulte aws:approve — Pausar uma automação para aprovação manual.	12 de fevereiro de 2024

Suporte adicional à versão operacional para Patch Manager	Adicionamos à lista de versões de sistema operacional compatíveis com o Patch Manager. O suporte também foi adicionado para: • Debian Server 11.x e 12.x • macOS 14.x (Sonoma) • SUSE Linux Enterprise Server (SLES) 15.5 • Ubuntu Server 23.04	4 de janeiro de 2024
Configurar atualizações automatizadas do SSM Agent usando o console do Application Manager	Agora, você pode usar o console do Application Manager para automatizar atualizações do SSM Agent para as instâncias da sua aplicação. Para obter mais informações, consulte Trabalhar com instâncias da sua aplicação.	21 de dezembro de 2023
Atualização do processo de registrar máquinas não Amazon EC2 em ambientes híbridos e de várias nuvens	O Systems Manager agora oferece o `ssm-setup-cli` para ajudar você a registrar máquinas não Amazon Elastic Compute Cloud (Amazon EC2) em ambientes híbridos e de várias nuvens. Para obter mais informações, consulte Como instalar o SSM Agent em nós híbridos do Linux e Como instalar o SSM Agent em nós híbridos do Windows.	20 de dezembro de 2023

Gerenciar volumes do Amazon EBS com o uso do Fleet Manager	Agora é possível usar o Fleet Manager, uma ferramenta do AWS Systems Manager, para gerenciar volumes do Amazon Elastic Block Store em suas instâncias gerenciadas. Por exemplo, é possível inicializar um volume do EBS, formatar uma partição e montar o volume para disponibilizá-lo para uso. Para obter mais informações, consulte Gerenciamento de volumes do EBS.	14 de dezembro de 2023
Aprimoramento de plug-ins do Session Manager	Adicionado suporte para transmitir uma resposta da API StartSession como variável de ambiente a session-manager-plugin.	4 de dezembro de 2023
Nova experiência de design visual para runbooks de automação	Agora você pode criar e editar runbooks usando uma nova experiência de design visual desenvolvida pelo Systems Manager Automation. A experiência de design visual fornece uma interface simples de arrastar e soltar para que você possa criar e editar runbooks com mais facilidade. Para obter mais informações, consulte Experiência de design visual para runbooks de automação.	26 de novembro de 2023

Novas ações, elementos de dados e aprimoramentos funcionais do Systems Manager Automation para runbooks	Agora você pode percorrer várias ações em um runbook usando a ação `aws:loop`. Essa nova ação oferece suporte a loops estilo `do while` e `for each`. Além disso, usando o novo elemento de dados variáveis, é possível definir, referenciar e atualizar valores dinamicamente dentro do contexto de um runbook. Para atualizar o valor de uma variável no seu runbook, use a nova ação `aws:updateVariable`. O Automation também adicionou suporte a conversões de tipos de dados dinâmicos para saídas. Isso significa que, se o valor de uma saída não corresponder ao tipo de dados que você especificou, o Automation tentará converter o tipo de dados. Por exemplo, se o valor retornado for um `Integer`, mas o `Type` especificado for `String`, o valor final da saída será um valor `String`. Finalmente, o Automation agora oferece suporte a expressões de filtro JSONPath para seletores. Para obter mais informações, consulte os tópicos a seguir.	17 de novembro de 2023

	• aws:loop: itera nas etapas de uma automação • aws:updateVariable: atualiza um valor para uma variável do runbook • Elementos de dados e parâmetros: elementos de dados de nível superior • Uso de saídas de ações como entradas. • Uso de JSONPath em runbooks.	
Suporte regional atualizado para conexões do Remote Desktop Protocol (RDP)	O Fleet Manager Remote Desktop, baseada no Amazon DCV, fornece conectividade segura às suas instâncias do Windows Server diretamente do console do Systems Manager. As três regiões adicionais a seguir foram habilitadas para conexões do Fleet Manager Remote Desktop: • África (Cidade do Cabo) (af-south-1) • Ásia-Pacífico (Jacarta) (ap-southeast-3) • Israel (Tel Aviv) (il-central-1)	15 de novembro de 2023

Patch Manager: suporte expandido da versão do sistema operacional para RHEL e macOS	Agora Patch Manager oferece suporte às seguintes versões de sistemas operacionais: • Red Hat Enterprise Linux: versão 8.8 • macOS: 11.5–11:7 (Big Sur) • macOS: 12.0—12.6 (Monterey) • macOS: 13.0—13.5 (Ventura)	23 de outubro de 2023
Nova OpsCenter API - ExcluirOpsItem	Agora OpsCenter oferece a OpsItem API Delete para excluir indivíduos OpsItems. Para obter mais informações, consulte DeleteOpsItem na Referência de API do AWS Systems Manager.	20 de outubro de 2023
Novo tipo de configuração de Quick Setup: atualizações do SSM Agent para toda a organização	O novo tipo de configuração Configuração padrão de gerenciamento de host possibilita que um administrador da organização, conforme definido emAWS Organizations, solicite verificações e atualizações automáticas de todas as instâncias do EC2 SSM Agent nas contas e regiões da organização. Para obter mais informações, consulte Gerenciamento de host padrão para uma organização.	16 de outubro de 2023

Novo formato de título e descrição para OpsItems criado pelo CloudWatch Application Insights	O título e a descrição de OpsItems criados pelo CloudWatch Application Insights serão alterados para um formato aprimorado em 16 de outubro de 2023. Para visualizar o novo formato, consulte Amazon CloudWatch Application Insights.	29 de setembro de 2023

Suporte para várias resoluções de tela em conexões RDP do Fleet Manager	Ao se conectar a nós gerenciados do Windows Server usando a opção Protocolo de Área de Trabalho Remota (RDP) no Fleet Manager, já é possível escolher a resolução da tela. Antes, todas as conexões usavam uma resolução fixa de 720P (1366 x 768). Você já pode escolher entre as seguintes opções para cada conexão: • Adaptar automaticamente (determina a resolução ideal com base no tamanho da tela detectada) • 1920 x 1080 • 1400 x 900 • 1366 x 768 • 800 x 600 Para obter mais informações, consulte Conectar-se a um nó gerenciado usando a Área de Trabalho Remota.	22 de setembro de 2023

Novo tópico: IDs aleatórios da lista de referência de patches em operações de política de patches	Adicionamos conteúdo para descrever como as políticas de patch da Quick Setup usam o parâmetro `Baseline0verride` no documento do SSM Command `AWS-RunPatchBaseline` para gerar IDs aleatórios para as listas de referência de patches sempre que uma operação de política de patch é executada. Para obter mais informações, consulte IDs aleatórios da lista de referência de patches em operações de política de patches.	22 de setembro de 2023
Um novo insight operacional para gerenciar OpsItems	O OpsCenter agora inclui um insight operacional chamado Recursos que geram mais OPSitems. Um insight desse tipo é gerado quando um recurso da AWS tem mais de dez OpsItems abertos. Use esse insight para localizar recursos com problemas. Use o runbook `AWS-BulkResolveOpsItems` de dentro de um insight para resolver rapidamente OpsItems associados a um recurso. Para obter mais informações, consulte Analisar insights operacionais para reduzir OpsItems.	22 de setembro de 2023

Chave pública GPG atualizada	Foi criada uma nova chave pública para verificar a assinatura do SSM Agent. Para obter mais informações, consulte Verificar a assinatura do SSM Agent.	5 de setembro de 2023
Foi adicionado suporte para outras versões do AlmaLinux , Oracle Linux, RHEL e Rocky Linux	As listas de sistemas operacionais compatíveis com o AWS Systems Manager e o Patch Manager foram atualizadas para refletir o suporte a estas versões adicionais do sistema operacional: • AlmaLinux: 9.2 • Oracle Linux: 8.7 e 9.2 • Red Hat Enterprise Linux (RHEL): 8.7, 9.1 e 9.2 • Rocky Linux: 8.6 e 8.7, 9.0-9.2	30 de agosto de 2023

O OpsCenter adicionou suporte para formatação Markdown no campo de descrição do OpsItem.	O OpsCenter já tem suporte para formatação Markdown no campo de descrição do OpsItem. Há suporte para os seguintes tipos de formatação Markdown: • Parágrafos • Espaçamento entre linhas • Linhas horizontais • Títulos • Formatação de texto • Links • Listas Para obter mais informações, consulte Usar o Markdown no console no Guia de conceitos básicos do AWS Management Console.	18 de agosto de 2023
Novas versões da extensão do Lambda para parâmetros e segredos da AWS	Já estão disponíveis novas versões da extensão do Lambda para parâmetros e segredos da AWS. Além disso, foi adicionado suporte a extensões para as regiões da Ásia-Pacífico (Melbourne) (ap-southeast-4) e Israel (Tel Aviv) (il-central-1) (e somente arquiteturas x86_64 e x86). Para obter mais informaçõ es, consulte Using Parameter Store parameters in AWS Lambda functions.	16 de agosto de 2023

Atualização: foram adicionadas informações sobre as permissões necessárias para buckets de políticas de patch da Quick Setup	Quando você cria uma política de patch, a Quick Setup cria um bucket do Amazon S3 que contém um arquivo chamado `baseline_overrides.json`. Esse arquivo armazena informações sobre as listas de referência de patches que você especific ou para a política de patch. Ao configurar a política de patch, você tem a opção de marcar a caixa de seleção Adicionar políticas do IAM necessári as aos perfis de instância existentes anexados às suas instâncias. Se você escolher não selecionar essa opção, deverá fornecer manualmen te determinados recursos com permissões para acessar esse bucket, ou suas operações de política poderão falhar. Para obter mais informações, consulte os tópicos a seguir. • Permissões para o bucket do S3 da política de patch • Problema: erro “Invoke-PatchBaselineOperation: Access Denied” ou erro “Unable to download file from S3” para `baseline_overrides.json`	6 de julho de 2023

Use a Quick Setup para configurar o OpsCenter para gerenciamento do OpsItem de várias contas	A Quick Setup para o OpsCenter ajuda a concluir as seguintes tarefas para gerenciar OpsItems entre contas: • Especificar a conta de administrador delegado • Criar políticas e perfis do AWS Identity and Access Management (IAM) necessários • Especificar uma organizaç ão do AWS Organizations ou um subconjunto de contas de membro em que um administrador delegado pode gerenciar OpsItems entre contas Para obter mais informações, consulte (Opcional) Configura r o OpsCenter para gerenciar OpsItems entre contas usando a Quick Setup.	19 de junho de 2023

Atualize os agentes de execução do Amazon EC2 usando a Quick Setup	Agora você pode permitir que o Systems Manager verifique a cada 30 dias se há uma nova versão do agente de execução instalada na instância. Se uma nova versão estiver disponível, o Systems Manager atualizará o agente na instância. Para obter mais informações, consulte Quick Setup Host Management.	19 de junho de 2023
O Patch Manager agora oferece suporte ao Ubuntu Server 22.04 LTS	Agora é possível usar o Patch Manager para aplicar patches nos nós do Ubuntu Server 22.04 LTS. Como outras versões do Ubuntu Server com suporte, a versão 22.04 LTS usa a lista de referência de patches `AWS-UbuntuDefaultPatchBaseline` gerenciada pela AWS.	15 de maio de 2023

O Systems Manager já oferece suporte ao AlmaLinux, incluindo o Patch Manager	Agora você pode usar o Systems Manager para gerenciar os nós AlmaLinux 8.3-8.7; 9.0-9.1. Muitas das regras que se aplicam ao RHEL 8 para patches também se aplicam ao AlmaLinux . O AlmaLinux usa a nova `AWS-DefaultAlmaLinuxPatchBaseline` . Para obter mais informações, consulte os tópicos a seguir. • Instalar o SSM Agent manualmente em instâncias do AlmaLinux • Como os patches de segurança são selecionados • Como os patches são instalados • Como as regras de lista de referência de patches funcionam no AlmaLinux, no RHEL e no Rocky Linux.	8 de maio de 2023
Implante o agente EC2Launch v2 usando a Quick Setup	Você já pode implantar o agente EC2Launch v2 usando a Quick Setup. Para obter mais informações, consulte Implantar pacotes do Distributor com o Quick Setup.	13 de abril de 2023

O Systems Manager agora oferece suporte ao Amazon Linux 2023	O Systems Manager já oferece suporte ao novo tipo de instância do EC2 do Amazon Linux 2023 (AL2023), incluindo suporte para operações do Patch Manager. Muitas das regras de aplicação de patches que se aplicam ao Amazon Linux 2 também se aplicam ao Amazon Linux 2023. (O Patch Manager também continua oferecendo suporte à versão de pré-visualização do Amazon Linux 2022.) Para obter mais informações, consulte os tópicos a seguir. • Como os patches de segurança são selecionados • Como os patches são instalados • Como as regras de lista de referência de patches funcionam no Amazon Linux 1, Amazon Linux 2, Amazon Linux 2022 e Amazon Linux 2023	23 de março de 2023

Conteúdo de configuração revisado para as instâncias do Amazon EC2	Revisamos o conteúdo de configuração para as instâncias do Amazon EC2. Agora é recomendável usar a configuração de gerenciamento de host padrão recém-lançada para as permissões de instância. Para obter mais informações, consulte Configurar permissões de instância obrigatórias para o Systems Manager.	15 de fevereiro de 2023
Gerenciamento automático de instâncias com a configuração de gerenciamento de host padrão	Agora é possível gerenciar automaticamente as instâncias do Amazon EC2 em uma Região da AWS completa usando o Systems Manager. Para obter mais informações, consulte Configuração de gerenciamento de host padrão.	15 de fevereiro de 2023

Adição de documentos do SSM aos seus favoritos	Para ajudar você a encontrar documentos do SSM usados com frequência, é possível adicionar documentos aos seus favoritos. É possível adicionar até 20 documento s como favoritos por tipo de documento, Conta da AWS e Região da AWS. É possível escolher, modificar e visualiza r os favoritos no console do Systems Manager para Documentos. Para obter mais informações, consulte Como adicionar documentos aos seus favoritos.	7 de fevereiro de 2023
Implementação de controles de alteração para o Automatio n usando o Change Calendar	Ao integrar o Automation com o Change Calendar, você agora pode implement ar controles de alteração para todas as automações em sua Conta da AWS. Para obter mais informações, consulte Implementação de controles de alteração para o Automatio n.	24 de janeiro de 2023

Novo fluxo de trabalho de aprovação do Change Manager	O fluxo de trabalho de aprovação do Change Manager passou a oferecer suporte para aprovações por nível em vez de aprovações por linha. Anteriormente, cada aprovador adicionado a um nível de aprovação precisava aprovar uma solicitação de alteração. Caso contrário, o nível não seria aprovado. Agora, é possível especific ar quantas aprovações são necessárias para o nível e adicionar um número igual ou superior de aprovador es. Por exemplo, você pode requerer três aprovações para um nível, mas especificar até cinco aprovadores. Aprovaçõe s de quaisquer três desses aprovadores são suficientes para aprovar o nível. Para obter mais informações, consulte Sobre as aprovações em seus modelos de alteração .	23 de janeiro de 2023

Novo: configure a aplicação de patches para uma organização inteira usando uma política de patch no Quick Setup	Com o Quick Setup, uma ferramenta do Systems Manager, agora é possível criar políticas de patch baseadas no Patch Manager. Uma política de patch define a programação e a lista de referência de patches a serem usados na aplicação automátic a de patches nos seus nós gerenciados. Usando uma única configuração de política de patch, é possível definir a aplicação de patches para todas as contas em todas as regiões da sua organizaç ão, somente para as contas e regiões que você escolher ou para um único par de conta-região. Para obter mais informações, consulte os tópicos a seguir. • Usar políticas de patch da Quick Setup • Automatize a aplicação de patches em toda a organização usando uma política de patch do Quick Setup	22 de dezembro de 2022

O Application Manager integra-se ao Amazon EC2 para exibir informações sobre suas instâncias no contexto de uma aplicação.	O Application Manager exibe o estado da instância, o status e a integridade do Amazon EC2 Auto Scaling para uma aplicação selecionada em um formato gráfico. A guia Instâncias também inclui uma tabela com as informações a seguir para cada instância na sua aplicação.
	• Estado da instância (pendente, sendo interrompida, em execução, interrompida)
	• Status de ping para SSM Agent
	• Status e nome do runbook mais recente do Systems Manager Automation processado na instância
	• Uma contagem de alarmes do Amazon CloudWatch Logs por estado.
	  • `ALARM`: a métrica ou a expressão está fora do limite definido.
	  • `OK`: a métrica ou a expressão está dentro do limite definido.
	  • `INSUFFICIENT_DATA` : o alarme acabou de ser acionado, a métrica não está disponível ou não há dados suficientes para
		22 de dezembro de 2022

	a métrica determinar o estado do alarme. • Integridade do grupo do Auto Scaling para os grupos de escalabilidade automátic a pai e individual	
Programe a inicialização e a interrupção das suas instância s do Amazon EC2 usando o Quick Setup	Agora é possível implantar a solução Programador de recursos para automatizar a inicialização e a interrupção de suas instâncias do Amazon EC2 usando o Quick Setup. Para obter mais informaçõ es, consulte Programador de recursos.	19 de dezembro de 2022

O OpsCenter agora oferece suporte ao trabalho com OpsItems entre contas	O OpsCenter oferece suporte ao trabalho com OpsItems a partir de uma conta de gerenciamento (seja uma conta de gerenciamento do AWS Organizations ou uma conta de administr ador delegado do Systems Manager) e uma conta de membro durante uma sessão. Após configurados, os usuários podem realizar os seguintes tipos de ações: • Crie, visualize e atualize OpsItems em uma conta de membro • Visualize informações detalhadas sobre os recursos da AWS especific ados em OpsItems em uma conta de membro • Inicie runbooks do Systems Manager Automation para corrigir problemas com recursos da AWS em uma conta de membro Para obter mais informaçõ es, consulte Configuração do OpsCenter para o trabalho com OpsItems entre contas.	16 de novembro de 2022

Acompanhe os detalhes das solicitações de alteração do Change Manager usando o AWS CloudTrail Lake	Agora é possível usar um armazenamento de dados de eventos no AWS CloudTrail Lake para capturar e analisar detalhes sobre as solicitações de alteração executadas no Change Manager para sua organização ou conta. Essas informações incluem detalhes auditáveis sobre a identidade do usuário que criou a solicitação de alteração, o endereço IP de onde a solicitação foi feita, as Regiões da AWS nas quais as alterações foram feitas, os recursos afetados e outros. Para obter mais informações, consulte Monitoramento dos seus eventos de solicitação de alteração e Revisar detalhes, tarefas e cronogramas das solicitações de alteração.	11 de novembro de 2022

Controles adicionais de tarefas do Systems Manager usando alarmes do CloudWatch	Agora é possível implementar um controle adicional ao executar automações entre várias contas e regiões usando alarmes do CloudWatch. Ao aplicar um alarme de métrica ou composto do CloudWatch a uma automação, é possível controlar quando uma automação é interrompida com base na métrica que você definir. Para obter mais informações sobre como aplicar um alarme do CloudWatch a uma automação em execução ao longo de várias contas e regiões, consulte Execução de uma automação em várias regiões e contas (console)	9 de novembro de 2022

Atualizado: "Usar parâmetros do Parameter Store nas funções do AWS Lambda"	Fornecemos informações adicionais para ajudá-lo a usar a extensão do Lambda para parâmetros e segredos da AWS para recuperar valores de parâmetros e armazená-los em cache para uso futuro em funções do Lambda. O uso da extensão do Lambda pode reduzir custos diminuindo o número de chamadas de API para a Parameter Store. Para obter informações, consulte Usar parâmetros do Parameter Store em funções do AWS Lambda.	25 de outubro de 2022

Controles adicionais de tarefas do Systems Manager usando alarmes do CloudWatch	Agora, você pode implementar um controle adicional ao executar automações e comandos usando alarmes do CloudWatch. Também é possível adicionar um alarme do CloudWatch a uma automação ou comando quando ele for registrado com uma tarefa de janela de manutenção ou associação a State Manager. Ao aplicar um alarme composto do CloudWatch a uma automação ou comando, é possível controlar quando uma automação ou comando é interrompido com base na métrica que você definir. Para obter mais informações sobre como aplicar um alarme do CloudWatch a uma automação ou comando, consulte os seguintes procedimentos: • Como os patches de segurança são selecionados • Como os patches são instalados • Como as regras de lista de referência de patches funcionam no Amazon Linux 1, no Amazon Linux 2 e no Amazon Linux 2022.	26 de setembro de 2022

Controles adicionais de tarefas do Systems Manager usando alarmes do CloudWatch	Agora, você pode implementar um controle adicional ao executar automações e comandos usando alarmes do CloudWatch. Também é possível adicionar um alarme do CloudWatch a uma automação ou comando quando ele for registrado com uma tarefa de janela de manutenção ou associação a State Manager. Ao aplicar um alarme composto do CloudWatch a uma automação ou comando, é possível controlar quando uma automação ou comando é interrompido com base na métrica que você definir. Para obter mais informações sobre como aplicar um alarme do CloudWatch a uma automação ou comando, consulte os seguintes procedimentos: • Executar uma automação simples • Executar comandos no console • Criar uma associação • Atribuir tarefas a uma janela de manutenção	26 de setembro de 2022

Esclarecimento sobre os requisitos do nível de instâncias avançadas	Com base no feedback dos clientes, esclarecemos os cenários que exigem que você ative o nível de instâncias avançadas em Configurar níveis de instâncias.	21 de setembro de 2022
Implantar o Amazon CloudWatch Agent usando o Quick Setup	Agora, você pode implantar o agente do Amazon CloudWatch usando o Quick Setup. Para obter mais informações, consulte Implantar pacotes do Distributor com o Quick Setup.	20 de setembro de 2022

Agora, a chave “PatchGroup” é compatível com grupos de patches quando houver permissão para metadados de instâncias do EC2	Quando você permitir tags nos metadados da instância do EC2, as chaves de tag que você criar não devem conter espaços. Anteriormente, isso impedia que os clientes adicionassem algumas de suas instâncias do EC2 aos grupos de patches no Patch Manager, porque era necessário aplicar a chave de tag `Patch Group` às instâncias. Agora, o Patch Manager oferece suporte para `Patch Group` (com um espaço) e `PatchGroup` (sem espaço) como chave de tag a fim de identificar instâncias para um grupo de patches. Agora, é possível adicionar aos grupos de patches no Patch Manager as instâncias do EC2 que têm permissão para tags nos metadados da instância. Para obter mais informações, consulte Sobre grupos de patches.	31 de agosto de 2022

Novo tópico: “Como as datas de lançamento e atualização de pacote são calculadas”	Na lista de referência de patches gerenciados pela AWS, os novos patches são aprovados automatic amente 7 dias após serem lançados ou atualizados. Na lista de referência de patches personalizada que você cria, é possível especific ar opcionalmente quantos dias aguardar após o lançamento ou atualização para aprovar automaticamente a instalaçã o. Para o Amazon Linux 1 e o Amazon Linux 2, vários fatores influenciam a forma de cálculo das últimas datas de lançamento e de atualizaç ão. A fim de ajudar você a evitar resultados inesperad os ao escolher esperas para a aprovação automática, esses fatores são explicados no tópico Como as datas de lançamento e atualização de pacote são calculadas.	24 de agosto de 2022

Conteúdo atualizado: aplique patches a uma AMI e atualize um grupo do Auto Scaling	Atualizamos o passo a passo Atualização do AMIs para grupos do Auto Scaling para usar modelos de execução em vez de configurações de execução. Além disso, implementamos as ações mais recentes e os tempos de execução do Automation mais recentes no conteúdo dos runbooks.	22 de junho de 2022
Change Manager: impeça que os usuários criem solicitações aprováveis automaticamente	É possível configurar modelos de alteração no Change Manager para oferecer suporte a aprovações automáticas, o que significa que os usuários com as permissões necessárias do IAM podem optar por iniciar a solicitação de alteração sem exigir aprovação adicional . Agora também é possível impedir usuários individuais, grupos ou perfis do IAM de enviar solicitações de aprovação automática, mesmo que um modelo de alteração ofereça suporte a eles. Isso é obtido por meio do uso de uma nova chave de condição do IAM, `ssm:AutoApprove` . Para obter mais informações, consulte Controlar o acesso a fluxos de trabalho do runbook de aprovação automática	15 de junho de 2022

Atualização da orientação para perfis de tarefas da janela de manutenção	Anteriormente, o console do Systems Manager permitia a você escolher o perfil `AWSServiceRoleForAmazonSSM` vinculado ao serviço do IAM gerenciado pela AWS para usar como perfil de manutenção para suas tarefas. O uso desse perfil e sua política associada, `AmazonSSMServiceRolePolicy`, para tarefas de janela de manutenção não é mais recomendado. Em vez disso, crie uma política personalizada e um perfil para tarefas de janela de manutenção. Para obter mais informações, consulte Configurar o Maintenance Windows.	9 de junho de 2022
Suporte ao encaminhamento de portas para hosts remotos para Session Manager	O Session Manager agora oferece suporte a sessões de encaminhamento de portas para hosts remotos. O host remoto não precisa ser gerenciado pelo Systems Manager. Para obter mais informações, consulte Iniciar uma sessão (encaminhamento de portas para host remoto).	25 de maio de 2022

Conteúdo atualizado: instruções para instalar o SSM Agent manualmente em instâncias do Amazon EC2 Linux	Em resposta ao feedback dos clientes, revisamos os tópicos que fornecem instruções para instalar o SSM Agent manualmente em instâncias do Amazon EC2. Esses tópicos agora fornecem comandos usando arquivos disponíveis globalmente que você pode copiar e colar para instalação rápida em instâncias do EC2 em qualquer Região da AWS. Esses tópicos também fornecem informações para ajudar você a criar comandos de instalação que usam arquivos disponíveis em sua própria região de trabalho. A última abordagem é recomendada quando você está instalando o agente em várias instâncias usando um script ou modelo. Para obter mais informações, consulte as instruções para seu sistema operacional Linux na seção Instalar o SSM Agent manualmente em instâncias do EC2 para Linux.	9 de maio de 2022

Novo tópico: Amazon Machine Images (AMIs) com SSM Agent pré-instalado	Em resposta ao feedback do cliente, centralizamos informações sobre quais AMIs gerenciadas pela AWS incluem o SSM Agent pré-instalado. Este tópico também fornece instruções sobre como verificar se uma instância do Amazon EC2 criada dessas AMIs foi instalada com sucesso e está em execução. Para casos raros em que o agente pode não ser instalado com êxito ou ser instalado , mas não iniciar, também fornecemos informações sobre como iniciar ou instalar o agente manualmente nessas instâncias. Para obter mais detalhes, consulte Amazon Machine Images (AMIs) com SSM Agent pré-instalado.	8 de maio de 2022
Nova seção State Manager	Adicionada uma nova seção que descreve os detalhes de quando State Manager executa associações. Para obter mais informações, consulte Sobre a programação de associações.	27 de abril de 2022

O Patch Manager agora oferece suporte para Rocky Linux	Agora é possível usar Patch Manager para aplicar o patch nos nós do Rocky Linux. Muitas das regras que se aplicam ao RHEL 8 para patches também se aplicam ao Rocky Linux. O Rocky Linux 8 usa o novo `AWS-DefaultRockyLinuxPatchBaseline` . Para obter mais informações, consulte os tópicos a seguir. • Como os patches de segurança são selecionados • Como os patches são instalados • Como as regras de lista de referência de patches funcionam no RHEL, CentOS Stream e Rocky Linux.	14 de abril de 2022

O Patch Manager agora oferece suporte ao CentOS Stream 8	Agora é possível usar o Patch Manager para aplicar patches às instâncias do CentOS Stream 8 e Red Hat Enterprise Linux (RHEL) 4.4-4.5. Muitas das regras que se aplicam ao RHEL 8 para patches também se aplicam ao CentOS Stream 8. O CentOS Stream 8 usa o `AWS-DefaultCentOSPatchBaseline` . Para obter mais informações, consulte os tópicos a seguir. • Como os patches de segurança são selecionados • Como os patches são instalados • Como as regras de lista de referência de patches patch funcionam no RHEL e no CentOS Stream	4 de abril de 2022

Criar uma função assumida para Change Manager	Uma nova seção esclarece os requisitos para criar e implementar uma função assumida para Change Manager. Uma função assumida é uma função de serviço (IAM) AWS Identity and Access Management que permite Change Manager executar com segurança os fluxos de trabalho de runbook especificados em uma solicitação de alteração aprovada em seu nome. A função concede AWS Systems Manager (AWS STS) `AssumeRole` confiança para Change Manager. Para obter informações, consulte Configurando funções e permissões para Change Manager.	18 de março de 2022
Aprovar ou rejeitar Change Manager altera as solicitações em massa	No console do Systems Manager, agora você pode selecionar várias solicitaç ões de alteração para aprovar ou rejeitar em uma única operação. Para obter informações, consulte Revisão e aprovação ou rejeição de solicitações de alteração (console).	8 de março de 2022

Suporte para Rocky Linux e Windows Server nós gerenciados de 2022	O Systems Manager oferece suporte a nós gerenciad os Rocky Linux e Windows Server de 2022, incluindo dispositivos de borda e máquinas híbridas on-premis es ou em outros provedore s de nuvem. Para usar o Systems Manager com esses sistemas operacionais, você deve concluir todos os procedimentos necessários de configuração do Systems Manager, incluindo procedime ntos para ambientes híbridos ou dispositivos de borda, se aplicável. Para obter mais informações, consulte Setting up Systems Manager (Configurar o gerenciador de sistemas). Para máquinas Rocky Linux, o SSM Agent também deve ser instalado manualmente. Para obter mais informações, consulte Instalar manualmente o SSM Agent nas instâncias Rocky Linux. Para as instância s do do Amazon Elastic Compute Cloud (Amazon EC2) Windows Server de 2022, SSM Agent é instalado por padrão.	1º de março de 2022

Permitir que o Automation se adapte às suas necessidades de simultaneidade e exiba métricas de uso do Automation	Agora você pode permitir que o Automation ajuste automaticamente sua cota de automação simultânea e visualize as métricas de uso do Automation publicadas no CloudWatch. Para obter mais informações sobre a simultaneidade adaptativa, consulte Permitir que o Automation se adapte às suas necessidades de simultaneidade. Para obter mais informações sobre como exibir as métricas de uso do Automation, consulte Monitoramento das métricas do Automation usando o Amazon CloudWatch.	27 de janeiro de 2022

Permitir que o Automation se adapte às suas necessidades de simultaneidade e exiba métricas de uso do Automation	Agora você pode permitir que o Automation ajuste automaticamente sua cota de automação simultânea e visualize as métricas de uso do Automation publicadas no CloudWatch. Para obter mais informações sobre a simultaneidade adaptativa, consulte Permitir que o Automation se adapte às suas necessidades de simultaneidade. Para obter mais informações sobre como exibir as métricas de uso do Automation, consulte Monitoramento das métricas do Automation usando o Amazon CloudWatch.	27 de janeiro de 2022
Documentos do Systems Manager organizados por categorias	Os documentos do Systems Manager de propriedade da Amazon agora são organizados por tipo e categorias para ajudar você a encontrar os documentos de que precisa.	13 de janeiro de 2022

Crie e invoque integrações para o Automation	Agora você pode enviar mensagens usando webhooks durante uma automação com a criação de uma integração. As integrações podem ser invocadas durante uma automação usando a nova ação `aws:invokeWebhook` em seu runbook. Para obter mais informações sobre a criação de integrações, consulte Criação de integrações de webhooks para o Automation. Para saber mais sobre a ação `aws:invokeWebhook` , consulte aws:invokeWebhook — Invoque uma integração de webhook do Automation.	13 de janeiro de 2022
Recursos não disponíveis na nova Região da AWS	Os recursos do Systems Manager a seguir não estão disponíveis no momento na nova região Ásia-Pacífico (Jacarta). • Application Manager • Change Calendar • Change Manager • Explorer • Fleet Manager • Incident Manager • Quick Setup	13 de dezembro de 2021

Exibir detalhes do custo do recurso de uma aplicação	O Application Manager é integrado ao Gerenciamento de Faturamento e Custos da AWS por meio do widget Cost Explorer. Depois de ativar o Cost Explorer no console Billing and Cost Management (Gerenciamento de custos e cobranças), o widget Cost Explorer no Application Manager mostra os dados de custos para uma aplicação fora de um contêiner específico ou um componente da aplicação . Você pode usar filtros no widget para exibir dados de custo de acordo com diferente s períodos de tempo, granulari dades e tipos de custo em um gráfico de barras ou linhas. Para obter mais informações, consulte Visualizar informaçõ es da visão geral sobre uma aplicação.	7 de dezembro de 2021
Gerencie processos usando o Fleet Manager	Agora é possível usar o Fleet Manager para gerenciar processos em seus nós. Para obter mais informaçõ es, consulte Trabalhar com processos.	6 de dezembro de 2021

Alteração de terminologia: instâncias gerenciadas agora são chamadas de nós gerenciados	Com suporte para dispositivos principais do AWS IoT Greengrass, a frase instância gerenciada foi alterada para nó gerenciado na maior parte da documentação do Systems Manager. O console do Systems Manager, as chamadas de API, as mensagens de erro e os documentos SSM ainda usam o termo instância.	29 de novembro de 2021

Suporte para dispositivos de borda	O Systems Manager oferece suporte às seguintes configurações de dispositivos de borda. • AWS IoT Greengrass: O Systems Manager agora oferece suporte a qualquer dispositivo configurado para AWS IoT Greengrass e executa o software principal do AWS IoT Greengrass. Para integrar seus dispositivos principais do AWS IoT Greengrass, crie uma função de serviço do AWS Identity and Access Management (IAM). Use também o console do AWS IoT Greengrass para implantar o SSM Agent como um componente do AWS IoT Greengrass em seus dispositivos. Para obter mais informações, consulte Setting up AWS Systems Manager for edge devices (Configurar o gerenciador de sistemas) para dispositivos de borda). • Dispositivos de borda em um ambiente híbrido: o Systems Manager também oferece suporte a dispositivos principais do AWS IoT e dispositivos IoT não AWS, depois de configurá	29 de novembro de 2021

	-los como máquinas on-premises. Para integrar seus dispositivos, você deve criar uma função de serviço do IAM, criar uma ativação do nó gerenciado para um ambiente híbrido e instalar manualmente o SSM Agent em seus dispositivos. Para obter mais informações, consulte Configurar o AWS Systems Manager para ambientes híbridos.	
Conecte-se às instâncias gerenciadas usando o Desktop Remoto	Agora é possível usar o Fleet Manager para conectar-se a instâncias do Windows gerenciadas usando o protocolo do Remote Desktop (RDP). Essas sessões do Remote Desktop com tecnologia Amazon DCV fornecem conexões seguras para suas instâncias diretamente do navegador. Para obter mais informações, consulte Conectar usando o Desktop Remoto.	23 de novembro de 2021

Especifique a duração máxima da sessão e forneça motivos para as sessões	Agora você pode especificar uma duração máxima da sessão para todas as sessões do Session Manager em uma Região da AWS na sua Conta da AWS. Quando uma sessão atinge a duração especificada, ela é encerrada. Agora, você também pode adicionar, opcionalmente, um motivo ao iniciar uma sessão. Para obter mais informações, consulte Specify maximum session duration (Especificar a duração máxima da sessão).	16 de novembro de 2021

O Patch Manager agora oferece suporte ao sistema operacional Raspberry Pi OS	Agora é possível usar o Patch Manager para aplicar patches em instâncias do Raspberry Pi OS. O Patch Manager oferece suporte a aplicação de patches no Raspberry Pi OS 9 (Stretch) e 10 (Buster). Como o sistema operacional Raspberry Pi OS é baseado no Debian, muitas das mesmas regras de patch se aplicam a ele como ao Debian Server. Para obter mais informações, consulte os tópicos a seguir. • Como os patches de segurança são selecionados • Como os patches são instalados • Como as regras de linha de base de patch funcionam no Debian Server e no Raspberry Pi OS	16 de novembro de 2021
Acesse o Red Hat Knowledge base Portal	Use o Fleet Manager para acessar o portal RHEL Knowledgebase para encontrar soluções, artigos, documentação e vídeos sobre o uso de produtos Red Hat. Para obter mais informações, consulte Accessing the Red Hat Knowledgebase portal.	3 de novembro de 2021

Edição em massa do OpsItems	O OpsCenter agora é compatível com o OpsItems para edição em massa. Você pode selecionar vários OpsItems e editar um dos seguintes campos: Status, Priority (Prioridade), Severity (Gravidade), Category (Categoria). Para obter mais informações, consulte Editar o OpsItems.	15 de outubro de 2021
Crie parâmetros de entrada que preenchem os recursos da AWS	Agora, você pode criar parâmetros de entrada em runbooks de automação que preenchem recursos da AWS no AWS Managemen t Console. Para obter mais informações, consulte Creating input parameters that populate AWS resources (Criar parâmetros de entrada que preencham os recursos da).	14 de outubro de 2021
Nova opção de corte de invocação de tarefas para janelas de manutenção	Agora você pode optar por bloquear qualquer nova chamada de tarefa desde o início, depois que o tempo limite especificado para uma janela de manutenção for atingido. Para obter informaçõ es, consulte Assign tasks to a maintenance window (console) (Atribuir tarefas a uma janela de manutenção - console).	13 de outubro de 2021

Suporte do Patch Manager para macOS 11.3.1 e 11.4 (Big Sur)	As instâncias do Amazon Elastic Compute Cloud (Amazon EC2) para macOS 11.3.1 e 11.4 (Big Sur) agora podem ser corrigidas usando o Patch Manager. Isso é além do suporte existente para macOS 10.14.x (Mojave) e 10.15.x (Catalina). Para obter informações sobre o trabalho com o Patch Manager, consulte AWS Systems ManagerPatch Manager.	1.º de outubro de 2021

Insights de aplicações no Application Manager	O Application Manager integra-se ao Amazon CloudWatch Application Insights. O Application Insights identifica e configura as principais métricas, logs e alarmes na pilha de tecnologia e nos recursos da aplicação. O Application Insights monitora continuam ente as métricas e os logs para detectar e correlaci onar anomalias e erros. Quando erros e anomalias são detectados, o Applicati on Insights gera CloudWatc h Events que você pode usar para configurar notificações ou executar ações. Você pode habilitar e exibir o Applicati on Insights em Overview (Visão geral) e nas guias Monitoring (Monitoramento) no Application Manager. Para obter mais informações sobre o Application Insights, consulte O que é o Amazon CloudWatc h Application Insights no Manual do usuário do Amazon CloudWatch.	21 de setembro de 2021

Importar eventos de outros calendários para Change Calendar	Agora você pode importar os eventos de um calendário de terceiros para um calendário no Change Calendar. Anteriormente, cada evento tinha que ser inserido manualmente em um calendário. Depois de exportar um calendário de um fornecedor de calendário de terceiros compatível para um arquivo do iCalendar (`.ics`), importe-o para Change Calendar e seus eventos serão incluídos nas regras para o calendário aberto ou fechado no Systems Manager. Os provedores compatíveis incluem o Calendário do iCloud, o Calendário do Google e o Microsoft Outlook. Para obter mais informações, consulte Importar e gerenciar eventos de calendários de terceiros.	8 de setembro de 2021

Novos recursos de marcação e runbook no Application Manager	Os aprimoramentos de marcação incluem a capacidade de adicionar ou excluir tags de um recurso específico ou de todos os recursos em uma aplicação do Application Manager. Os aprimoramentos de runbooks incluem a capacidade de exibir uma lista filtrada de runbooks referente a um tipo de recurso específico ou iniciar um runbook em todos os recursos do mesmo tipo. Para obter mais informaçõ es, consulte Trabalhar com tags no Application Manager e Trabalhar com runbooks no Application Manager.	31 de agosto de 2021
Novo exemplo: Crie uma solicitação de alteração usando a AWS CLI	Um exemplo de criação de uma solicitação de alteração com a AWS CLI foi adicionado ao capítulo Change Manager. O exemplo usa o exemplo de modelo de alteração `AWS-HelloWorldChan geTemplate` e o `AWS-HelloWorld runbook` : • Criar solicitações de alteração (AWS CLI)	20 de agosto de 2021

Nova seção: Usar parâmetros no Amazon EKS	Uma nova seção foi adicionada ao capítulo Parameter Store. Este tópico é uma demonstração sobre como usar seus parâmetros em clusters do Amazon EKS. Para obter mais informações, consulte Usar parâmetros do Parameter Store no Amazon Elastic Kubernetes Service.	19 de agosto de 2021
Hooks do ciclo de vida do Patch Manager atualizados	O Patch Manager agora fornece um hook do ciclo de vida, a capacidade de executar um documento do Systems Manager Command, para um ponto adicional durante uma operação de aplicação de patches, Patch now (Aplicar patches agora). Caso programe reinicializações de instância após executar o comando Aplicar patch agora, você poderá especificar um hook do ciclo de vida para ser executado após a conclusão da reinicialização. Para obter mais informações, consulte Usar os hooks do ciclo de vida “Patch now” e Sobre o documento do SSM `AWS-RunPatchBaselineWithHooks`.	9 de agosto de 2021

Aprovações automáticas agora suportadas para as solicitações do Change Manager	Agora, você pode configurar modelos de alteração no Change Manager para oferecer suporte a aprovações automáticas, o que significa que os usuários com as permissões necessárias do IAM podem optar por iniciar a solicitação de alteração sem exigir aprovação adicional. Os usuários que têm acesso a modelos de aprovação automática ainda podem optar por especificar aprovadores se quiserem. Para ajudar você a controlar seus processos do Change Manager, as aprovações ainda são necessárias para todas as solicitações durante os períodos de congelamento de alterações. Para obter mais informações, consulte os tópicos a seguir. • Criar modelos de alteração • Criar solicitações de alteração • Experimente o modelo de alteração `Hello World` gerenciado pela AWS	30 de julho de 2021

Insights operacionais do OpsCenter	O OpsCenter analisa automaticamente o OpsItems em sua conta e gera insights. Um insight inclui informações para ajudar você a entender quantas duplicatas OpsItems estão na sua conta e quais fontes estão criando essas duplicatas. Os insights também fornecem práticas recomendadas e runbooks do Automation para ajudar a resolver OpsItems duplicados. Para obter mais informações, consulte Trabalhar com insights operacionais.	13 de julho de 2021
Visualizar instâncias interrompidas no Fleet Manager	Agora é possível visualizar quais instâncias estão `running` e quais estão `stopped` no console do Fleet Manager. Para obter mais informações, consulte AWS Systems ManagerFleet Manager.	12 de julho de 2021
Novo tópico: Runbooks de Automação de Criação	Um novo tópico sobre a Criação de runbooks de automação fornece orientações e exemplos narrativos de como criar conteúdo para runbooks de Automação personalizados.	8 de julho de 2021

Criação de pilha e modelo do AWS CloudFormation no Application Manager	O Application Manager ajuda você a provisionar e gerenciar recursos para suas aplicações integrando o CloudFormation. Você pode criar, editar e excluir modelos e pilhas do AWS CloudFormation no Application Manager. O Application Manager também inclui uma biblioteca de modelos na qual você pode clonar, criar e armazenar modelos. O Application Manager e o CloudFormation exibem as mesmas informações sobre o status atual de uma pilha. Modelos e atualizações de modelos são armazenados no Systems Manager até que você provisione a pilha, momento em que as alterações também são exibidas no CloudFormation. Para obter mais informações, consulte Como trabalhar com pilhas do AWS CloudFormation no Application Manager.	8 de julho de 2021

Novo tópico: Alternar automaticamente chaves privadas para o SSM Agent em instâncias híbridas	Um novo tópico sobre a Configuração da rotação automática da chave privada fornece instruções sobre como fortalecer sua postura de segurança, configurando o SSM Agent para girar automaticamente a chave privada do ambiente híbrido.	15 de junho de 2021
O plugin Session Manager para a AWS CLI versão 1.2.205.0	Uma nova versão do plugin do Session Manager para a AWS CLI foi liberada. Para obter mais informações, consulte Versão mais recente do plugin do Session Manager e histórico de versões.	10 de junho de 2021
Nova função vinculada a serviços do IAM	Ao habilitar os insights operacionais do OpsCenter, o Systems Manager cria uma nova função vinculada a serviços do AWS Identity and Access Management (IAM) chamada `AWSSSMOpsInsightsServiceRolePolicy`. Para obter mais informações sobre essa função, consulte Usar funções para criar OpsItems de insights operacionais no Systems Manager OpsCenter: `AWSSSMOpsInsightsServiceRolePolicy`.	9 de junho de 2021

Novo conteúdo de solução de problemas do Patch Manager para Linux	Um novo tópico sobre Erros na execução do `AWS-RunPatchBaseline` on Linux fornece descrições e soluções para vários problemas que podem ocorrer ao aplicar patches em instâncias gerenciadas com sistemas operacionais Linux.	8 de junho de 2021

Suporte aprimorado para tarefas de janela de manutenção que não exigem destinos especificados (console)	Agora, você pode criar tarefas da janela de manutenção no console sem precisar especificar um destino na tarefa se não for necessário. Anteriormente, essa opção estava disponível somente ao usar a AWS CLI ou a API. Esta opção se aplica à automação, AWS Lambda, e tipos de tarefa do AWS Step Functions. Por exemplo, se você criar uma tarefa de automação e os recursos a serem atualizados forem especificados nos parâmetros do documento de automação, não será mais necessário especificar um destino na própria tarefa. Para obter mais informações, consulte Registro de tarefas da janela de manutenção sem destinos, Atribuição de tarefas a uma janela de manutenção (console) e Programação de automações com janelas de manutenção.	28 de maio de 2021
Referência de runbooks de automação realocada	A referência de runbooks de automação foi transferida para um novo local. Para obter mais informações, consulte Referência de runbooks do Systems Manager Automation.	10 de maio de 2021

Inicialização do AWS Systems Manager Incident Manager	O Incident Manager é um console de gerenciamento de incidentes criado para ajudar os usuários na mitigação e recuperação de incidente s que afetam os aplicativ os hospedados na AWS. Para obter mais informações, consulte o Guia do usuário do AWS Systems Manager Incident Manager.	10 de maio de 2021
O State Manager oferece suporte ao Change Calendar	Agora é possível especificar nomes do recurso da Amazon (ARNs) do Change Calendar ao criar ou atualizar uma associação do State Manager. O State Manager aplica associações somente quando o calendário de alterações estiver aberto, e não quando estiver fechado. Para obter mais informações, consulte Criar associações e Editar e criar uma versão de uma associação.	6 de maio de 2021
Clonar documentos do Systems Manager	Usando o console dos Documentos do Systems Manager, agora você pode copiar o conteúdo de um documento existente para um novo documento que pode ser modificado. Para saber mais, consulte Clonar um documento do SSM.	4 de maio de 2021

Integrar o Security Hub ao Explorer e OpsCenter	Agora, você pode integrar o Explorer e OpsCenter ao AWS Security Hub. O Security Hub fornece uma visão abrangente do estado de segurança na AWS e ajuda a verificar o ambiente de acordo com os padrões e as práticas recomendadas do setor de segurança. Quando integrado ao Explorer, você pode visualizar as descobert as de segurança no widget Security Hub no painel do Explorer. Quando integrado ao OpsCenter, você pode criar OpsItems para as descobert as do Security Hub. Para obter mais informações, consulte Receber descobertas do AWS Security Hub no Explorer e Receber descobertas do AWS Security Hub no OpsCenter.	27 de abril de 2021
Novo tópico: Convenções de documentos	Adicionamos um novo tópico para ajudar os usuários a entender as convençõe s tipográficas comuns do Manual do usuário do AWS Systems Manager. Para obter mais informações, consulte Convenções do documento.	21 de abril de 2021

Tópico atualizado: sobre a aplicação de patches em aplicações lançadas pela Microsoft no Windows Server	O tópico About patching applications released by Microsoft on (Sobre a aplicação de patches em aplicações lançadas pela Microsoft no Windows Server) agora esclarece que, para que o Patch Manager possa aplicar patches em aplicações lançadas pela Microsoft nas instâncias gerenciadas pelo Windows Server, a opção de atualização do Windows Give me updates for other Microsoft products when I update Windows (Fornecer atualizações para outros produtos Microsoft quando eu atualizar o Windows) deverá ser permitida na instância.	12 de abril de 2021
Reorganização da referência de runbooks de automação	Para ajudar você a encontrar os runbooks de que precisa e analisar a referência de forma mais eficiente, reorganizamos o conteúdo da referência de runbooks de automação pelo AWS service (Serviço da AWS) relevante. Para visualizar essas alterações, consulte Referência de runbooks do Systems Manager Automation.	12 de abril de 2021

Patch Manager: gerar relatórios .csv de conformidade de patches	O Patch Manager passou a oferecer suporte à capacidade de gerar relatórios de conformidade de patchs para as instâncias e salvar o relatório em um bucket do S3 de sua escolha, no formato .csv. Em seguida, usando uma ferramenta como Amazon QuickSight, é possível analisar os dados do relatório de conformidade de patches. Você pode gerar um relatório de conformidade de patch para uma única instância ou para todas as instâncias na Conta da AWS. Você pode gerar um relatório único sob demanda ou configurar uma programação para que os relatórios sejam criados automaticamente. Você também pode especificar um tópico do Amazon Simple Notification Service para fornecer notificações quando um relatório for gerado. Para obter mais informações, consulte Gerar relatórios de conformidade de patches CSV.	9 de abril de 2021

Exclua os rótulos do parâmetro do Parameter Store	Agora é possível excluir os rótulos do parâmetro do Parameter Store usando o console do Systems Manager ou a AWS CLI. Para obter mais informações, consulte Trabalhar com rótulos de parâmetros.	6 de abril de 2021
Agendar reinicializações de instância ao usar Patch Now	O Patch Manager agora é compatível com o agendamento de um horário para que suas instâncias sejam reinicializadas após a instalação de patches usando o recurso Patch Now (Aplicar patches agora). Isso se soma às opções existentes para reinicializar instâncias somente se necessário a fim de concluir uma instalação de patch ou ignorar todas as reinicializações após a operação de patch. Para obter mais informações, consulte Aplicação de patches de instâncias sob demanda.	1.º de abril de 2021

Novo tópico: Descubra parâmetros públicos	Os parâmetros públicos do Parameter Store agora podem ser encontrados usando a AWS CLI ou o console do Systems Manager. Para obter mais informações, consulte Finding public parameters (Descobrir parâmetros públicos).	1.º de abril de 2021
O patch agora é atualizado: armazene logs no S3 e execute hooks de ciclo de vida	Ao executar a operação Patch Now do Patch Manager, você pode escolher um bucket do S3 no qual armazenar automaticamente os logs de aplicação de patches. Além disso, você pode optar por executar documentos de comando do Systems Manager (documentos SSM) como ganchos de ciclo de vida em três pontos durante a operação: Antes da instalação, Após a instalação, e Na saída. Para obter mais informações, consulte Aplicar patch em instâncias sob demanda.	31 de março de 2021

O Systems Manager agora relata alterações em suas políticas gerenciadas pela AWS	Desde 24 de março de 2021, as alterações nas políticas gerenciadas são relatadas no tópico Atualizações do Systems Manager nas políticas gerenciadas pela AWS. A primeira alteração listada é a adição de suporte para a ferramenta Explorer gerar relatórios do OpsData e OpsItems de várias contas e regiões.	24 de março de 2021

O Explorer permite automaticamente todas as fontes do OpsData para sincronizações de dados de recursos, com base nas contas do AWS Organizations	Ao criar uma sincronização de dados do recurso, se você escolher uma das opções do AWS Organizations, o Systems Manager permitirá automaticamente todas as origens do OpsData em todas as Regiões da AWS e Contas da AWS selecionadas para a sua organização (ou nas unidades organizacionais selecionadas). Isso significa, por exemplo, que mesmo que você não tenha permitido o Explorer em uma Região da AWS, se você selecionar uma opção do AWS Organizations para a sincronização de dados de seus recursos, o Systems Manager coletará automaticamente o OpsData dessa região. Para obter mais informações, consulte Sobre sincronizações de dados de recursos em várias contas e regiões.	24 de março de 2021

O Systems Manager Automation fornece uma nova variável de sistema para seus runbooks	Com a nova variável de sistema `global:AWS_PARTITION`, você pode especificar a partição da AWS em que um recurso está localizado ao criar seus runbooks. Para obter mais informações, consulte Variáveis do sistema de automação.	18 de março de 2021
Permitir vários níveis de aprovação para as solicitações de alteração do Change Manager	Ao criar um modelo de alteração do Change Manager, agora você pode exigir que mais de um nível de aprovadores conceda permissão para que uma solicitação de alteração seja executada. Por exemplo, você pode exigir que os revisores técnicos aprovem primeiro uma solicitação de alteração criada a partir de um modelo de alteração e, em seguida, solicitar um segundo nível de aprovações de um ou mais gerentes. Para obter mais informações consulte Criar modelos de alteração.	4 de março de 2021

O Patch Manager agora oferece suporte ao Oracle Linux 8.x	Agora é possível usar o Patch Manager para aplicar patches às instâncias do Oracle Linux 8.x, por meio da versão 8.3. Para obter mais informações, consulte os tópicos a seguir. • Como os patches de segurança são selecionados • Como os patches são instalados • Como as regras de lista de referência de patches funcionam no Oracle Linux	1.º de março de 2021
O OpsCenter exibe outros OpsItems para um recurso selecionado	Para ajudar você a investigar problemas e fornecer contexto para um problema, você pode exibir uma lista de OpsItems para um recurso específic o do AWS. A lista exibe o status, gravidade e título de cada OpsItem. A lista também inclui links profundos para cada OpsItem. Para obter mais informações, consulte Visualizar outros OpsItems para um recurso específico.	1.º de março de 2021
Defina as preferências para a aplicação de patches no runtime	Agora você pode definir preferências de patche no runtime usando o recurso de substituição da lista de referência. Para obter mais informações, consulte Usar o parâmetro BaselineOverride.	25 de fevereiro de 2021

Novo tipo de documento do Systems Manager	Os modelos do AWS CloudFormation agora podem ser armazenados como documentos do Systems Manager. O armazenamento de modelos do CloudFormation como documentos do Systems Manager permite que você se beneficie de recursos de documentos do Systems Manager, como controle de versão, comparação de conteúdo da versão e compartilhamento com contas. Para mais informações, consulte Documentos do AWS Systems Manager.	9 de fevereiro de 2021

Instâncias de patch usando hooks opcionais	Novo documento do MUSA`AWS-RunPatchBaseli neWithHooks` fornece ganchos que você pode usar para executar documentos SSM em três pontos durante o ciclo de aplicação de patches da instância. Para obter informações sobre o `AWS-RunPatchBaseli neWithHooks` , consulte Sobre o documento SSM do `AWS-RunPatchBaseli neWithHooks` . Para obter um exemplo demonstrativo de uma operação de patches que usa todos os três hooks, consulte Demonstração: atualizar dependências de aplicações, corrigir uma instância e executar uma verificação de integridade específica para a aplicação.	2 de fevereiro de 2021

Novo tópico: Validar servidores on-premises e máquinas virtuais usando uma impressão digital do hardware	O SSM Agent verifica a identidade de servidores on-premises e máquinas virtuais (VMs) que você registra no serviço, usando uma impressão digital computada . A impressão digital é uma string opaca, armazenada noCofreque o agente passa para determinadas APIs do Systems Manager. Para obter informações sobre a impressão digital de hardware e instruções para configura r um limite de similaridade para auxiliar na verificação da máquina, consulte Validar servidores on-premises e máquinas virtuais usando uma impressão digital de hardware.	25 de janeiro de 2021
Novo tópico: Referência técnica do SSM Agent	O tópico Referência técnica do SSM Agent reúne informaçõ es para ajudar a implement ar o AWS Systems Manager SSM Agent e entender como o agente funciona. Este tópico inclui uma seção totalmente nova,SSM AgentAtualizações contínuasRegiões da AWS.	21 de janeiro de 2021

SSM Agent no Windows Server 2008	A partir de 14 de janeiro de 2020, o Windows Server 2008 não será mais compatível com recursos ou atualizações de segurança da Microsoft. As AMIs do Windows Server 2008 já incluem o SSM Agent, mas o agente não será mais atualizado para esse sistema operacional.	5 de janeiro de 2021
Suporte aprimorado para tarefas de janela de manutenção que não exigem destinos especificados (AWS CLI e somente API)	Agora você pode criar tarefas de janela de manutenção sem precisar especificar um destino na tarefa se não for necessário (somente AWS CLI e API). Isso se aplica ao Automation, tipos de tarefa AWS Lambda e AWS Step Functions. Por exemplo, se você criar uma tarefa de automação e os recursos a serem atualizados forem especificados nos parâmetros do runbook de automação, não será mais necessário especificar um destino na própria tarefa. Para obter mais informações, consulte Registro de tarefas da janela de manutenção sem destinos e Programação de automações com janelas de manutenção.	23 de dezembro de 2020

Novos recursos de automação	Uma nova propriedade compartilhada foi adicionad a aos runbooks do Systems Manager Automation. O`onCancel`permite que você especifique para qual etapa a automação deve ir no caso de um usuário cancelar a automação. Para obter mais informações, consulte Properties shared by all actions (Propriedades compartilhadas por todas as ações).	21 de dezembro de 2020
Novo tópico: Trabalhar com associações usando o IAM	Um novo tópico foi adicionad o ao capítulo State Manager do Systems Manager, que descreve as práticas recomendadas para criar associações usando o IAM. Para obter mais informaçõ es, consulte Trabalhar com associações usando o IAM.	18 de dezembro de 2020
O State Manager agora é compatível com várias regiões e contas múltiplas	As associações agora podem ser criadas ou atualizadas com várias regiões ou contas. Para obter mais informações, consulte Criar associações.	15 de dezembro de 2020

Nova ferramenta: Fleet Manager	O Fleet Manager, uma ferramenta do AWS Systems Manager, é uma experiênc ia de interface de usuário unificada (UI) que ajuda você a gerenciar remotamen te sua frota de servidores em execução na AWS ou on-premises. com o Fleet Manager, é possível visualiza r o status de integridade e a performance de toda a sua frota de servidores em um console. Você também pode coletar dados de instância s individuais para executar tarefas comuns de solução de problemas e gerenciam ento no console. Para obter informações, consulte, AWS Systems Manager Fleet Manager.	15 de dezembro de 2020

Nova ferramenta: Change Manager	A Amazon Web Services lançou o Change Manager, um framework de gerenciamento de alterações corporativas para solicitar, aprovar, implementar e emitir relatórios sobre alterações operacionais na configuração e infraestrutura das aplicações. Em uma única Conta de administrador delegado, se você usar o AWS Organizations, poderá gerenciar alterações em várias Contas da AWS e Regiões da AWS. Como alternativa, usando um conta local, você pode gerenciar alterações para uma única Conta da AWS. Use o Change Manager para gerenciar alterações em recursos da AWS e recursos on-premises. Para obter informações, consulte, AWS Systems Manager Change Manager.	15 de dezembro de 2020

Nova ferramenta: Application Manager	O Application Manager ajuda você a investigar e corrigir problemas com os recursos da AWS no contexto de suas aplicações. O Application Manager agrega informaçõ es de operações de vários Serviços da AWS e de ferramentas do Systems Manager em um único AWS Management Console. Para obter informações, consulte, AWS Systems Manager Application Manager.	15 de dezembro de 2020

O AWS Systems Manager é compatível com as instâncias do Amazon EC2 para macOS	Em conjunto com o lançamento do suporte do Amazon Elastic Compute Cloud (Amazon EC2) para instâncias do macOS, o Systems Manager agora oferece suporte a muitas operações em instâncias do EC2 para macOS. As versões compatíveis incluem o macOS 10.14.x (Mojave) e 10.15.x (Catalina). Para obter mais informações, consulte os tópicos a seguir. • Para obter informações sobre a instalação do SSM Agent em instâncias do EC2 para macOS, consulte Instalar e configurar o SSM Agent em instâncias do EC2 para macOS.. • Para obter informações sobre patches de instâncias do EC2 para macOS, consulte Como os patches são instalados e Criação de uma lista personalizada de referência de patches (macOS). • Para obter mais informações sobre o suporte a instâncias do EC2 para macOS, consulte Instâncias Mac do Amazon EC2 no Guia do usuário do Amazon EC2.	30 de novembro de 2020

Pseudo parâmetros da janela de manutenção: Novo tipo de recurso suportado para `{{TARGET_ID}}` e `{{RESOURCE_ID}}`	Um tipo de recurso adicional agora está disponível para uso com os pseudo parâmetros `{{TARGET_ID}}` e `{{RESOURCE_ID}}`. Agora é possível usar o tipo de recurso `AWS::RDS::DBCluster` com os dois pseudoparâmetros. Para obter mais informações sobre os pseudoparâmetros da janela de manutenção, consulte Usar pseudoparâmetros ao registrar tarefas da janela de manutenção.	27 de novembro de 2020
O plugin Session Manager para a AWS CLI versão 1.2.30.0	Uma nova versão do plugin do Session Manager para a AWS CLI foi liberada. Para obter mais informações, consulte Versão mais recente do plugin do Session Manager e histórico de versões.	24 de novembro de 2020
Novo tópico: Comparando versões de documentos do SSM	Agora você pode comparar as diferenças de conteúdo entre versões de documentos SSM no console Documents (Documentos) do Systems Manager. Para obter mais informações, consulte Comparar versões de documentos do SSM.	24 de novembro de 2020

O Systems Manager agora oferece suporte às políticas de VPC endpoint	Agora você pode criar políticas para endpoints da interface da VPC para o Systems Manager. Para obter mais informações, consulte Criar uma política de endpoint da VPC na interface.	18 de novembro de 2020
Novo tópico: Especificar um valor de tempo limite de sessão ociosa	Agora você pode especific ar a quantidade de tempo para permitir que um usuário fique inativo antes que uma sessão termine com o Session Manager. Para obter mais informações, consulte Especificar um valor de tempo limite para a sessão ociosa.	18 de novembro de 2020
Novo recurso de log do Session Manager	Agora você pode enviar um stream contínuo de logs de dados de sessão formatado s em JSON para o Amazon CloudWatch Logs. Para obter mais informações, consulte Transmitir dados da sessão usando o Amazon CloudWatc h Logs.	18 de novembro de 2020
Novo tópico: Verificar a assinatura do SSM Agent	Agora você pode verificar a assinatura criptográfica do pacote do instalador do SSM Agent em instâncias do Linux. Para obter mais informações, consulte Esquemas e recursos de documentos do SSM.	17 de novembro de 2020

Novo tópico: Noções básicas sobre status de automação	Um novo tópico foi adicionado ao capítulo do Systems Manager Automation, que descreve os status de ações e automações. Para obter mais informações, consulte Noções básicas do status de automação.	17 de novembro de 2020
Novos tipos de origem para o plugin `aws:downloadContent`	O Git e o HTTP agora são compatíveis como tipos de fonte para o plugin do `aws:downloadContent`. Para obter mais informações, consulte `aws:downloadContent`.	17 de novembro de 2020
Novo recurso de esquema de documento do Systems Manager (documento SSM)	Em documentos do SSM com a versão 2.2 ou superior do esquema, o parâmetro `precondition` agora suporta referenciar os parâmetros de entrada do documento. Para obter mais informações, consulte Esquemas e recursos de documentos do SSM.	17 de novembro de 2020

Nova origem dos dados no Explorer: AWS Config	O Explorer agora exibe informações sobre a conformidade AWS Config, incluindo um resumo geral das regrass AWS Config de conformidade e não-conformidade, o número de recursos compatíveis e não compatíveis e detalhes específicos sobre cada um (quando você aprofunda em uma regra ou recurso não compatível). Para obter mais informações, consulte Editar as fontes de dados do Systems Manager Explorer.	11 de novembro de 2020
Novo tópico: Executando grupos Auto Scaling com associações	Uma nova seção foi adicionada ao State Manager, que descreve as práticas recomendadas para criar associações para executar grupos de Auto Scaling. Para obter mais informações, consulte Executar grupos Auto Scaling com associações.	10 de novembro de 2020
O Quick Setup agora é compatível com o direcionamento de um grupo de recursos	O Quick Setup agora é compatível com a escolha de um grupo de recursos como destino para o tipo de configuração local. Para obter mais informações, consulte Escolher destinos para o Quick Setup.	5 de novembro de 2020

O Patch Manager adiciona suporte ao Debian Server 10 LTS, Oracle Linux 7.9 LTS e Ubuntu Server 20.10 STR	Agora é possível usar o Patch Manager para aplicar patches a instâncias do Debian Server 10 LTS, Oracle Linux 7.9 LTS e Ubuntu Server 20.10 STR. Para obter mais informações, consulte os tópicos a seguir. • Pré-requisitos do Patch Manager • Como os patches de segurança são selecionados • Como os patches são instalados • Como as regras de lista de referência de patches funcionam no Debian Server • Como as regras de lista de referência de patches funcionam no Oracle Linux • Como as regras de lista de referência de patches funcionam no Ubuntu Server	4 de novembro de 2020

Novo suporte do EventBridge ao AWS Systems Manager Change Calendar	O Amazon EventBridge agora fornece suporte para eventos Change Calendar em regras de evento. Quando o estado de um calendário muda, o EventBridge pode iniciar a ação de destino que você definiu como uma regra do EventBridge. Para obter informações sobre como trabalhar com eventos do EventBridge e do Systems Manager, consulte os tópicos a seguir. • Configurar o EventBridge para eventos do Systems Manager • Referência: padrões e tipos de eventos do Amazon EventBridge para Systems Manager	4 de novembro de 2020

Configure o CloudWatch para criar OpsItems dos alarmes	Você pode configurar o Amazon CloudWatch para criar automaticamente um OpsItem no Systems Manager OpsCenter quando um alarme entrar no estado `ALARM`. Isso permite que você diagnostique e corrija problemas com o recursos do AWS em um único console. Para obter mais informaçõ es, consulteConfigurar o CloudWatch para criar OpsItems nos alarmes.	4 de novembro de 2020
Suporte ao Ubuntu Server 20.10	O AWS Systems Manager agora oferece suporte ao Ubuntu Server 20.10 versão de curto prazo (STR). Para obter mais informações, consulte os tópicos a seguir. • Sistemas operacionais com suporte • Instalar o SSM Agent para um ambiente híbrido (Linux) • Instalar manualmente o SSM Agent em instâncias do Ubuntu Server • Verificar o status do SSM Agent e iniciar o agente	22 de outubro de 2020

Novo tópico: Permitir perfis de shell configuráveis	Agora você pode permitir perfis de shell configuráveis com o Session Manager. Ao permitir perfis de shell configuráveis, você pode personalizar preferências dentro de sessões como preferências de shell, variáveis de ambiente, diretórios de trabalho e executar vários comandos quando uma sessão é iniciada. Para obter mais informações, consulte Permitir perfis de shell configuráveis.	21 de outubro de 2020
Os resultados da conformidade de patches agora relatam quais CVEs são resolvidos por quais patches	Para a maioria dos sistemas Linux compatíveis, quando você visualiza os resultados de conformidade de patches para as instâncias gerenciadas, os detalhes que você pode visualizar agora relatam quais problemas do boletim Common Vulnerabilities and Exposure (CVE) são resolvidos pelos patches disponíveis. Essas informações podem ajudar você a determinar com que urgência você precisa instalar um patch ausente ou com falha. Para obter mais informações, consulte Visualizar os resultados da conformidade do patch.	20 de outubro de 2020

Suporte expandido para metadados de patch do Linux	Agora você pode ver muitos detalhes sobre os patches do Linux disponíveis no Patch Manager. Você pode optar por visualizar os dados do patch, como arquitetura, época, versão, ID do CVE, ID do Advisory, ID do Bugzilla, repositório e muito mais. Além disso, a operação da API DescribeAvailablePatches foi atualizada para oferecer suporte a sistemas operacionais Linux e à filtragem de acordo com esses novos tipos de metadados de patch disponíveis. Para obter mais informações, consulte os tópicos a seguir. • Visualizar patches disponíveis • DescribeAvailablePatches e Patch na AWS Systems Manager API Reference • describe-available-patches na seção AWS Systems Manager da AWS CLI Command Reference	16 de outubro de 2020

O plugin Session Manager para a AWS CLI versão 1.2.7.0	Uma nova versão do plugin do Session Manager para a AWS CLI foi liberada. Para obter mais informações, consulte Versão mais recente do plugin do Session Manager e histórico de versões.	15 de outubro de 2020
Novo tópico: Esquema do documento de sessão	Novo tópicoEsquema do documento dedescreve os elementos do esquema de um documento Session. Essas informações podem ajudar você a criar documentos de Sessão personalizados, onde você especifica preferênc ias para os tipos de sessões que você usa com o Session Manager.	15 de outubro de 2020
Novo tópico: Pesquisa de texto livre para documentos SSM	A caixa de pesquisa na página Documentos do Systems Manager agora oferece suporte à pesquisa de texto livre. A pesquisa de texto livre compara o termo ou termos de pesquisa inseridos com o nome do documento em cada documento do SSM. Para obter mais informações, consulte Usar a pesquisa de texto livre.	15 de outubro de 2020

Novo tópico: Solução de problemas de disponibilidade de instâncias gerenciadas do Amazon EC2	Novo tópicoSolução de problemas de disponibilidade de instância gerenciada do Amazonajuda a investigar por que uma instância do Amazon EC2 confirmada está em execução não está disponíve l nas listas de instâncias gerenciadas disponíveis no Systems Manager.	6 de outubro de 2020
Reorganização do capítulo sobre Parameter Store	Para ajudar você a encontrar as informações de que precisa de forma mais eficiente, reorganizamos o conteúdo no capítulo Parameter Store do Guia do usuário do AWS Systems Manager. A maior parte do conteúdo agora está organizada nas seções Configurar o Parameter Store e Trabalhar com o Parameter Store. Além disso, o tópico AWS Systems ManagerPa rameter Store foi expandido para incluir as seguintes seções: • Como o Parameter Store beneficia minha organizaç ão? • Quem deve usar o Parameter Store? • Quais são os recursos do Parameter Store? • O que é um parâmetro ?	1.º de outubro de 2020

Novos tópicos relacionados à conformidade de patch	Os tópicos a seguir foram adicionados para ajudar você a identificar instâncias gerenciadas que estão fora de conformidade de patches, entender os diferentes tipos de verificações de conformidade de patches e tomar as etapas apropriadas para colocar suas instâncias em conformidade. • Identificar instâncias fora de conformidade • Aplicação de patches em instâncias fora de conformidade • Visualizar resultados de conformidade de patches	24 de setembro de 2020
SSM Agent versão 3.0	O Systems Manager lançou uma nova versão do SSM Agent.	21 de setembro de 2020

Tópicos novos e atualizados: O Amazon EventBridge substitui o CloudWatch Events para gerenciamento de eventos	O CloudWatch Events e o EventBridge são o mesmo serviço subjacente e API, mas o EventBridge oferece mais recursos e é agora a forma preferida de gerenciamento de eventos na AWS. (As alterações feitas no CloudWatch ou no EventBridge aparecem em cada console) As referências ao CloudWatch Events e procedimentos existentes em todo o Guia do usuário do AWS Systems Manager foram atualizadas para refletir o suporte do EventBridge. Além disso, os seguintes novos tópicos foram adicionados: • Monitorar eventos do Systems Manager • Configurar o EventBridge para eventos do Systems Manager • Exemplos de tipo de alvo do Systems Manager • Referência: padrões e tipos de eventos do Amazon EventBridge para Systems Manager	18 de setembro de 2020

Integração do AWS Security Hub e do Patch Manager	Agora, você pode integrar o Patch Manager ao AWS Security Hub. O Security Hub fornece uma visão abrangente do estado de segurança na AWS e ajuda a verificar o ambiente de acordo com os padrões e as práticas recomendadas do setor de segurança. Quando integrado ao Patch Manager, o Security Hub monitora o status da aplicação de patches das frotas, do ponto de vista da segurança. Para obter mais informações, consulte Integrating Patch Manager with AWS Security Hub (Integrar com).	17 de setembro de 2020

Pseudo parâmetros da janela de manutenção: Novos tipos de recurso compatíveis com o{{TARGET_ID}} e{{RESOURCE_ID}}	Ao registrar uma tarefa da janela de manutenção, use a opção `--task-invocation-parameters` para especificar os parâmetros que são exclusivos para cada um dos quatro tipos de tarefa. Você também pode fazer referência a determinados valores usando a sintaxe de pseudoparâmetro, como `{{TARGET_ID}}` e `{{RESOURCE_ID}}` . Quando a tarefa de janela de manutenção é executada, ela envia os valores corretos em vez dos espaços reservados do pseudoparâmetro. Dois tipos de recursos adicionais agora estão disponíveis para uso com os pseudoparâmetros `{{TARGET_ID}}` e `{{RESOURCE_ID}}` . Agora é possível usar os tipos de recursos `AWS::RDS::DBInstance` e `AWS::SSM::ManagedInstance` com esses dois pseudoparâmetros. Para obter mais informações sobre os pseudoparâmetros da janela de manutenção, consulte Usar pseudoparâmetros ao registrar tarefas da janela de manutenção.	14 de setembro de 2020

Aplique patches em instâncias sob demanda com a nova opção "Aplicar patch agora"	Agora você pode usar o console do Systems Manager para corrigir instâncias ou verificar se há patches ausentes, a qualquer momento. Você pode fazer isso sem ter que criar ou modificar um agendamento, ou especificar opções completas de configuração de patches para acomodar uma necessidade imediata de correção. Você só precisa especificar se deseja verificar ou instalar patches e identificar as instâncias de destino para a operação. O Patch Manager aplica automaticamente a lista de referência de patch padrão atual para seus tipos de instância e aplica opções de práticas recomendadas para quantas instâncias forem corrigidas de uma só vez e quantos erros forem permitidos antes da operação falhar. Para obter mais informações, consulte Aplicar patch em instâncias sob demanda.	9 de setembro de 2020

Novo tópico: Verificar o status do SSM Agent e iniciar o agente	O novo tópico Verificar o status de SSM Agent e iniciar o agente fornece comandos para verificar se o SSM Agent está sendo executado em cada sistema operacional compatível. Ele também fornece os comandos para iniciar o agente se ele não estiver em execução.	7 de setembro de 2020
O Patch Manager agora oferece suporte ao Ubuntu Server 20.04 LTS	Agora é possível usar o Patch Manager para aplicar patches em instâncias do Ubuntu Server 20.04. Para obter mais informações, consulte os tópicos a seguir. • Como os patches de segurança são selecionados • Como os patches são instalados • Como as regras de lista de referência de patches funcionam no Ubuntu Server	31 de agosto de 2020
Novo tópico para casos de uso e práticas recomendadas	Adicionamos um novo tópico para ajudar os usuários a entender rapidamen te as diferenças entre o Maintenance Windows e o State Manager. Para obter mais informações, consulte Escolher entre o State Manager e a Maintenance Windows.	28 de agosto de 2020

Novos recursos do OpsCenter	O OpsCenter inclui novos recursos para ajudar você a localizar e executar rapidamente os runbooks de automação para corrigir problemas. Para obter mais informações, consulte Recursos do runbook de automação no OpsCenter.	19 de agosto de 2020
Nova fonte de dados no Explorer: casos do AWS Support	O Explorer agora exibe informações sobre casos do Suporte. Você deve ter uma conta Enterprise ou Business configurada com Suporte. Para obter mais informações, consulte Editar as fontes de dados do Systems Manager Explorer.	13 de agosto de 2020
O Distributor agora fornece um pacote de terceiros da Trend Micro.	O Distributor agora inclui um pacote de terceiros da Trend Micro. Você pode usar o Distributor para instalar o agente do Trend Micro Cloud One em suas instâncias gerenciadas. O Trend Micro Cloud One ajuda você a proteger suas cargas de trabalho na nuvem. Para obter mais informações, consulte AWSDistributor.	12 de agosto de 2020

O plugin de documentos `aws:configurePackage` agora inclui o parâmetro AdditionalArguments.	O plugin de documentos `aws:configurePackage` do Systems Manager agora suporta o fornecimento de parâmetros adicionais para seus scripts (instalar, desinstalar e atualizar) com o novo parâmetro `additionalArguments`. Para obter mais informações, consulte o tópico `aws:configurePackage`	11 de agosto de 2020
O conteúdo do AppConfig movido para um manual do usuário separado	Informações sobre o AWS AppConfig foram movidas para um manual do usuário separado. Para obter mais informações, consulte What Is AWSAppConfig? (O que é o ?) AppConfig também tem uma página inicial de documentação separada com links para o guia do usuário, a referência da API do AppConfig e um novo workshop do AppConfig.	3 de agosto de 2020

O Quick Setup agora oferece suporte para AWS Organizations	O Quick Setup agora oferece suporte ao AWS Organizations, permitindo que você configure rapidamente os perfis de segurança necessários e as ferramentas do Systems Manager comumente usadas em várias contas e regiões. Para obter mais informações, consulte AWS Systems ManagerQuick Setup.	23 de julho de 2020
Nova fonte de dados no Explorer: conformidade de associações	O Explorer agora exibe dados de conformidade de associação do State Manager. Para obter mais informações, consulte Editar as fontes de dados do Systems Manager Explorer.	23 de julho de 2020

Novo documento de comandos do Systems Manager para ativar e desativar o Kernel Live Patching	O documento `AWS-ConfigureKernelLivePatching` agora está disponível para uso com o Run Command quando você deseja ativar ou desativar o Kernel Live Patching em instâncias do Amazon Linux 2. Este documento substitui a necessidade de criar seus próprios documentos de Comando personalizados para essas tarefas. Para obter informações, consulte Usar o Kernel Live Patching em instâncias do Amazon Linux 2	22 de julho de 2020
Cotas de automação atualizadas	As cotas de serviço para automação foram atualizadas, incluindo uma fila separada para automações de controle de taxa. Para obter mais informações, consulte Automação do AWS Systems Manager.	20 de julho de 2020

Especificar o número de dias de deslocamento de programação para uma janela de manutenção usando o console	Usando o console do Systems Manager, você agora pode especificar um número de dias de espera após a data e a hora especificadas por uma expressão CRON antes de executar a janela de manutenção. (Anteriormente, essa opção estava disponível somente ao usar um AWS SDK ou uma ferramenta de linha de comando). Por exemplo, se a expressão CRON agenda a execução de uma janela de manutenção na terceira terça-feira de cada mês, às 23h30 – `cron(0 30 23 ? * TUE#3 *)` – e você especifica um deslocamento de programação de 2, a janela só será executada dois dias depois às 23h30. Para obter mais informações, consulte Expressões cron e rate do Systems Manager e Especificar o número de dias de deslocamento de programação para uma janela de manutenção.	17 de julho de 2020

Atualizar o PowerShell usando o Run Command	Para ajudar você a atualizar o PowerShell para a versão 5.1 em suas instâncias do Windows Server R2 2012 e 2012, adicionamos um passo a passo ao Guia do usuário do AWS Systems Manager. Para obter mais informaçõ es, consulte Atualizar o PowerShell usando o Run Command.	30 de junho de 2020
O Patch Manager agora é compatível com o CentOS 8.0 e 8.1	Agora é possível usar o Patch Manager para aplicar patches às instâncias do CentOS 8.0 e 8.1. Para obter mais informaçõ es, consulte os tópicos a seguir: • Como os patches de segurança são selecionados • Como os patches são instalados • Como as regras de lista de referência de patches funcionam no CentOS • Instalar manualmente o SSM Agent em instâncias do CentOS • Como instalar o SSM Agent em nós híbridos do Linux	27 de junho de 2020

O AppConfig integra-se ao AWS CodePipeline	O AppConfig é uma ação de implantação integrada para o AWS CodePipeline (CodePipeline). O CodePipeline é um serviço de entrega contínua totalmente gerenciado que ajuda a automatizar os pipelines de lançamento para atualizações rápidas e confiáveis de aplicações e infraestrutura. O CodePipeline automatiza as fases de compilação, teste e implantação do processo de lançamento, sempre que ocorrer uma alteração de código, de acordo com os modelos definidos para o processo de lançamento. Integração do AppConfig com o CodePipeline oferece os benefícios a seguir. Para obter mais informações, consulte Integração do AppConfig com o CodePipeline. • Os clientes que usam o CodePipeline para gerenciar a orquestração agora têm um meio leve de implantar alterações de configuração nas aplicações, sem precisar implantar toda a base de código. • Os clientes que quiserem usar o AppConfig para	25 de junho de 2020

	gerenciar implantações de configuração, mas que se sentem limitados porque o AppConfig não é compatível com o código atual ou com o armazenamento de configuração, agora têm outras opções. O CodePipeline oferece suporte ao AWS CodeCommit, ao GitHub e ao BitBucket (apenas para citar alguns).	
Novo capítulo: Integrações de produtos e serviços	Para ajudar você a entender como o Systems Manager se integra a Serviços da AWS e a outros produtos e serviços, um novo capítulo foi adicionado ao Guia do usuário do AWS Systems Manager. Para obter mais informações, consulte Integrações de produtos e serviços ao Systems Manager.	23 de junho de 2020
Reorganização do capítulo de automação	Para ajudar você a encontrar o que precisa, reorganizamos tópicos no capítulo Automação doGuia do usuário do AWS Systems Manager. Por exemplo, as ações de automação e as referências de runbooks de automação agora são seções de nível superior no capítulo. Para obter mais informações, consulte Automação do AWS Systems Manager.	23 de junho de 2020

Especificar o número de dias de deslocamento de programação para uma janela de manutenção	Usando uma ferramenta de linha de comando ou o AWS SDK, agora é possível especificar um número de dias de espera após a data e a hora especificadas por uma expressão CRON antes de executar a janela de manutenção. Por exemplo, se a expressão CRON agenda a execução de uma janela de manutenção na terceira terça-feira de cada mês, às 23h30 – `cron(0 30 23 ? * TUE#3 *)` – e você especifica um deslocamento de programação de 2, a janela só será executada dois dias depois às 23h30. Para obter mais informações, consulte Expressões cron e rate do Systems Manager e Especificar o número de dias de deslocamento de programação para uma janela de manutenção.	19 de junho de 2020

Suporte do Patch Manager para o Kernel Live Patching em instâncias do Amazon Linux 2	O Kernel Live Patching para Amazon Linux 2 permite que você aplique vulnerabilidades de segurança e patches de erros críticos a um kernel do Linux em execução, sem reinicializações ou interrupç ões a aplicações de execução. Agora você pode permitir o recurso e aplicar patches dinâmicos do kernel usando o Patch Manager. Para obter informações, consulte Usar o Kernel Live Patching em instâncias do Amazon Linux 2.	16 de junho de 2020
O Patch Manager aumenta o suporte a versões do Oracle Linux	Anteriormente, o Patch Manager oferecia suporte apenas à versão 7.6 do Oracle Linux. Conforme listado nos pré-requisitos do Patch Manager, o suporte agora abrange as versões 7.5 a 7.8.	16 de junho de 2020

Cenário de exemplo do uso do`InstallOverrideList` Parâmetro nas operações de aplicação de patches	O novo tópico Cenário de exemplo do uso do parâmetro `InstallOverrideList` descreve uma estratégia para usar o parâmetro `InstallOverrideList` no documento `AWS-RunPatchBaseline` para aplicar diferentes tipos de patches a um grupo de destino, em diferentes programações de janelas de manutenção, enquanto ainda usa uma única lista de referência de patches.	11 de junho de 2020
Estratégias de implantação predefinidas para o AppConfig	O AppConfig agora oferece estratégias de implantação predefinidas. Para obter mais informações, consulte Criar uma estratégia de implantação.	10 de junho de 2020

O Patch Manager agora oferece suporte para Red Hat Enterprise Linux (RHEL) 7.8 a 8.2	Agora é possível usar o Patch Manager para aplicar patches às instâncias do RHEL 7.8 a 8.2. Para obter mais informaçõ es, consulte os tópicos a seguir: • Como os patches de segurança são selecionados • Como os patches são instalados • Como as regras de lista de referência de patches funcionam no RHEL • Instalar manualmente o SSM Agent em instâncias do Red Hat Enterprise Linux • Como instalar o SSM Agent em nós híbridos do Linux	9 de junho de 2020

O Explorer é compatível com a administração delegada	Se você agregar dados do Explorer de várias Regiões da AWS e Contas da AWS usando a sincronização de dados de recursos com o AWS Organizations, sugerimos que configure um administrador delegado para o Explorer. Um administrador delegado melhora a segurança do Explorer limitando o número de administradores do Explorer que podem criar ou excluir sincronizações de dados de recurso de várias contas e regiões a apenas um indivíduo. Também não é mais necessário estar conectado à conta de gerenciamento do AWS Organizations para administrar as sincronizações de dados de recursos no Explorer. Para obter mais informações, consulte Configurar um administrador delegado.	3 de junho de 2020

Aplicar a associação do State Manager somente no próximo intervalo do Cron especificado	Se você não quiser que uma associação do State Manager seja executada imediatamente após sua criação, selecione a opção Apply association only at the next specified Cron interval (Aplicar associação somente no próximo intervalo Cron) especificada no console do Systems Manager. Para obter mais informações, consulte Criar associações.	3 de junho de 2020
Nova origem dos dados no Explorer: AWS Compute Optimizer	O Explorer agora exibe dados de AWS Compute Optimizer. Isso inclui uma contagem de instâncias do EC2, descobert as de otimização, detalhes de preço sob demanda e recomendações para preço e tipo de instância Subprovisionados e Superprov isionados. Para obter mais informações, consulte os detalhes para configuração do AWS Compute Optimizer em Configurando serviços relacionados.	26 de maio de 2020

Instalar as atualizações de versões secundárias do Windows Service Packs e do Linux usando o Patch Manager	O novo tópico Tutorial: criar uma lista de referência de patches para instalar o Windows Service Packs (console) demonstra como você pode criar uma lista de referência de patches dedicada exclusivamente à instalação do Windows Service Packs. O tópico Criar uma lista de referência de patches personalizada (Linux) foi atualizado com informações sobre como incluir atualizações de versões secundárias para sistemas operacionais Linux nas listas de referência de patches.	21 de maio de 2020
Reorganização do capítulo sobre Parameter Store	Todos os tópicos que lidam com a configuração ou a definição de opções para operações do Parameter Store foram consolidados na seção Setting up Parameter Store (Configurar o Parameter Store). Isso inclui os tópicos Gerenciar camadas de parâmetros e Aumentar a throughput do Parameter Store, que foram realocados de outras partes do capítulo.	18 de maio de 2020

Novo tópico para criar strings de data e hora para interagir com operações da API do Systems Manager.	O novo tópico Criar strings de data e hora formatadas para o Systems Manager descreve como criar strings de data e hora formatadas para interagir com operações da API do Systems Manager.	13 de maio de 2020
Sobre permissões para criptografar parâmetros SecureString	O novo tópico Restringir acesso aos parâmetros do Systems Manager usando políticas do IAM explica a diferença entre criptografar os parâmetros `SecureString` usando um AWS KMS key e usar o Chave gerenciada pela AWS fornecido pela AWS.	13 de maio de 2020

O Patch Manager agora oferece suporte aos sistemas operacionais Debian Server e Oracle Linux 7.6	Agora é possível usar o Patch Manager para aplicar patches a instâncias do Debian Server e do Oracle Linux. O Patch Manager oferece suporte à aplicação de patches às versões do Debian Server 8.x e 9.x e Oracle Linux 7.6. Para obter mais informações, consulte os tópicos a seguir. • Como os patches de segurança são selecionados • Como os patches são instalados • Como as regras de lista de referência de patches funcionam no Debian Server • Como as regras de lista de referência de patches funcionam no Oracle Linux	7 de maio de 2020
Criar associações do State Manager direcionadas ao AWS Resource Groups	Além de segmentar tags, instâncias individuais e todas as instâncias em sua conta da Conta da AWS, agora você pode criar associações do State Manager direciona das a instâncias no AWS Resource Groups. Para obter mais informações, consulte Sobre destinos e controles de taxa em associações do State Manager	7 de maio de 2020

Novo tipo de dados `aws:ec2:image` no Parameter Store para validar IDs de AMI	Ao criar um parâmetro `String`, você pode agora especificar um tipo de dado, como `aws:ec2:image` , para garantir que o valor do parâmetro inserido seja um formato válido de ID da Amazon Machine Image (AMI). O suporte para formatos de ID de AMI significa que você não precisa atualizar todos os scripts e modelos com um novo ID sempre que a AMI que deseja usar em seus processos for alterada. Você pode criar um parâmetro com o tipo de dados `aws:ec2:image` e, em seu valor, inserir o ID de uma AMI. Esta é a AMI a partir da qual você deseja que novas instância s sejam criadas. Depois, você faz referência a esse parâmetro em seus modelos e comandos. Quando estiver pronto para usar uma AMI diferente, atualize o valor do parâmetro. O Parameter Store valida o novo ID da AMI e você não precisa atualizar seus scripts e modelos. Para obter mais informações, consulte Native parameter support for Amazon Machine Image IDs (Suporte	5 de maio de 2020

	a parâmetros nativos para IDs da imagem de máquina da Amazon).	
Gerenciar códigos de saída em comandos do Run Command	O Run Command permite definir como os códigos de saída são manipulados nos scripts. Por padrão, o código de saída do último comando executado em um script é relatado como o código de saída de todo o script. No entanto, você pode incluir uma instrução condicional shell para sair do script se algum comando antes do final falhar usando a abordagem a seguir. Para obter exemplos, consulte o novo tópico Gerenciando códigos de saída em comandos do Run Command.	5 de maio de 2020

Novos parâmetros públicos lançados para zonas de disponibilidade e zonas locais	Parâmetros públicos foram lançados para disponibilizar informações sobre zonas de disponibilidade da AWS e zonas locais programat icamente. Eles são além dos parâmetros públicos de infraestrutura global existente s para Serviços da AWS e Regiões da AWS. Para obter mais informações, consulte Chamar parâmetros públicos para Serviços da AWS, regiões, endpoints, zonas de disponibilidade, zonas locais e zonas do Wavelength.	4 de maio de 2020
Nova origem dos dados no Explorer: AWS Trusted Advisor	O Explorer agora exibe dados de AWS Trusted Advisor. Isso inclui o status das verificações de práticas recomendadas e recomendações nas seguintes áreas: otimização de custos, segurança, tolerância a falhas, performance e Service Quotas. Para obter mais informações, consulte os detalhes para configuração do Trusted Advisor em Configura ndo serviços relacionados.	4 de maio de 2020

Criar associações do State Manager que executam receitas do Chef	É possível criar associaçõ es do State Manager que executam receitas do Chef usando o documento `AWS-ApplyChefRecipes` . Esse documento oferece os seguintes benefícios para a execução de receitas do Chef: • Compatível com várias versões do Chef (Chef 11 a Chef 14). • Instala automaticamente o software cliente Chef em instâncias de destino. • Opcionalmente, executa as verificações de conformid ade do Systems Manager nas instâncias de destino e armazena os resultados das verificações de conformid ade em um bucket do S3. • Executa vários livros de receitas e receitas em uma única execução do documento. • Opcionalmente, executa receitas no modo `why-run`, para mostrar quais receitas serão alteradas nas instâncias de destino sem fazer alterações. • Opcionalmente aplica atributos JSON personali	19 de março de 2020

	zados a execuções do `chef-client` . Para obter mais informaçõ es, consulte Criar associaçõ es que executam receitas do Chef.	
Sincronizar dados de inventári o de várias Contas da AWS para um bucket central do Amazon S3	Você pode sincronizar os dados do Systems Manager Inventory em várias Contas da AWS em um bucket central do S3. As contas devem estar definidas no AWS Organizations. Para obter mais informações, consulte Criar uma sincronização de dados de recursos do inventári o para várias contas definidas na AWS Organizations.	16 de março de 2020
Armazenar configurações do AppConfig no Amazon S3	Anteriormente, o AppConfig era compatível apenas com as configurações da aplicação que eram armazenadas nos documentos do Systems Manager (SSM) ou em parâmetros do Parameter Store. Além dessas opções, o AppConfig agora oferece suporte ao armazenamento de configurações no Amazon S3. Para obter mais informações, consulte Sobre configurações armazenadas no Amazon S3.	13 de março de 2020

O SSM Agent instalado por padrão nas AMIs otimizadas para o Amazon ECS	O SSM Agent agora é instalado por padrão nas AMIs otimizadas para o Amazon ECS. Para obter mais informações, consulte Trabalhar com SSM Agent.	25 de fevereiro de 2020
Criar configurações do AppConfig no console	O AppConfig agora permite que você crie uma configuração de aplicação no console quando criar um perfil de configuração. Para obter mais informações, consulte Criar uma configuração e um perfil de configuração.	13 de fevereiro de 2020

Aprovar automaticamente somente os patches liberados até uma data especificada	Além da opção de aprovação automática de patches para instalação por um número especificado de dias após o lançamento, o Patch Manager agora oferece suporte à capacidade de aprovar automaticamente somente patches liberados em uma data especificada por você ou antes dela. Por exemplo, se você especificar 7 de julho de 2020 como a data limite na linha de base do patch, nenhum patch liberado em 8 de julho de 2020 ou após essa data será instalado automatic amente. Para obter mais informações, consulte Sobre listas de referência personali zadas e Trabalhar com listas de referência personalizadas (console).	12 de fevereiro de 2020

Usar o pseudoparâmetro {{RESOURCE_ID}} nas tarefas da janela de manutenção	Ao registrar uma tarefa da janela de manutenção, especifique os parâmetros que são exclusivos para o tipo de tarefa. Também é possível fazer referência a determinados valores usando sintaxe de pseudoparâmetro, como `{{TARGET_ID}}`, `{{TARGET_TYPE}}` e `{{WINDOW_TARGET_ID}}`. Quando a tarefa de janela de manutenção é executada, ela envia os valores corretos em vez dos espaços reservados do pseudoparâmetro. Para oferecer suporte a recursos que fazem parte de um grupo de recursos como destino, é possível usar o pseudoparâmetro `{{RESOURCE_ID}}` para passar valores para recursos, como tabelas do DynamoDB, buckets do S3 e outros tipos compatíveis. Para obter mais informações, consulte os seguintes tópicos no Tutorial: Criar e configurar uma janela de manutenção (AWS CLI): • Usar pseudoparâmetros ao registrar tarefas da janela de manutenção	6 de fevereiro de 2020

	• Exemplos: Registrar tarefas em uma janela de manutenção	
Executar comandos novamente com rapidez	O Systems Manager inclui duas opções para ajudar você a executar novamente um comando na página Run Command no console do AWS Systems Manager. Rerun (Reexecutar): este botão permite que você execute o mesmo comando sem fazer alterações nele. Copy to new (Copiar para novo): este botão copia as configurações de um comando para um novo comando e dá a você a opção de editar essas configura ções antes de executá-lo. Para obter mais informações, consulte Executar comandos novamente.	5 de fevereiro de 2020

Revesão do nível de instâncias avançadas para o nível de instâncias padrão	Se você configurou anteriormente todas as instâncias on-premises em execução no ambiente híbrido a fim de usar a camada de instâncias avançadas, agora será possível configurar rapidamente essas instâncias para usar a camada de instância padrão. A reversão para o nível de instâncias padrão se aplica a todas as instâncias híbridas em uma Conta da AWS e em uma única Região da AWS. A reversão para o nível de instâncias padrão afeta a disponibilidade de algumas ferramentas do Systems Manager. Para obter mais informações, consulte Reverter do nível de instâncias avançadas para o nível de instâncias padrão.	16 de janeiro de 2020

Nova opção para ignorar reinicializações da instância após a instalação do patch	Anteriormente, as instâncias gerenciadas sempre eram reinicializadas depois que o Patch Manager instalava patches nelas. Um novo parâmetro `RebootOption` no documento `AWS-RunPatchBaseline` do SSM permite especificar se você deseja ou não que as instâncias sejam reinicializadas automaticamente após a instalação de novos patches. Para obter mais informações, consulte Nome do parâmetro: RebootOption no tópico Sobre o documento `AWS-RunPatchBaseline` do SSM.	15 de janeiro de 2020
Novo tópico: 'Execução de scripts do PowerShell em instâncias do Linux'	Um novo tópico que descreve como usar Run Command para executar scripts do PowerShell em instâncias do Linux. Para obter mais informações, consulte Executar scripts do PowerShell em instâncias do Linux.	10 de janeiro de 2020

Atualizações para 'configurar o SSM Agent para usar um proxy'	Os valores a serem especificados ao configurar o SSM Agent para usar um proxy foram atualizados a fim de refletir opções para servidores de proxy HTTP e servidores de proxy HTTPS. Para obter mais informações, consulte Configurar o SSM Agent para usar um proxy.	9 de janeiro de 2020

O novo capítulo “Segurança” descreve as práticas para proteger recursos do Systems Manager	Um novo capítulo sobre Segurança no AWS Systems Manager Guia do usuário do ajuda a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Systems Manager. Os tópicos a seguir mostram como configurar o Systems Manager para atingir seus objetivos de segurança e conformidade. Saiba também como usar outros Serviços da AWS que ajudam a monitorar e proteger os recursos do Systems Manager. **Note** Como parte desta atualização, o capítulo do guia do usuário “Autenticação e controle de acesso” foi substituído por uma seção nova e mais simples, Identity and Access Management para o AWS Systems Manager.	24 de dezembro de 2019

Novos runbooks personalizados de exemplo	Um conjunto de runbooks personalizados de automação de exemplo foi adicionado ao manual do usuário. Esses exemplos mostram como usar várias ações de automação para simplificar tarefas de implantação, solução de problemas e manutenção e destinam-se a ajudar você a escrever seus próprios manuais de automação personalizados. Para obter mais informações, consulte Custom Automation runbook samples (Exemplos de runbooks do Automation personalizados). Você também pode visualizar o conteúdo do runbook de automação gerenciado pela Amazon no console do Systems Manager. Para obter mais informações, consulte Referência de runbooks do Systems Manager Automation.	23 de dezembro de 2019

Suporte ao Oracle Linux	O Systems Manager agora oferece suporte ao Oracle Linux 7.5 e 7.7. Para obter informações sobre a instalaçã o manual do SSM Agent em instâncias do EC2 para instâncias do Oracle Linux, consulte Oracle Linux. Para obter informações sobre a instalação do SSM Agent em servidores Oracle Linux em um ambiente híbrido, consulte Como instalar o SSM Agent em nós híbridos do Linux.	19 de dezembro de 2019

Executar sessões do Session Manager do console do Amazon EC2	Agora é possível iniciar sessões do Session Manager no console do Amazon Elastic Compute Cloud (Amazon EC2). Trabalhar com tarefas relacionadas a sessões do console do Amazon EC2 requer permissões do IAM diferentes para usuários e administradores. Você pode fornecer permissões para usar apenas o console do Session Manager e a AWS CLI, o console do Amazon EC2 ou todas as três ferramentas. Para obter mais informações, consulte os tópicos a seguir. • Políticas padrão do IAM do Quickstart para o Session Manager • Iniciar uma sessão (console do Amazon EC2)	18 de dezembro de 2019

Suporte do CloudWatch para métricas e alarmes do Run Command	O AWS Systems Manager agora publica métricas sobre o status de comandos do Run Command no CloudWatch, permitindo definir alarmes com base nessas métricas. Os valores de status do terminal de comandos para os quais é possível rastrear métricas incluem `Success`, `Failed` e `Delivery Timed Out`. Para obter mais informaçõ es, consulte Monitoring Run Command metrics using Amazon CloudWatch (Monitorar as métricas do Run Command usando o Amazon CloudWatch).	17 de dezembro de 2019
Nova ferramenta do Systems Manager: Change Calendar	Use o Change Calendar do Systems Manager para especificar períodos (eventos) durante os quais você deseja limitar ou impedir alterações de código (como runbooks do Systems Manager Automation ou funções do AWS Lambda) nos recursos. Um calendário de alterações é um novo tipo de documento do Systems Manager que armazena dados do iCalendar 2.0 em formato de texto simples. Para obter mais informações, consulte Calendário de alterações do AWS Systems Manager.	11 de dezembro de 2019

Nova ferramenta do Systems Manager: AWSAppConfig	Use o AppConfig para criar, gerenciar e implantar rapidamente configurações de aplicativos. O AppConfig oferece suporte a implantaç ões controladas em aplicativ os de qualquer tamanho. Você pode usar o AppConfig com aplicativos hospedados em instâncias do EC2, AWS Lambda, contêineres, aplicativ os móveis ou dispositivos de IoT. Para evitar erros ao implantar configurações de aplicativos, o AppConfig inclui validadores. Um validador fornece uma verificação sintática ou semântica para garantir que a configuração que você deseja implantar funcione conforme pretendid o. Durante uma implantação de configuração, o AppConfig monitora o aplicativo para garantir que a implantação seja bem-sucedida. Se o sistema encontrar um erro ou se a implantação iniciar um alarme, o AppConfig reverterá a alteração para minimizar o impacto para os usuários da aplicação. Para obter mais informações, consulte AWSAppConfig.	25 de novembro de 2019

Nova ferramenta do Systems Manager: Systems Manager Explorer	O AWS Systems Manager Explorer é um painel de operações personalizável que informa sobre seus recursos da AWS. O Explorer exibe uma visualização agregada dos dados de operações (OpsData) para as Contas da AWS em todas as Regiões da AWS. No Explorer, os OpsData incluem metadados sobre suas instâncias do EC2, detalhes de conformidade de patches e itens de trabalho operacionais (OpsItems). O Explorer fornece contexto sobre como os OpsItems são distribuídos em suas unidades de negócios ou aplicativos, a tendência ao longo do tempo e como eles variam de acordo com a categoria. Você pode agrupar e filtrar informações no Explorer para se concentra r em itens que são relevantes para você e que exigem ação. Ao identificar problemas de alta prioridade, você pode usar o Systems Manager OpsCenter para executar runbooks do Automatio n e resolver rapidamente esses problemas. Para obter informações, consulte, AWS Systems ManagerExplorer.	18 de novembro de 2019

> **Note**
>
> A configuração do OpsCenter do Systems Manager é integrada à configuração do Explorer. Se você já tiver configurado o OpsCenter, ainda precisará concluir a configuração integrada para verificar as definições e opções. Se você não tiver configurado o OpsCenter, poderá usar a configuração integrada para começar a usar ambas as ferramentas. Para obter mais informações, consulte Conceitos básicos sobre o Explorer e o OpsCenter.

Recursos aprimorados de pesquisa de parâmetros	As ferramentas de pesquisa de parâmetros agora facilitam a localização de um parâmetro quando você tem um grande número deles em sua conta ou quando não se lembra do nome exato de um parâmetro . Com a ferramenta de pesquisa, é possível filtrar por `contains`. Anteriorm ente, as ferramentas de pesquisa permitiam a localizaç ão de nomes de parâmetro somente por `equals` e `begins-with` . Para obter mais informações, consulte Pesquisar parâmetros do Systems Manager.	15 de novembro de 2019

Novo criador de documentos para automação | suporte com base no console para execução de scripts em etapas de automação	Agora você pode usar o Systems Manager Automation para criar e compartilhar manuais operacionais padronizados para garantir a consistência entre usuários, Contas da AWS e Regiões da AWS. Com essa capacidade de executar scripts e adicionar documentação em linha aos seus runbooks de automação usando o Markdown, você pode reduzir erros e eliminar etapas manuais, como navegar por procedimentos escritos em wikis e executar comandos de terminal. Para obter mais informações, consulte os tópicos a seguir. • Demonstração: usar o Document Builder para criar um runbook de automação • `aws:executeScript` (Referência de ações de automação) • Criar runbooks de automação usando o Document Builder • Novos recursos de automação no Systems Manager no Blog de notícias da AWS	14 de novembro de 2019

Executar uma atualização de pacote local usando o Distributor	Anteriormente, quando você queria instalar uma atualização em um pacote usando o Distributor, a única opção era desinstalar o pacote inteiro e reinstalar a nova versão. Agora você pode optar por executar uma atualização local. Durante uma atualização local, o Distributor instala somente arquivos novos ou alterados desde a última instalação, de acordo com o script de atualização incluído no pacote. Com essa opção, seu aplicativo de pacote pode permanecer disponível e não ser colocado offline durante a atualização. Para obter mais informações, consulte os tópicos a seguir. • Criar um pacote • Instalar ou atualizar pacotes	11 de novembro de 2019

Novo recurso de atualização automática do SSM Agent	Com um clique, você pode configurar todas as instâncias em sua Conta da AWS para verificar e baixar automaticamente novas versões do SSM Agent. Para fazer isso, escolha Agent auto update (Atualização automática do agente) na página Managed instances (Instâncias gerenciadas) no console do AWS Systems Manager. Para obter informações, consulte Automatizar atualizações para o SSM Agent.	5 de novembro de 2019
Restringir o acesso ao Session Manager usando tags fornecidas pela AWS	Agora está disponível um segundo método para controlar o acesso do usuário às ações de sessão. Com esse novo método, é possível criar políticas de acesso do IAM usando etiquetas de sessão fornecidas pela AWS em vez de usar a variável `{aws:username}` . O uso das tags de sessão fornecidas pela AWS possibilita que as organizações que usam IDs federados controlem o acesso dos usuários às sessões. Para obter informações, consulte Permitir que um usuário encerre somente sessões iniciadas por ele.	2 de outubro de 2019

Novo documento de comandos do SSM para aplicar manuais do Ansible	É possível criar associações do State Manager que executam manuais do Ansible usando o documento `AWS-ApplyAnsiblePlaybooks`. Esse documento oferece os seguintes benefícios para executar manuais: • Suporte à execução de manuais complexos • Support ao download de playbooks do GitHub e do Amazon Simple Storage Service (Amazon S3) • Suporte à estrutura de manual compactada • Registro em log aprimorado • Capacidade de especificar qual manual executar quando os manuais estiverem empacotados Para obter mais informações, consulte Criar associações que executam manuais do Ansible.	24 de setembro de 2019

Suporte ao encaminhamento de portas para o Session Manager	O Session Manager agora oferece suporte a sessões de encaminhamento de portas. O encaminhamento de portas permite que você crie túneis com segurança entre as instâncias implantadas em sub-redes privadas, sem a necessidade de iniciar o serviço SSH no servidor, para abrir a porta SSH no grupo de segurança ou usar um bastion host. Semelhante aos túneis SSH, o encaminhamento de portas permite encaminhar o tráfego entre o laptop para abrir portas na instância. Assim que o encaminhamento de portas estiver configurado, você pode se conectar à porta local e acessar o aplicativ o do servidor em execução dentro da instância. Para obter mais informações, consulte os tópicos a seguir. • Encaminhamento de portas usando o AWS Systems ManagerSession Manager no Blog de notícias da AWS • Iniciar uma sessão (encaminhamento de portas)	29 de agosto de 2019

Especificar um nível de parâmetro padrão ou automatizar a seleção de níveis	Agora você pode especificar um nível de parâmetro padrão a ser usado para solicitações de criação ou atualização de um parâmetro que não especificam um nível. Você pode definir o nível padrão como parâmetros padrão, parâmetros avançados ou uma nova opção, Intelligent-Tiering. O Intelligent-Tiering avalia cada solicitação PutParameter e cria um parâmetro avançado somente quando necessário. (Os parâmetros avançados serão necessários se o tamanho do valor do parâmetro for superior a 4 KB, se uma política de parâmetro estiver associada ao parâmetro ou se o número máximo de 10.000 parâmetros compatíveis com o nível padrão já estiverem criados.) Para obter mais informações sobre como especificar um nível padrão e usar o Intelligent-Tiering, consulte Especificar um nível de parâmetro padrão.	27 de agosto de 2019

A seção Trabalhar com associações foi atualizada com os procedimentos da CLI e do PowerShell	A seção Como trabalhar com associações foi atualizada para incluir a documentação processual para o gerenciamento de associações usando a AWS CLI ou o Ferramentas da AWS para PowerShell. Para obter informações, consulte Trabalhar com associações no Systems Manager.	26 de agosto de 2019
A seção Como trabalhar com execuções de automação foi atualizada com os procedimentos da CLI e do PowerShell	A seção Como trabalhar com associações foi atualizada para incluir a documentação processual para executar fluxos de trabalho de Automação usando a AWS CLI ou o Ferramentas da AWS para PowerShell. Para obter informações, consulte Trabalhar com execuções de automação.	20 de agosto de 2019

O OpsCenter integra-se ao Application Insights	O OpsCenter se integra ao Amazon CloudWatch Application Insights para .NET e SQL Server. Isso significa que você pode criar automaticamente OpsItems para problemas detectados em seus aplicativos. Para obter informações sobre como configurar o Application Insights para criar o OpsItems, consulte Instalar, configurar e gerenciar a aplicação para monitoramento, no Guia do usuário do Amazon CloudWatch.	7 de agosto de 2019

Novo recurso do console: AWS Systems Manager Quick Setup	O Quick Setup é um novo recurso no console do Systems Manager que ajuda a configurar rapidamente vários componentes do Systems Manager nas instâncias do EC2. Especificamente, a Configuração rápida ajuda você a configurar os seguintes componentes nas instância s escolhidas ou de destino usando tags: • Uma função do perfil da instância AWS Identity and Access Management (IAM) para Systems Manager. • Uma atualização bimensal programada do SSM Agent. • Uma coleção programada de metadados de inventário a cada 30 minutos. • Uma verificação diária de suas instâncias para identificar patches ausentes. • Uma instalação e uma configuração únicas do agente do Amazon CloudWatch. • Uma atualização mensal programada do agente do CloudWatch.	7 de agosto de 2019

Para obter mais informações, consulte Configuração rápida do AWS Systems Manager.

Registrar um grupo de recursos como um destino de janela de manutenção	Além de registrar instâncias gerenciadas como destino de uma janela de manutenção, agora você pode registrar um grupo de recursos como destino da janela de manutenção. O Maintenance Windows oferece suporte a todos os tipos de recursos da AWS que forem compatíveis com o AWS Resource Groups, incluindo `AWS::EC2::Instance`, `AWS::DynamoDB::Table`, `AWS::OpsWorks::Instance`, `AWS::Redshift::Cluster` e muito mais. Com esta versão, você também pode enviar comandos a um grupo de recursos, por exemplo, usando o console do Run Command ou o comando send-command da AWS CLI. Para obter mais informações, consulte os tópicos a seguir. • Atribuir destinos a uma janela de manutenção (console) • Exemplos: Registrar destinos em uma janela de manutenção • Usar destinos e controles de taxa para enviar comandos para uma frota	23 de julho de 2019

Criação e versionamento simplificados de pacotes com o Distributor do AWS Systems Manager	O Distributor tem um novo fluxo de criação simplificado de pacotes, que pode gerar manifesto, scripts e hashes de arquivo de um pacote para você. Você também pode usar o fluxo de trabalho simplificado ao adicionar uma versão a um pacote existente.	22 de julho de 2019
Novo painel de categorias de documentos para a automação do Systems Manager	O Systems Manager inclui um novo painel de categorias de documentos quando você executa uma automação no console. Use esse painel para filtrar runbooks de automação com base em sua finalidade.	18 de julho de 2019
Suporte para iniciar sessões do Session Manager usando as credenciais do usuário do sistema operacional	Por padrão, as sessões do Session Manager são executadas usando as credenciais de uma conta `ssm-user` gerada pelo sistema criada em uma instância gerenciada. Em máquinas do Linux, agora você pode, em vez disso, iniciar sessões usando as credenciais de uma conta do sistema operacional. Para obter informações, consulte Ativar a opção “Run as support” (Executar como suporte) para instâncias do Linux.	9 de julho de 2019

Suporte para iniciar sessões do Session Manager usando SSH	Agora, você pode usar a AWS CLI para iniciar uma sessão SSH em uma instância gerenciada usando o Session Manager. Para obter informações sobre permitir sessões SSH com o Session Manager, consulte (Opcional) Ativar sessões SSH do Session Manager. Para obter informações sobre como iniciar uma sessão SSH usando Session Manager, consulte Como iniciar uma sessão (SSH).	9 de julho de 2019
Suporte para alteração de senhas em instâncias gerenciadas	Agora, você pode redefinir senhas em máquinas que gerenciem usando o Systems Manager (instâncias gerenciadas). Você pode redefinir a senha usando o console do Systems Manager ou a AWS CLI. Para obter informações, consulte Redefinir senhas em instâncias gerenciadas.	9 de julho de 2019

Revisões para "O que é o AWS Systems Manager?"	O conteúdo introdutório em O que é o AWS Systems Manager? foi expandido para oferecer uma introduçã o mais abrangente para o serviço e refletir as ferrament as do Systems Manager que foram lançadas recentemente. Além disso, outros conteúdos na seção foram movidos para tópicos individuais para melhorar a capacidade de descoberta.	10 de junho de 2019

Nova ferramenta do Systems Manager: OpsCenter	O OpsCenter fornece um local central onde engenheiros de operações e profissionais de TI podem visualizar, investigar e resolver itens de trabalho operacionais (OpsItems) relacionados a recursos da AWS. O OpsCenter foi projetado para reduzir o tempo médio de resolução de problemas que afetam os recursos da AWS. Essa ferramenta do Systems Manager agrega e padroniza o OpsItems em todos os serviços enquanto fornece dados de investigação contextuais sobre cada OpsItem, OpsItems relacionados e recursos relacionados. O OpsCenter também fornece runbooks do Systems Manager Automation que você pode usar para resolver problemas rapidamente. Você pode especificar dados personalizados e pesquisáveis para cada OpsItem. Você também pode visualizar relatórios de resumo gerados automaticamente sobre o OpsItems por status e origem. Para obter mais informações, consulte AWS Systems ManagerOpsCenter.	6 de junho de 2019

Alterações no painel de navegação esquerdo do Systems Manager no AWS Management Console	O painel de navegação esquerdo do Systems Manager no AWS Management Console inclui novos cabeçalhos, entre eles um novo cabeçalho para Ops Center, que oferecem um agrupamento mais lógico das ferramentas do Systems Manager.	6 de junho de 2019

Tutorial revisado para a criação e a configuração de uma janela de manutenção usando a AWS CLI	O Tutorial: Criar e configurar uma janela de manutenção (AWS CLI) foi reorganizado para fornecer um caminho simples por meio de etapas práticas. Crie uma única janela de manutenção, identifique um único destino e configure uma tarefa simples para a execução da janela de manutenção. Ao longo do caminho, fornecemos informações e exemplos que podem ser usados para criar seus próprios comandos de registro de tarefas, incluindo informações para usar pseudoparâmetros, como `{{TARGET_ID}}` . Para obter informações adicionais e exemplos, veja estes tópicos: • Exemplos: Registrar destinos em uma janela de manutenção • Exemplos: Registrar tarefas em uma janela de manutenção • Sobre as opções de register-task-with-maintenance-windows • Usar pseudoparâmetros ao registrar tarefas da janela de manutenção	31 de maio de 2019

Notificações sobre atualizações do SSM Agent	Para receber notificações sobre atualizações do SSM Agent, inscreva-se na página Notas de versão do SSM Agent no GitHub.	24 de maio de 2019
Receber notificações ou acionar ações com base em alterações no Parameter Store	O tópico Configurar notificações ou acionar ações com base em eventos do Parameter Store agora ajuda você a configurar regras do Amazon EventBridge para responder a alterações no Parameter Store. Você pode receber notificações ou acionar outras ações quando qualquer um destes itens ocorrer: • Um parâmetro é criado, atualizado ou excluído. • A versão de um rótulo de parâmetro é criada, atualizada ou excluída. • Um parâmetro expira, vai expirar, ou não foi alterado em um período de tempo especificado.	22 de maio de 2019

Principais revisões do conteúdo de configuração e conceitos básicos	Nós expandimos e reorganizamos o conteúdo de Configuração e Conceitos básicos no Guia do usuário do AWS Systems Manager. O conteúdo de Configuração foi dividido em duas seções. Uma seção se concentra nas tarefas para definir o Systems Manager para configurar e gerenciar as instâncias do EC2. A outra se concentra nas tarefas para definir o Systems Manager para configurar e gerenciar os servidores on-premises e as máquinas virtuais (VMs) em um ambiente híbrido. Agora ambas as seções apresentam todos os tópicos de configuração como etapas principais enumeradas na ordem de conclusão recomendada. Um novo capítulo Getting Started (Conceitos básicos) se concentra em como ajudar os usuários finais a começarem a usar o Systems Manager após a conclusão das tarefas de configuração da conta e do serviço. • Configurar o AWS Systems Manager	15 de maio de 2019

- Configurar o AWS Systems Manager para ambientes híbridos
- Conceitos básicos da AWS Systems Manager

Incluir patches para aplicações da Microsoft em listas de referência de patches (Windows)	O Patch Manager agora é compatível com atualizações de patch para aplicações da Microsoft em instâncias do Windows Server. Anteriormente, apenas patches para o sistema operacional Windows Server eram compatíveis. O Patch Manager fornece duas listas de referência de patches predefinidas para instâncias do Windows Server. A lista de referência do patch `AWS-WindowsPredefinedPatchBaseline-OS` aplica-se somente a patches de sistema operacional. O `AWS-WindowsPredefinedPatchBaseline-OS-Applications` aplica-se ao sistema operacional Windows Server e a aplicações no Microsoft Windows. Para obter informações sobre como criar uma lista de referência de patches personalizada que inclua patches para aplicações da Microsoft, consulte o primeiro procedimento em Criar uma linha de base de patches personalizada. Além disso, como parte dessa atualização, os nomes das listas de referência de patches predefinidas fornecidas pela AWS estão sendo alterados	7 de maio de 2019

	. Para obter mais informaçõ es, consulte Linhas de base predefinidas.	
Exemplos para registrar destinos da janela de manutenção usando a AWS CLI	O novo tópico Exemplos: registrar destinos com uma janela de manutenção fornece três exemplos de comandos para demonstrar diferentes maneiras de especificar os destinos de uma janela de manutenção quando você usa a AWS CLI. O tópico também explica o melhor caso de uso para cada um dos comandos de exemplo.	3 de maio de 2019

Atualizações em tópicos de grupos de patches	O tópico Sobre grupos de patches foi atualizado para incluir uma seção sobre como as instâncias gerenciadas determinam a linha de base de patches apropriada para uso durante operações de aplicação de patch. Além disso, houve a adição de instruções para usar a AWS CLI ou o console do Systems Manager para adicionar as tags Patch Group ou PatchGroup às suas instâncias gerenciadas e como adicionar Patch Group ou PatchGroup a uma lista de referência de patches. (Você deve usar **`PatchGroup`** , sem espaço, se tiver permissão para tags nos metadados da instância do EC2.) Para obter mais informações, consulte Criar um grupo de patches e Adicionar um grupo de patches a uma linha de base de patches.	1º de maio de 2019

Novos recursos do Parameter Store	O Parameter Store oferece os seguintes novos recursos: • Parâmetros avançados: agora, o Parameter Store permite que você configure individualmente parâmetros para usar um nível de parâmetros padrão (o nível padrão) ou um nível de parâmetros avançados. Parâmetros avançados oferecem uma cota de tamanho maior para o valor do parâmetro, uma cota superior para o número de parâmetros que você pode criar por Conta da AWS e Região da AWS, bem como a capacidade de usar políticas de parâmetros. Para obter mais informações sobre parâmetros avançados, consulte Sobre parâmetros avançados do Systems Manager. • Políticas de parâmetros: políticas de parâmetros ajudam a gerenciar um conjunto crescente de parâmetros, permitindo atribuir critérios específicos a um parâmetro, como uma data de validade ou vida útil. Políticas de parâmetro	25 de abril de 2019

s são especialmente úteis para forçar você a atualizar ou excluir senhas e dados de configuração armazenados no Parameter Store. Políticas de parâmetros estão disponíveis apenas para parâmetros que usam o nível de parâmetros avançados. Para obter mais informações, consulte Trabalhar com políticas de parâmetros.

- Maior throughput: agora, você pode aumentar a cota da throughput do Parameter Store para um máximo de 1.000 transações por segundo. Para obter mais informações, consulte Aumentar a throughput do Parameter Store.

Atualizações para na seção Automação	A seção Automação foi atualizada para melhorar a performance de descoberta. Além disso, três novos tópicos foram adicionados à seção Automação: • Executar uma automação passo a passo • Executar uma automação que exija aprovações • Programação de automações com associações do State Manager	17 de abril de 2019

Criptografe dados da sessão usando uma chave do AWS KMS	Por padrão, o Session Manager usa o TLS 1.2 para criptografar dados de sessão transmitidos entre as máquinas locais de usuários na sua conta e suas instâncias do EC2. Agora, você pode optar por criptografar esses dados usando uma AWS KMS key criada no AWS Key Management Service. Você pode usar uma chave criada na sua Conta da AWS ou uma que tenha sido compartilhada com você por outra conta. Para obter informações sobre como especificar uma KMS para criptografar dados de sessão, consulte Ativar a criptografia de chaves de sessão do AWS KMS (console), Criar preferências do Session Manager (AWS CLI) ou Atualizar preferências do Session Manager (AWS CLI).	4 de abril de 2019

Configurar notificações do Amazon SNS para o AWS Systems Manager	Adição de instruções para usar a AWS CLI ou o console do Systems Manager para configurar notificações do Amazon SNS para o Run Command e as tarefas do Run Command registradas em uma janela de manutenção. Para obter mais informações, consulte Configurar notificações do Amazon SNS para o AWS Systems Manager.	6 de março de 2019

Instâncias avançadas para servidores e VMs em ambientes híbridos	O AWS Systems Manager oferece um nível de instâncias padrão e um nível de instâncias avançadas para servidores e VMs no seu ambiente híbrido. O nível de instâncias padrão permite registrar no máximo 1.000 VMs ou servidores por Conta da AWS por Região da AWS. Se precisar registrar mais de 1.000 servidores ou VMs em uma única conta e região, use o nível de instâncias avançadas. Você pode criar quantas instâncias quiser no nível de instâncias avançadas, mas todas as instâncias configuradas para o Systems Manager estão disponíveis mediante pagamento conforme o uso. Instâncias avançadas também permitem que você se conecte às suas máquinas híbridas usando o AWS Systems Manager Session Manager. O Session Manager fornece acesso via shell interativo às suas instâncias. Para obter mais informações sobre como permitir instâncias avançadas, consulte Usar o nível de instâncias avançadas.	4 de março de 2019

Crie associações do State Manager que usem documentos do SSM compartilhados	Você pode criar associações do State Manager que usam documentos de Comando e Automação de runbooks do SSM compartilhados de outras Contas da AWS. Criar associações usando documentos compartilhados do SSM ajuda a manter o Amazon EC2 e sua infraestrutura híbrida em um estado consistente, mesmo quando as instâncias não estiverem na mesma conta. Para obter mais informações sobre o compartilhamento de documentos do SSM, consulte Documentos do AWS Systems Manager. Para obter informações sobre como criar uma associação do State Manager, consulte Criar uma associação.	28 de fevereiro de 2019
Visualizar listas de eventos do Systems Manager compatíveis com as regras do Amazon EventBridge	O novo tópico Monitorar eventos do Systems Manager com o Amazon EventBridge fornece um resumo dos vários eventos emitidos pelo Systems Manager para os quais você pode configurar regras de monitoramento de eventos no EventBridge.	25 de fevereiro de 2019

Adicionar tags ao criar recursos do Systems Manager.	Agora, o Systems Manager é compatível com a capacidade de adicionar etiquetas a determinados tipos de recurso quando você os cria. Os recursos que você pode marcar ao criá-los com a AWS CLI ou um SDK incluem janelas de manutenção, linhas de base de patch, parâmetros do Parameter Store e documentos do SSM. Você também pode atribuir tags a uma instância gerenciada ao criar uma ativação para ela. Ao usar o console do Systems Manager, você pode adicionar tags a janelas de manutenção, linhas de base de patches e parâmetros.	24 de fevereiro de 2019

Criação automática da função do IAM para o Systems Manager Inventory	Anteriormente, era necessário criar uma função do AWS Identity and Access Management (IAM) e anexar políticas separadas a essa função para visualizar dados de inventário na página Inventory Detail View (Exibição de detalhes do inventário) no console. Não é mais necessário criar essa função ou anexar políticas a ela. Quando você escolhe um Remote Data Sync na página Inventory Detail View (Visualização de detalhes do inventário), o Systems Manager cria automaticamente a função `Amazon-GlueServicePolicyForSSM` e atribui a política Amazon-GlueServicePolicyForSSM-{S3 bucket name} e a política AWSGlueServiceRole a ela. Para obter mais informações, consulte Consultar dados de inventário de várias regiões e contas.	14 de fevereiro de 2019

Demonstrações do Maintenance Windows para atualizar o SSM Agent	Duas novas demonstrações adicionadas à documentação do Maintenance Windows. Essas demonstrações detalham como usar o console do Systems Manager ou a AWS CLI para criar uma janela de manutenção que mantém o SSM Agent atualizado automaticamente. Para obter mais informações, consulte Demonstrações de Maintenance Windows.	11 de fevereiro de 2019
Usar parâmetros públicos do Parameter Store	Adição da uma seção rápida que descreve parâmetros públicos do Parameter Store. Para obter mais informaçõ es, consulte Usar parâmetros públicos do Systems Manager.	31 de janeiro de 2019
Usar a AWS CLI para criar preferências do Session Manager	Adição de instruções para usar a AWS CLI para criar preferências do Session Manager, como CloudWatc h Logs, opções de registro em log de bucket do S3 e configurações de criptografia da sessão. Para obter mais informações, consulte Usar a AWS CLI para criar preferênc ias do Session Manager.	22 de janeiro de 2019

Executar fluxos de trabalho do Systems Manager Automation usando o State Manager	Agora, o AWS Systems Manager State Manager é compatível com a criação de associações que usam runbooks do SSM Automation. Anteriormente, o State Manager apenas era compatível com documentos `command` e `policy`, ou seja, somente era possível criar associações direcionadas a instâncias gerenciadas. Com o suporte para runbooks de automação do SSM, agora é possível criar associações direcionadas a diferentes tipos de recursos da AWS. Para obter mais informações, consulte Executar os fluxos de trabalho do Systems Manager Automation usando o State Manager.	22 de janeiro de 2019

Atualizações de referência para expressões de cron e rate e opções de programação da janela de manutenção	O tópico de referência Expressões cron e rate para o Systems Manager foi revisado. A nova versão fornece mais exemplos e explicações melhores de como usar expressões cron e rate para programar as associações da janela de manutenção e do State Manager. Além disso, o novo tópico Programação do Maintenance Windows e opções de período ativo explica como as várias opções relacionadas à programação das janelas de manutenção (Data de início, Data de término, Fuso horário, Frequência de programação) se relacionam entre si.	6 de dezembro de 2018
Ativar o registro da depuração do SSM Agent	Você pode habilitar o log de depuração do SSM Agent editando o arquivo seelog.xml.template na instância gerenciada. Para obter mais informações, consulte Ativar o log de depuração do SSM Agent.	30 de novembro de 2018

Suporte para arquiteturas de processador ARM64	O AWS Systems Manager agora oferece suporte a versões ARM64 dos sistemas operacionais Amazon Linux 2, Red Hat Enterprise Linux 7.6 e Ubuntu Server (18.04 LTS e 16.04 LTS). Para obter mais informações, consulte as instruções de instalação do Amazon Linux 2, RHEL e Ubuntu Server 18.04 e 16.04 LTS com pacotes Snap. Para obter mais informações sobre o tipo de instância A1, consulte Instâncias de uso geral no Guia do usuário do Amazon EC2.	26 de novembro de 2018

Criar e implantar pacotes usando o AWS Systems Manager Distributor	Usando o AWS Systems Manager Distributor, você empacota seu próprio software ou encontra pacotes de software de agente fornecidos pela AWS, como o `AmazonCloudWatchAgent` para instalar em instâncias gerenciadas do AWS Systems Manager. O Distributor publica recursos, como pacotes de software, para instâncias gerenciadas do AWS Systems Manager. A publicação de um pacote anuncia versões específicas do documento do pacote: um documento do Systems Manager que você cria quando adiciona o pacote no Distributor para instâncias gerenciadas, identificado por IDs de instância gerenciada, IDs da Conta da AWS, tags ou de uma Região da AWS. Para obter mais informações, consulte AWS Systems ManagerDistributor.	20 de novembro de 2018

Simultaneamente, execute os fluxos de trabalho de automação do AWS Systems Manager em várias Regiões da AWS e Contas da AWS em uma conta central	Você pode executar simultaneamente fluxos de trabalho de automação do AWS Systems Manager em várias Regiões da AWS e Contas da AWS ou em unidades organizacionais da AWS (UOs) em uma conta de gerenciamento de automação. Executar simultaneamente Automações em várias regiões e contas ou UOs reduz o tempo necessário para administrar seus recursos da AWS, ao mesmo tempo em que melhora a segurança do seu ambiente de computação. Para obter mais informações, consulte Executar fluxos de trabalho do Automation em várias Regiões da AWS e Contas da AWS.	19 de novembro de 2018

Consultar dados de inventário de várias Regiões da AWS e Contas da AWS	O Systems Manager Inventory integra-se ao Amazon Athena para ajudar você a consultar dados de inventário de várias Regiões da AWS e Contas da AWS. A integração com o Athena usa a sincronização de dados dos recursos para que você possa visualizar os dados do inventário de todas as instâncias gerenciadas na página Inventory Detailed View (Visualização detalhada do inventário) no console do AWS Systems Manager. Para obter mais informações, consulte Consultar dados de inventário de várias regiões e contas.	15 de novembro de 2018

Criar associações do State Manager que executam arquivos MOF	É possível executar arquivos Managed Object Format (MOF) para impor um estado de destino em instâncias gerenciadas do Windows Server com o State Manager ao usar o documento do SSM, `AWS-ApplyDSCMofs` . O documento `AWS-ApplyDSCMofs` tem dois modos de execução: Com o primeiro modo, é possível configurar a associação para verificar e relatar se as instâncias gerenciadas estão atualmente no estado de destino definido nos arquivos MOF especificados. No segundo modo, você pode executar os arquivos MOF e alterar a configuração de suas instâncias com base nos recursos e seus valores definida nos arquivos MOF. O `AWS-ApplyDSCMofs` permite que você faça download e execute arquivos de configuração MOF no Amazon Simple Storage Service (Amazon S3), em um compartilhamento local ou em um site seguro com um domínio HTTPS. Para obter mais informações, consulte Criar associações que executam arquivos MOF.	15 de novembro de 2018

Restringir o acesso administrativo em sessões do Session Manager	As sessões do Session Manager são executadas usando as credenciais de uma conta de usuário que é criada com permissões de administrador ou de raiz padrão chamadas `ssm-user`. As informações sobre como restringir o controle administrativo dessa conta agora estão disponíveis no tópico Ativar ou desativar as permissões administrativas da conta do ssm-user.	13 de novembro de 2018
Referência de exemplos de YAML em ações de automação	A Referência de ações de automação agora inclui um exemplo de YAML para cada ação que já inclui um exemplo JSON.	31 de outubro de 2018

Atribuir níveis de gravidade de conformidade a associações	Agora você pode atribuir níveis de gravidade da conformidade a associaçõ es do State Manager. Esses níveis de gravidade são relatados no painel de conformidade e também podem ser usados para filtrar seus relatórios de conformid ade. Os níveis de gravidade que você pode atribuir incluem Crítico, Alto, Médio, Baixo e Não especificado. Para obter mais informações, consulte Criar uma associação (console).	26 de outubro de 2018
Usar destinos e controles de taxa com automação e State Manager	Controle a execução de Automações e associações do State Manager em sua frota de recursos usando destinos, simultaneidade e limites de erro. Para obter mais informações, consulte Usar destinos e controles de taxa para executar fluxos de trabalho de automação em uma frota e Usar destinos e controles de taxa com associações do State Manager.	23 de outubro de 2018

Especificar os períodos ativos e fusos horários internacionais para janelas de manutenção	Você também pode especificar datas anteriores e posteriores (data de início e data de término) em que uma janela de manutenção não deverá ser executada, bem como o fuso horário internacional no qual basear a programação da janela de manutenção. Para obter mais informações, consulte Criar uma janela de manutenção (console) e Atualizar uma janela de manutenção (AWS CLI).	9 de outubro de 2018
Manter uma lista personalizada de patches para sua linha de base de patch em um bucket do S3	Com o novo parâmetro “InstallOverrideList” no documento de comando do SSM `AWS-RunPatchBaseline`, você pode especificar um URL de estilo caminho do Amazon Simple Storage Service (Amazon S3) para uma lista de patches a serem instalados. Esta lista de instalação de patches mantida em um bucket do S3 no formato YAML substitui os patches especificados pela linha de base de patch padrão. Para obter mais informações, consulte Nome do parâmetro: InstallOverrideList.	5 de outubro de 2018

Mais controle sobre se as dependências de patches são instaladas	Anteriormente, se um patch em sua lista de patches rejeitados fosse identificado como uma dependência de outro patch, continuaria a ser instalado. Agora, você pode escolher se deseja instalar essas dependências ou impedir que sejam instalada s. Para obter mais informaçõ es, consulte Criar uma linha de base de patches.	5 de outubro de 2018
Criar fluxos de trabalho de automação dinâmicos com ramificações condicionais	A ação de automação `aws:branch` permite que você crie um fluxo de trabalho de Automação dinâmico que avalia várias opções em uma única etapa e, em seguida, salta para outra etapa no runbook de automação com base nos resultados da avaliação. Para obter mais informações, consulte Uso de instruções condicionais em runbooks.	26 de setembro de 2018

Usar a AWS CLI para atualizar as preferências do Session Manager	As instruções para usar a CLI para atualizar as preferências do Session Manager, como opções de log do CloudWatch Logs e de buckets do S3, foram adicionadas ao Guia do usuário do AWS Systems Manager. Para obter informações, consulte Usar a AWS CLI para atualizar as preferências do Session Manager.	25 de setembro de 2018
Requisito do SSM Agent atualizado para Session Manager	O Session Manager agora requer o SSM Agent versão 2.3.68.0 ou posterior. Para obter mais informações sobre pré-requisitos do Session Manager, consulte Pré-requisitos completos do Session Manager.	17 de setembro de 2018

Gerenciar instâncias sem abrir portas de entrada ou manter bastion hosts usando Session Manager	Usando o Session Manager, uma ferramenta do AWS Systems Manager totalment e gerenciada, é possível gerenciar suas instância s do EC2 por meio de um shell baseado em navegador interativo de um clique por meio da AWS CLI. O Session Manager fornece gerenciam ento de instâncias seguro e auditável sem a necessida de de abrir portas de entrada, manter bastion hosts ou gerenciar chaves SSH. O Session Manager também facilita a conformidade com políticas corporativas que exigem acesso controlad o a instâncias, práticas rígidas de segurança e logs totalmente auditáveis com detalhes de acesso a instância s, sem deixar de fornecer aos usuários finais acesso simples de um clique a suas instâncias do EC2. Para obter mais informações, consulte Saiba mais sobre o Session Manager.	11 de setembro de 2018

Invocar outros Serviços da AWS em um fluxo de trabalho do Systems Manager Automation	Você pode invocar outros Serviços da AWS e outras ferramentas do Systems Manager em seu fluxo de trabalho do Automation, usando três novas ações do Automation (ou plug-ins) em seus runbooks. Para obter mais informações, consulte Uso de saídas de ações como entradas.	28 de agosto de 2018
Usar chaves de condição específicas do Systems Manager em políticas do IAM	O tópico Especificar condições em uma política foi atualizado para listar as chaves de condição do IAM para o Systems Manager que você pode incorporar em políticas. Você pode usar essas chaves para especificar as condições sob as quais uma política deve entrar em vigor. O tópico também inclui links para políticas de exemplo e outros tópicos relacionados.	18 de agosto de 2018

Agregar dados do inventário com grupos para ver quais instâncias estão e quais não estão configuradas para coletar um tipo de inventário	Os grupos permitem que você veja rapidamente uma contagem de quais instâncias gerenciadas estão e quais não estão configuradas para coletar um ou mais tipos de inventário. Com grupos, você especifica um ou mais tipos de inventário e um filtro que usa o operador `exists`. Para obter mais informações, consulte Agregar dados de inventário.	16 de agosto de 2018
Exibir o histórico e o controle de alterações do inventário e da conformidade de configuração	Agora, você pode visualizar o histórico e o controle de alterações do inventário coletados em suas instâncias gerenciadas. Você também pode visualizar o histórico e o controle de alterações da aplicação de patches do Patch Manager e as associações do State Manager relatadas pela conformidade da configuração. Para obter mais informações, consulte Visualizar o histórico e o controle de alterações do inventário.	9 de agosto de 2018

O Parameter Store integra-se ao Secrets Manager	O Parameter Store agora está integrado ao AWS Secrets Manager para que você possa recuperar segredos do Secrets Manager ao usar outros Serviços da AWS que já oferecem suporte a referênci as a parâmetros do Parameter Store. Esses serviços incluem o Amazon EC2, Amazon Elastic Container Service, AWS Lambda, AWS CloudFormation, AWS CodeBuild, AWS CodeDeplo y e outras ferramentas do Systems Manager. Usando o Parameter Store para referenciar segredos do Secrets Manager, você cria um processo consistente e seguro para chamar e usar segredos e referenciar dados no código e nos scripts de configuração. Para obter informações, consulte Fazer referência a segredos do AWS Secrets Manager em parâmetros do Parameter Store.	26 de julho de 2018

Anexe rótulos aos parâmetros do Parameter Store	Um rótulo de parâmetro é um alias definido pelo usuário para ajudar você a gerenciar diferentes versões de um parâmetro. Quando você modifica um parâmetro, o Systems Manager salva automaticamente uma nova versão e incrementa o número da versão em um. Um rótulo pode ajudar você a lembrar-s e do objetivo de uma versão de parâmetro quando houver várias versões. Para obter informações, consulte Rotular parâmetros.	26 de julho de 2018

Criar fluxos de trabalho de automação dinâmicos	Por padrão, as etapas (ou ações) que você define na seção mainSteps de um runbook de automação são executadas em ordem sequencial. Depois que uma ação é concluída, a próxima ação especificada na seção mainSteps começa. Com essa versão, agora é possível criar fluxos de trabalho de automação que executam ramificações condicionais. Isso significa que você pode criar fluxos de trabalho de automação que respondem dinamicamente a alterações nas condições e saltam para determinada etapa. Para obter informações, consulte Uso de instruções condicionais em runbooks.	18 de julho de 2018

O SSM Agent agora é pré-instalado no Ubuntu Server 16.04 AMIs usando Snap	A partir das instâncias criadas com Ubuntu Server 16.04 AMIs identificadas com `20180627`, o SSM Agent é pré-instalado usando pacotes Snap. Em instâncias criadas por AMIs anteriores, você deve continuar usando pacotes do instalador deb. Para obter informações, consulte Sobre instalações do SSM Agent em instâncias do Ubuntu Server 16.04 de 64 bits.	7 de julho de 2018
Revisar as permissões mínimas do S3 exigidas pelo SSM Agent	O novo tópico Permissões mínimas do bucket do S3 para o SSM Agent fornece as informações sobre os buckets do Amazon Simple Storage Service (Amazon S3) que os recursos podem precisar acessar para executar operações do Systems Manager. Você pode especific ar esses buckets em uma política personalizada, se quiser limitar o acesso ao bucket do S3 para um perfil de instância ou um endpoint da VPC para o mínimo necessári o para usar o Systems Manager.	5 de julho de 2018

Visualizar o histórico de execução completo de determinado ID de associação do State Manager	O novo tópico Visualizar históricos de associação descreve como visualizar todas as execuções de um determinado ID de associação e os detalhes da execução de um ou mais recursos.	2 de julho de 2018
O Patch Manager apresenta o suporte para o Amazon Linux 2	Agora, você pode usar o Patch Manager para aplicar patches às instâncias do Amazon Linux 2. Para obter informaçõ es gerais sobre suporte de sistemas operacionais para o Patch Manager, consulte Patch Manager prerequisites (Pré-requisitos do gerenciad or de patches). Para obter informações sobre os pares de chave-valor compatíveis com o Amazon Linux 2 ao definir um filtro de patch, consulte PatchFilter na Referência da API AWS Systems Manager.	26 de junho de 2018
Envie a saída de comando ao Amazon CloudWatch Logs	O novo tópico Configura r o Amazon CloudWatch Logs para o Run Command descreve como enviar a saída do Run Command para o CloudWatch Logs.	18 de junho de 2018

Criar ou excluir rapidamente a sincronização de dados de recursos para inventário usando o AWS CloudFormation	Você também pode usar o AWS CloudFormation para criar ou excluir uma sincronização de dados de recursos para o Systems Manager Inventory. Para usar AWS CloudFormation, adicione o recurso AWS::SSM::ResourceDataSync ao seu modelo de AWS CloudFormation. Para obter mais informações, consulte Trabalhar com modelos do AWS CloudFormation no Manual do usuário do AWS CloudFormation. Você também pode criar manualmente uma sincronização de dados dos recursos para o inventário, como descrito em Creating a resource data sync for Inventory.	11 de junho de 2018

As notificações de atualização do Manual do usuário do AWS Systems Manager já estão disponíveis por meio de RSS	A versão HTML do Guia do usuário do Systems Manager agora oferece suporte a um feed RSS das atualizações que estão documentadas na página Histórico de atualizações da documentação do Systems Manager. O feed RSS inclui atualizações feitas em junho de 2018, e posterior. As atualizações anunciadas anteriormente ainda estão disponíveis na página Histórico de atualizações da documentação do Systems Manager. Use o botão RSS no menu superior do painel para assinar o feed.	6 de junho de 2018
Especificar um código de saída em scripts para reinicializar instâncias gerenciadas	O novo tópico Reinicializar instâncias gerenciadas de scripts descreve como instruir o Systems Manager a reinicializar instâncias gerenciadas especificando um código de saída em scripts que você executa com o Run Command.	3 de junho de 2018

Crie um evento no Amazon EventBridge sempre que o inventário personalizado for excluído	O novo tópico Visualizar ações de exclusão de inventário no EventBridge descreve como configurar o Amazon EventBridge para criar um evento sempre que um usuário excluir o inventário personalizado.	1º de junho de 2018

Atualizações antes de junho de 2018

A tabela a seguir descreve alterações importantes em cada versão do Guia do usuário do AWS Systems Manager antes de junho de 2018.

Alteração	Descrição	Data de lançamento
Inventário de todas as instâncias gerenciadas na sua Conta da AWS	Você pode criar facilmente um inventário de todas as instâncias gerenciadas na sua Conta da AWS por meio de uma associação de inventário global. Para obter mais informações, consulte Inventário de todos os nós gerenciados na sua Conta da AWS. **Note** As associações globais de inventário estão disponíveis no SSM Agent versão 2.0.790.0 ou posterior. Para obter informações sobre como atualizar o SSM Agent nas suas instâncias, consulte Atualização do SSM Agent por meio de Run Command.	3 de maio de 2018
SSM Agent instalado por	O SSM Agent é instalado, por padrão, nas AMIs 18.04 LTS de 64 bits e 32 bits do Ubuntu Server.	2 de maio de 2018

Alteração	Descrição	Data de lançamento
padrão no Ubuntu Server 18		
Novo tópico	O novo tópico Execução de comandos usando uma versão específica de documento descreve como usar o parâmetro de versão do documento para especificar a versão de um documento do SSM a ser usada quando o comando é executado.	1.º de maio de 2018
Novo tópico	O novo tópico Excluir inventário personalizado descreve como excluir dados de Inventário personalizado do Amazon S3 usando a AWS CLI. O tópico também descreve como usar a opção `SchemaDeleteOption` para gerenciar o inventário personalizado desativando ou excluindo um tipo de inventário personalizado. Este novo recurso usa a operação de API DeleteInventory.	19 de abril de 2018
Notificações do Amazon SNS para SSM Agent	Você pode se inscrever em um tópico do Amazon SNS para receber notificações quando uma nova versão do SSM Agent estiver disponível. Para obter mais informaçõ es, consulte Assinar as notificações do SSM Agent.	9 de abril de 2018
Suporte à aplicação de patches do CentOS	O Systems Manager agora oferece suporte à aplicação de patches em instâncias do CentOS. Para obter informações sobre as versões com suporte do CentOS, consulte Pré-requisitos da Patch Manager. Para obter mais informaçõ es sobre como a aplicação de patches funciona, consulte Como operações do Patch Manager funcionam.	29 de março de 2018
Nova seção	Para fornecer uma única origem de informações de referência no Guia do usuário do AWS Systems Manager, uma nova seção foi introduzida, Referência do AWS Systems Manager. Conteúdo adicional será adicionado a esta seção conforme disponibilizado.	15 de março de 2018

Alteração	Descrição	Data de lançamento
Novo tópico	O novo tópico Sobre formatos de nomes de pacotes para listas de patches aprovados e rejeitados detalha os formatos de nomes de pacotes que você pode inserir nas listas de patches aprovados e rejeitados para uma linha de base de patches personalizados. Os formatos de exemplo são fornecidos para cada tipo de sistema operacional com suporte do Patch Manager.	9 de março de 2018
Novo tópico	O Systems Manager agora se integra ao Chef InSpec. O InSpec é uma estrutura de runtime de código aberto que permite criar perfis legíveis no GitHub ou no Amazon S3. Em seguida, você pode usar o Systems Manager para executar verificações de compatibilidade e visualizar instâncias compatíveis e não compatíveis. Para obter mais informações, consulte Usar os perfis do Chef InSpec com o Systems Manager Compliance.	7 de março de 2018
Novo tópico	O novo tópico Usar perfis vinculados a serviço do Systems Manager descreve como usar uma função vinculada a serviços do AWS Identity and Access Management (IAM) com o Systems Manager. No momento, as funções vinculadas ao serviço são necessárias apenas ao usar o Systems Manager Inventory para coletar metadados sobre tags e grupos de recursos.	27 de fevereiro de 2018

Alteração	Descrição	Data de lançamento
Tópicos novos e atualizados	Agora, você pode usar o Patch Manager para instalar patches que estão em um repositório de origem diferente do padrão configurado na instância. Isso é útil para instalar patches em instâncias com atualizações não relacionadas à segurança, com o conteúdo do Personal Package Archives (PPA) para o Ubuntu Server, com atualizações para aplicativos corporativos internos, etc. Especifique os repositórios de origem de patches alternativos ao criar uma linha de base de patch personalizada. Para obter mais informações, consulte os tópicos a seguir. • Como especificar um repositório de origem de patches alternativo (Linux) • Trabalhando com linhas de base de patch personalizadas • Criar uma linha de base de patch com repositórios personalizados para diferentes versões do SO Além disso, agora você pode usar Patch Manager para corrigir instâncias SUSE Linux Enterprise Server. Patch Manager suporta aplicação de patches SLES 12.* (somente 64 bits). Para obter mais informações, consulte as informações específicas do SLES nos tópicos a seguir: • Como os patches de segurança são selecionados • Como os patches são instalados • Como as regras de linha de base de patch funcionam no SUSE Linux Enterprise Server	6 de fevereiro de 2018

Alteração	Descrição	Data de lançamento
Novo tópico	O novo tópico Documentos do comando do SSM para aplicação de patches em nós gerenciados descreve os sete documentos do SSM disponíveis para ajudar você a manter suas instâncias gerenciadas protegidas com as mais recentes atualizações relacionadas à segurança.	10 de janeiro de 2018
Atualizações importantes sobre o suporte do Linux	Vários tópicos foram atualizados com as seguintes informações: • Por padrão, o SSM Agent é instalado em AMIs básicas do Amazon Linux 1 datadas de 9/2017 e posteriores. • Você deve instalar manualmente o SSM Agent em outras versões do Linux, incluindo imagens que não sejam base, como AMIs otimizadas para o Amazon ECS.	9 de janeiro de 2018
Novo tópico	Um novo tópico, Documento de comando do SSM para a aplicação de patches: `AWS-RunPatchBaseline`, oferece detalhes sobre como este documento do SSM funciona nos sistemas Windows e Linux. Oferece também informações sobre os dois parâmetros disponíveis no documento `AWS-RunPatchBaseline` , `Operation` e `Snapshot ID`.	5 de janeiro de 2018
Novos tópicos	Uma nova seção, Como operações do Patch Manager funcionam, oferece detalhes técnicos que explicam como o Patch Manager determina quais patches de segurança devem ser instalados e como ele os instala em cada sistema operacional compatível. Oferece também informações sobre como as regras de linha de base de patch funcionam em diferentes distribuições do sistema operacional Linux.	2 de janeiro de 2018

Alteração	Descrição	Data de lançamento
Referência de ações de automação do Systems Manager reintitulada e movida de lugar	Com base no feedback dos clientes, a referência de ações de automação agora é chamada de Referência de runbooks do Systems Manager Automation. Além disso, mudamos a referência para Recursos compartilhados > nó Documentos para que fique mais próximo do Referência de plug-ins de documentos de comando. Para obter mais informações, consulte Referência de ações do Systems Manager Automation.	20 de dezembro de 2017
Novo capítulo sobre monitoramento e novo conteúdo	Um novo capítulo, Registrar em log e monitorar no AWS Systems Manager, fornece instruções para enviar métricas e dados de log para o Amazon CloudWatch Logs. Um novo tópico, Enviar logs de nós para o CloudWatch Logs unificado (agente do CloudWatch), fornece instruções para a migração de tarefas de monitoramento somente em instâncias do Windows Server de 64 bits, do SSM Agent para o agente do CloudWatch.	14 de dezembro de 2017
Novo capítulo	Um novo capítulo, Gerenciamento de identidade e acesso para o AWS Systems Manager, fornece informações completas sobre como usar o AWS Identity and Access Management (IAM) e o AWS Systems Manager para ajudar no acesso seguro aos seus recursos por meio do uso de credenciais. Essas credenciais fornecem as permissões necessárias para acessar recursos da AWS, como acessar dados armazenados em buckets do S3 e enviar comandos e ler tags em instâncias do EC2.	11 de dezembro de 2017
Alterações na navegação à esquerda	Alteramos cabeçalhos na navegação à esquerda deste guia do usuário para que correspondam aos cabeçalhos no novo console do AWS Systems Manager.	8 de dezembro de 2017

Alteração	Descrição	Data de lançamento
Várias alterações no re: Invent 2017	• Lançamento oficial do AWS Systems Manager: o AWS Systems Manager (anteriormente Amazon EC2 Systems Manager) é uma interface unificada que permite centralizar dados operacionais facilmente e automatizar tarefas em recursos da AWS. Você pode acessar o novo console do AWS Systems Manager aqui. Para obter mais informações, consulte O que é o AWS Systems Manager?. • Compatibilidade com o YAML: você pode criar documentos do SSM no YAML. Para obter mais informações, consulte Documentos do AWS Systems Manager.	29 de novembro de 2017
Uso do Run Command para tirar snapshots de volumes do EBS	Com o Run Command, é possível gerar snapshots consistentes com a aplicação de todos os volumes do Amazon Elastic Block Store (Amazon EBS) anexados a suas instâncias Windows do Amazon EC2. O processo de snapshot usa o Serviço de Cópias de Sombra de Volume (VSS) do Windows para fazer backups no nível da imagem das aplicações que reconhecem o VSS, incluindo os dados de transações pendentes entre essas aplicações e o disco. Além disso, você não precisa desligar as instâncias ou desconectá-las quando precisar fazer backup de todos os volumes anexados. Para obter mais informações, consulte Tirar snapshots ativados pelo Microsoft VSS usando o AWS Systems Manager no Guia do usuário do Amazon EC2.	20 de novembro de 2017

Alteração	Descrição	Data de lançamento
Segurança aprimorada do Systems Manager disponível por meio de endpoints da VPC	Você pode melhorar a postura de segurança de suas instâncias gerenciadas (incluindo instâncias gerenciadas em seu ambiente híbrido) ao configurar o Systems Manager para usar um endpoint da VPC de interface. Os endpoints da interface são habilitados pelo PrivateLink, tecnologia que permite que você acesse privadamente APIs do Amazon EC2 e do Systems Manager usando endereços IP privados. O PrivateLink restringe todo o tráfego de rede entre instâncias gerenciadas, o Systems Manager e o EC2 e a rede da Amazon (as instâncias gerenciadas não têm acesso à Internet). Além disso, você não precisa de um Internet gateway, de um dispositivo NAT ou de um gateway privado virtual. Para obter mais informações, consulte Melhorar a segurança das instâncias do EC2 usando endpoints da VPC para o Systems Manager.	7 de novembro de 2017

Alteração	Descrição	Data de lançamento
Recurso de inventário para arquivos, serviços, funções do Windows e o Registro do Windows	O SSM Inventory agora comporta a coleta das seguintes informações de suas instâncias gerenciadas. • Arquivos: nome, tamanho, versão, data de instalação, horário de modificação e último acesso etc. • Serviços: nome, nome de exibição, status, serviços dependentes, tipo de serviço, tipo de início etc. • Registro do Windows: caminho da chave do registro, nome do valor, tipo de valor e valor. • Funções do Windows: nome, nome de exibição, caminho, tipo de recurso, estado de instalação etc. Antes de tentar coletar informações para esses tipos de inventário, atualize o SSM Agent nas instâncias em que deseja gerar inventário. Ao executar a versão mais recente do SSM Agent, você garante que pode coletar metadados para todos os tipos de inventário comportados. Para obter informações sobre como atualizar o SSM Agent usando o State Manager, consulte Walkthrough: Automatically update SSM Agent with the AWS CLI. Para obter mais informações sobre inventário, consulte Saiba mais sobre o Systems Manager Inventory.	6 de novembro de 2017
Atualizações na documentação de automação	Correção de vários problemas nas informações sobre definição e configuração de acesso ao Systems Manager Automation. Para obter mais informações, consulte Configurar a automação.	31 de outubro de 2017

Alteração	Descrição	Data de lançamento
Integração ao GitHub e ao Amazon S3	Executar scripts remotos: o Systems Manager agora oferece suporte ao download e à execução de scripts em repositório privado ou público do GitHub e do Amazon S3. Usando o documento `AWS-RunRemoteScript` predefinido do SSM ou o plugin `aws:downloadContent` em um documento personalizado do SSM, você pode executar playbooks e scripts do Ansible no Python, Ruby ou PowerShell, entre outros. Essas alterações promovem ainda mais a infraestrutura como código quando você usa o Systems Manager para automatizar a configuração e a implantação de instâncias do EC2 e instâncias gerenciadas on-premises em seu ambiente híbrido. Para obter mais informações, consulte Executar scripts do GitHub e Executar scripts no Amazon S3. Crie documentos compostos do SSM: agora o Systems Manager comporta a execução de um ou mais documentos secundários do SSM usando um documento primário do SSM. Esses documentos primários que executam outros documentos são chamados de documentos compostos. Os documentos compostos permitem criar e compartilhar um conjunto padrão de documentos secundários do SSM nas contas da Contas da AWS para tarefas comuns, como inicialização de software antivírus ou integração de domínio de instâncias. Você pode executar documentos compostos e secundários armazenados no Systems Manager, GitHub ou Amazon S3. Ao criar um documento composto, você poderá executá-lo usando o documento `AWS-RunDocument` predefinido do SSM. Para obter mais informações, consulte Criar documentos compostos e Executar documentos do em locais remotos. Referência de plugins de documentos do SSM: para facilitar o acesso, tiramos a Referência de plugins do	26 de outubro de 2017

Alteração	Descrição	Data de lançamento
	SSM para documentos do SSM da Referência de API do Systems Manager e transferimos para o Guia do Usuário. Para obter mais informações, consulte Referência de plug-ins de documentos de comando.	
Compatibilidade com versões de parâmetro no Parameter Store	Agora, ao editar um parâmetro, o Parameter Store itera automaticamente o número da versão com um incremento de 1. Você pode especificar o nome de um parâmetro e um número de versão específico em chamadas de API e em documentos do SSM. Se não especificar um número de versão, o sistema usará automaticamente a versão mais recente. As versões de parâmetro fornecem uma camada de proteção para o caso de um parâmetro ser alterado acidentalmente. Você pode visualizar os valores de todas as versões e consultar versões anteriores, se necessári o. Você pode também usar versões de parâmetro para ver quantas vezes um parâmetro mudou ao longo de um período. Para obter mais informações, consulte Como trabalhar com versões de parâmetros no Parameter Store.	24 de outubro de 2017
Support para marcação de documentos do Systems Manager	Agora você pode usar a API AddTagsToResource, a AWS CLI ou o Ferramentas da AWS para PowerShell para marcar documentos do Systems Manager com pares de chave/valor. A marcação ajuda a identificar rapidamen te recursos específicos com base nas tags que você atribuiu a eles. Além disso, há suporte para a marcação em instâncias gerenciadas, janelas de manutenção, parâmetro s do Parameter Store e linhas de base de patches.	3 de outubro de 2017

Alteração	Descrição	Data de lançamento
Diversas atualizações de documentação para corrigir erros ou atualizar conteúdo com base em comentários	• Atualização de Gerenciar nós em ambientes híbridos e multinuvem com o Systems Manager com informações para o Raspbian Linux. • Atualizado o Gerenciar instâncias do EC2 com o Systems Manager com o novo requisito para instâncias do Windows Server. O SSM Agent requer o Windows PowerShell 3.0 ou posterior para executar determinados documentos do Windows Server em instâncias do `AWS-ApplyPatchBaseline` (por exemplo, o documento herdado do SSM). Verifique se as suas instâncias do Windows Server estão executando o Windows Management Framework 3.0 ou posterior. O framework inclui o PowerShell. Para obter mais informações, consulte o Windows Management Framework 3.0.	2 de outubro de 2017
Solução de problemas de instâncias do Windows inacessíveis usando o fluxo de trabalho de Automação EC2Rescue	O EC2Rescue pode ajudar a diagnosticar e solucionar problemas em instâncias do Amazon EC2 no Windows Server. Você pode executar a ferramenta como um fluxo de trabalho do Systems Manager Automation usando o documento AWSSupport-ExecuteEC2Rescue. O documento AWSSupport-ExecuteEC2Rescue foi projetado para realizar uma combinação de ações do Systems Manager, ações do AWS CloudFormation e funções do Lambda que automatizam as etapas normalmente necessárias para usar EC2Rescue. Para obter mais informações, consulte Executar a ferramenta EC2Rescue em instâncias inacessíveis.	29 de setembro de 2017
SSM Agent Instalado por padrão no Amazon Linux	Por padrão, o SSM Agent é instalado em AMIs do Amazon Linux de 09/2017 e posteriores. Instale manualmente o SSM Agent em outras versões do Linux, conforme descrito em Trabalhar com o SSM Agent em instâncias do EC2 para Linux.	27 de setembro de 2017

Alteração	Descrição	Data de lançamento
Melhorias no Run Command	O Run Command contém as melhorias a seguir. • É possível restringir a execução de comandos a instâncias específicas ao criar e atribuir uma política do IAM que inclua uma condição em que o usuário poderá somente executar comandos em instâncias que estiverem marcadas com etiquetas específicas do Amazon EC2. Para obter mais informações, consulte Restringir o acesso ao Run Command com base em etiquetas. • Você tem mais opções para definir destinos para instâncias usando tags do Amazon EC2. Agora, você pode especificar várias chaves de rótulo e vários valores de tag ao enviar comandos. Para obter mais informaçõ es, consulte Execução de comandos em escala.	12 de setembro de 2017
Systems Manager com suporte no Raspbian	Agora, o Systems Manager pode ser executado em dispositivos Raspbian Jessie e Raspbian Stretch, incluindo o Raspberry Pi (32 bits).	7 de setembro de 2017
Enviar automatic amente os logs do SSM Agent para o Amazon CloudWatc h Logs	Agora, você pode fazer uma alteração de configuração simples em suas instâncias para que o SSM Agent envie arquivos de log ao CloudWatch. Para obter mais informaçõ es, consulte Enviar logs do SSM Agent ao CloudWatch Logs.	7 de setembro de 2017
Criptografe a sincronização de dados dos recursos	Com a sincronização de dados de recursos do Systems Manager, é possível agregar dados de inventário coletados em dezenas ou centenas de instâncias gerenciadas em um bucket central do S3. Agora, você pode criptografar a sincronização de dados de recursos usando uma chave do AWS Key Management Service. Para obter mais informaçõ es, consulte Demonstração: usar a sincronização de dados de recursos para agregar dados do inventário.	1 de setembro de 2017

Alteração	Descrição	Data de lançamento
Novas demonstrações do State Manager	Duas novas demonstrações adicionadas à documentação do State Manager: Walkthrough: Automatically update SSM Agent with the AWS CLI Walkthrough: Automatically update PV drivers on EC2 instances for Windows Server	31 de agosto de 2017
Conformidade de configuração do Systems Manager	Use a Conformidade de Configuração para escanear sua frota de instâncias gerenciadas quanto à conformidade de patches e inconsistências de configuração. Você pode coletar e agregar dados de várias Contas da AWS e Regiões da AWS e depois fazer buscas detalhadas em recursos específicos que não forem compatíveis. Por padrão, Configuration Compliance (Conformidade das configurações) exibe dados de conformidade sobre a aplicação de patches do Patch Manager e as associações do State Manager. Você também pode personalizar o serviço e criar seus próprios tipos de conformidade com base nos seus requisitos de TI ou negócios. Para obter mais informações, consulte AWS Systems Manager Compliance.	28 de agosto de 2017
Nova ação de automação: `aws:executeAutomation`	Executa um fluxo de trabalho de Automação secundário chamando um runbook de Automação secundário. Com essa ação, você pode criar runbooks de Automação para seus fluxos de trabalho mais comuns e fazer referência a esses documentos durante uma execução da Automação. Essa ação pode simplificar seus runbooks de Automação, removendo a necessidade de duplicar etapas em runbooks semelhantes. Para obter mais informações, consulte `aws:executeAutomation` – Executa outra automação.	22 de agosto de 2017

Alteração	Descrição	Data de lançamento
Automação como destino do CloudWatch Event	Você pode iniciar um fluxo de trabalho de Automação especificando um runbook de Automação como o destino de um evento do Amazon CloudWatch. É possível iniciar fluxos de trabalho de acordo com um cronogram a ou quando ocorrer um evento específico do sistema AWS. Para obter mais informações, consulte Executar automações com base em eventos do EventBridge.	21 de agosto de 2017
Versionamento de associações e atualizações gerais do State Manager	Agora, você pode criar diferentes versões de associaçã o do State Manager. Existe uma cota de 1.000 versões para cada associação. Você também pode especificar nomes para suas associações. Além disso, a documenta ção do State Manager foi atualizada para tratar informaçõ es desatualizadas e inconsistências. Para obter mais informações, consulte AWS Systems Manager State Manager.	21 de agosto de 2017

Alteração	Descrição	Data de lançamento
Alterações em Maintenance Windows	Maintenance Windows inclui as seguintes alterações ou melhorias: • Anteriormente, a Maintenance Windows podia executar tarefas somente usando o Run Command. Agora, você pode executar tarefas usando o Systems Manager Automation, o AWS Lambda e o AWS Step Functions. • É possível editar destinos de uma janela de manutenção, além de especificar um nome, uma descrição e um proprietário de destino. • É possível editar tarefas em uma janela de manutenção, incluindo especificar um novo documento do SSM para o Run Command e tarefas de Automação. • Agora, todos os parâmetros do Run Command são comportados, incluindo DocumentHash, DocumentHashType, TimeoutSeconds, Comment e NotificationConfig. • Agora, você pode usar um sinalizador `safe` ao tentar cancelar o registro de um destino. Se ativado, o sistema retornará um erro se o destino for referenciado por qualquer tarefa. Para obter mais informações, consulte AWS Systems Manager Maintenance Windows.	16 de agosto de 2017
Nova ação de automação: `aws:approve`	Essa nova ação para runbooks de Automação pausa temporariamente uma execução de Automação até que as entidades principais designadas aprovem ou rejeitem essa ação. Depois que o número necessário de aprovações for atingido, a execução da Automação será retomada. Para obter mais informações, consulte Referência de ações do Systems Manager Automation.	10 de agosto de 2017

Alteração	Descrição	Data de lançamento
Função de admissão do Automation não mais necessária	Antes, a Automação exigia especificar uma função de serviço (ou uma função de admissão) para que o serviço tivesse permissão para realizar ações em seu nome. A Automação não exige mais essa função, pois o serviço agora opera usando o contexto do usuário que invocou a execução. No entanto, as seguintes situações ainda exigem que você especifique uma função de serviço para Automação: • Quando você quer restringir as permissões de um usuário em um recurso, mas deseja que esse usuário execute um fluxo de trabalho de automação que exija privilégios elevados. Nesse cenário, você poderá criar uma função de serviço com permissões elevadas e permitir que o usuário execute o fluxo de trabalho. • As operações que você espera executar por mais de 12 horas exigem uma função de serviço. Para obter mais informações, consulte Configurar a automação.	3 de agosto de 2017
Conformidade de configuração	Use a Conformidade de configuração do Amazon EC2 Systems Manager para verificar sua frota de instância s gerenciadas no que diz respeito à conformidade de patches e a inconsistências de configuração. Você pode coletar e agregar dados de várias Contas da AWS e Regiões da AWS e depois fazer buscas detalhadas em recursos específicos que não forem compatíveis. Para obter mais informações, consulte AWS Systems Manager Compliance.	8 de agosto de 2017

Alteração	Descrição	Data de lançamento
Aprimoramento em documentos do SSM	Agora, documentos de Comando e Política do SSM oferecem suporte entre plataformas. Isso significa que um único documento do SSM pode processar plugins para sistemas operacionais Windows e Linux. O suporte entre plataformas permite consolidar o número de documentos que você gerencia. O suporte entre plataformas é oferecido em documentos do SSM que usam o esquema versão 2.2 ou posterior. Os documentos de Comando do SSM que usam o esquema versão 2.0 ou posterior agora podem incluir vários plugins do mesmo tipo. Por exemplo, você pode criar um documento de Comando que chama o plugin `aws:runRunShellScript` várias vezes. Para obter mais informações sobre as alterações do esquema versão 2.2, consulte Documentos do AWS Systems Manager. Para obter mais informações sobre plug-ins do SSM, consulte Command document plugin reference.	12 de julho de 2017

Alteração	Descrição	Data de lançamento
Aplicação de patches do Linux	O Patch Manager agora pode corrigir as seguintes distribuições do Linux: Sistemas de 64 bits e 32 bits • Amazon Linux 2014.03, 2014.09 ou posterior • Ubuntu Server 16.04 LTS, 14.04 LTS ou 12.04 LTS • Red Hat Enterprise Linux (RHEL) 6.5 ou posterior Somente sistemas de 64 bits • Amazon Linux 2015.03, 2015.09 ou posterior • Red Hat Enterprise Linux (RHEL) 7.x ou posterior Para obter mais informações, consulte AWS Systems Manager Patch Manager. Note • Para aplicar patch a instâncias do Linux, essas instâncias devem executar a versão 2.0.834.0 ou posterior do SSM Agent. Para obter informações sobre como atualizar o agente, consulte a seção intitulada Exemplo: atualizar o SSM Agent no Executar comandos no console. • O `AWS-ApplyPatchBaseline` Documento do SSM está sendo substituído pelo `AWS-RunPatchBaseline` document.	6 de julho de 2017

Alteração	Descrição	Data de lançamento
Sincronização de dados de recursos	Você pode usar a sincronização de dados de recursos do Systems Manager para enviar dados do inventário coletados de todas as suas instâncias gerenciadas para um único bucket do Amazon S3. A sincronização de dados de recursos atualizará automaticamente os dados centralizados quando novos dados de inventário forem coletados. Com todos os dados de inventário armazenados em um bucket do S3 de destino, você pode usar serviços, como o Amazon Athena e o Amazon QuickSight, para consultar e analisar os dados agregados. Para obter mais informações, consulte Creating a resource data sync for Inventory. Para obter um exemplo de como trabalhar com a sincronização de dados de recursos, consulte Demonstração: usar a sincronização de dados de recursos para agregar dados do inventário.	29 de junho de 2017
Hierarquias de parâmetros do Systems Manager	Gerenciar dezenas ou centenas de parâmetro do Systems Manager como uma lista simples é um processo demorado e propenso a erros. Você pode usar hierarquias de parâmetros para ajudar a organizar e gerenciar parâmetros do Systems Manager. Uma hierarquia é um nome de parâmetro que inclui um caminho que você define usando barras. Veja a seguir um exemplo que usa três níveis de hierarquia no nome para identificar o seguinte: /Environment/Type of computer/Application/Data `/Dev/DBServer/MySQL/db-string13` Para obter mais informações, consulte Trabalhar com hierarquias de parâmetros no Parameter Store.	22 de junho de 2017

Alteração	Descrição	Data de lançamento
Suporte do SSM Agent para SUSE Linux Enterprise Server	Você pode instalar o SSM Agent no SUSE Linux Enterprise Server de 64 bits (SLES). Para obter mais informações, consulte Trabalhar com o SSM Agent em instâncias do EC2 para Linux.	14 de junho de 2017

Convenções do documento

Veja a seguir as convenções tipográficas comuns para o Guia do usuário do AWS Systems Manager.

Exemplos diferenciados para sistemas operacionais locais ou idiomas de linha de comando

Usamos guias para apresentar exemplos diferentes de comandos com base no tipo de sistema operacional local do usuário. Em exemplos do Linux e macOS, usamos barra invertida (\) para quebrar comandos longos em várias linhas. Para exemplos do Windows Server, usamos o caractere circunflexo (^) para dividir os comandos em múltiplas linhas.

Exemplo:

Linux & macOS

```
aws ssm update-service-setting \
    --setting-id arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
instance/activation-tier \
    --setting-value advanced
```

Windows

```
aws ssm update-service-setting ^
    --setting-id arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
instance/activation-tier ^
    --setting-value advanced
```

Elementos da interface de usuário

Formatação: texto em negrito

Exemplo: selecione File, Properties.

Entrada do usuário (texto que um usuário digita)

Formatação: texto em uma fonte monoespaçada

Exemplo: para o nome, digite **`my-new-resource`**.

Texto de espaço reservado para um valor necessário

Formatação: texto em *itálico*

Exemplo:

```
aws ec2 register-image --image-location amzn-s3-demo-bucket/image.manifest.xml
```